Event-Based Energy Disaggregation Algorithm for Activity Monitoring From a Single-Point Sensor

José Alcalá, Jesús Ureña, *Senior Member, IEEE*, Álvaro Hernández, *Senior Member, IEEE*, David Gualda

***Abstract*— The massive deployment of smart meters and other customized meters has motivated the development of nonintrusive load monitoring (NILM) systems. This is the process of disaggregating the total energy consumption in a building into individual electrical loads using a single-point sensor. Most literature is oriented to energy saving. Nevertheless, activity of daily livings monitoring through NILM is recently receiving much interest. This proposal presents an event-based NILM algorithm of high performance for activity monitoring applications. This is divided into two stages: 1) an event detector and 2) an event classification algorithm. The first one does not need to be trained and shows a detection rate up to 94%. The event classification algorithm uses a novel load signature based on trajectories of active, reactive, and distortion power (PQD) to obtain general models of appliance classes using principal component analysis. The F1 score and the F0.5 score (the last one is more relevant to activity monitoring) draw values of 90.6% and 98.5, respectively.**

***Index Terms*— Activity monitoring, energy disaggregation, event detection, nonintrusive load monitoring (NILM), principal component analysis (PCA), smart metering.**

I. Introduction

The electricity market is rapidly changing since the breakthrough of smart grids. This has motivated many studies around smart meters or other customized meters. Special attention is drawn to the appliance-level consumption and activity monitoring. Reference [1] uses low-power protocols to monitor appliances. Nevertheless, sensors might be expensive and too much intrusive, which makes difficult the deployment. In [2], blind source separation is used to disaggregate vibratory information from a single-point sensor to monitor the activity of machines in factories and to reduce the number of sensors. Nonintrusive load monitoring (NILM) techniques can solve this issue. This is the process of disaggregating the total energy consumption in a building into individual electrical loads using a single-point sensor, in most cases the smart meter. First appearance dated from 1992 in [3]. In this study, step-like variations in the active and reactive power (P and Q) readings from the mains were used to detect and classify events (i.e. changes on the power consumption due to appliance connections or state changes). Nevertheless, it has not been until the smart metering emergence that the interest of many researches has raised.

Typically, the smart metering standardization regulates the measurement of harmonics to evaluate distortion in the mains [4], [5]. That means regularized smart meters sample up to some kilohertz [6], [7]. However, this sampling rate is only available internally and they transmit their readings to the central station at a lower resolution. In [8], the importance of enabling access to the firmware is highlighted to make feasible new smart meter applications. Power line communication is often used for transmitting load curves that have been downsampled to one sample every 15 or 30 min. In some countries, such as U.K., smart meters may include a wireless port (e.g., ZigBee) to enable higher resolutions (i.e. load curves sampled every 1 or 10 s) via the home area network (HAN). Hence, most recent efforts on NILM have been focused on this sampling rate, since measurements can be easily collected and sent to a cloud service where disaggregation is performed [9]. Nevertheless, communication through the HAN is not guaranteed in most countries. Accordingly, many research groups [10] and third-party companies have dedicated their efforts to develop NILM algorithms with higher data rate to improve accuracy, as well as to enable real-time applications based on single-point sensors. These require measurements from only one location and they comprise smart meter, whether the firmware can be modified, and third-party devices.

The use of load curves sampled every 1 or 10 s makes the event detection (i.e., appliances switching ON and OFF) impractical and, consequently, probabilistic models, such as hidden Markov models [11]–[13] and deep neural networks (NNs) [14], are approached. They are known as eventless NILM algorithms and they predict the state of each appliance during every sampling period, what exponentially increases the computational cost as the number of appliances to be predicted does. The aim is to save energy and, likewise, these algorithms are optimized to disaggregate the major appliances (e.g., the air conditioning), whereas the disaggregation error rapidly increases for minor (less power consuming) appliances.

On the other hand, event-based NILM algorithms are those where events are detected and classified, using an event detector and an event classification algorithm, respectively. They provide a finer granularity at a lower computational burden than eventless NILM algorithms, thanks to the use of a higher resolution data. Likewise, a sampling rate in

Manuscript received December 16, 2016; revised February 28, 2017; accepted April 24, 2017. This work was supported in part by the Spanish Ministry of Economy and Competitiveness under SOC-PLC project (TEC2015-64835-C3-2-R) and TARSIUS project (TIN2015-71564-c4-1-R). The Associate Editor coordinating the review process was Dr. Carlo Muscas. *(Corresponding author: Jesús Ureña.)*

The authors are with the Department of Electronics, University of Alcalá, E-28805 Madrid, Spain (e-mail: jesus.urena@uah.es; alvaro.hernandez@uah.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIM.2017.2700987

the range of 600 Hz enables the analysis of transient states during the device startup that typically lasts 100–500ms [10]. Higher frequencies than 1.2 kHz allow harmonics analysis [15] and, even higher (i.e., from 20 kHz to 30 MHz), switch mode power supplies and electromagnetic interference enable to differentiate between the same types of appliances (e.g., fluorescent lights) placed in different rooms [16]. Event-based algorithms are commonly divided into two stages: 1) event detection and 2) event classification.

Event detectors aim to detect transient states. There are three approaches: 1) expert heuristics; 2) probabilistic models; and 3) matched filters [17]. Expert heuristics methods require a prior knowledge about appliances to be detected. For instance, the major appliances are detected based on rules, such as duration of the transient state [18]. Probabilistic models perform suitably, i.e., around 95% of detection rate. Nevertheless, it is necessary to train the statistics models in different scenarios (i.e., different household mains) and to initialize some parameters [17], [19], [20]. Matched filters may require higher data acquisition rates and computation capabilities. Although they may require parameters' initialization, no training process is needed. Signal normalization is usually applied to shield the analysis voltage drops and to generalize their approaches along houses. Approaches point to the use of average and derivative filters whence the classification is also performed; they obtain recognition rates (i.e., appliance events correctly classified) of 87%. The envelope extraction from the current signal is also a common technique and, for that purpose, the Hilbert transform and the Kalman filter can be used [21], [22].

Regarding the classification algorithms, they often have two processes: 1) learning load signatures and 2) their classification. Load signatures must be carefully selected to avoid overlapping [15], [23]. The latter uses wave shape features based on V–I trajectories, obtaining a high recognition rate (i.e., 98% under lab conditions). Furthermore, there are "delta form" (signature of a single appliance) and "snapshots" (aggregated signatures of multiple appliances). Active power P and reactive power Q are the most frequently features used to model load behavior [24]. These variables characterize load appliances (e.g., resistive loads, such as a kettle, have active power and a null reactive power). The generated distortion power D from nonlinear devices allows to achieve a higher performance [25].

Within the classification algorithms, clustering of signatures is a recurrent technique. Likewise, P, Q, and D delta values are extracted during steady states and grouped by clusters [3], [25]. Moreover, NN-based approaches, support vector machine, and adaptive boost techniques are used in [23] and [26]. The accuracy is high (almost 100%), but it rapidly drops when nonideal conditions are introduced (i.e., for some appliances, it decreases down to 60%). Multisignatures and multialgorithms are used in [27]. Then, a committee decision mechanism (CDM) decides on the disaggregation based on the outcome from those multialgorithms. The results are presented for simulated data and the accuracy in the disaggregation is 92.7% for the CDM, in this case a maximum likelihood estimator, before the addition of noise.

Event-based algorithms are appropriated for real-time diagnostic and activity recognition applications [23]. In particular, the health monitoring of the elderlies through activity of daily livings (ADLs) is a relevant topic. There is a strong correlation between ADLs and energy disaggregation. The *a priori* knowledge of ADLs helps to improve the disaggregation performance [28]. Furthermore, a coarse track of activities can be performed using 15-min data readings from the water meter [29]. However, the ADLs tracking is a more complex task that needs a finer granularity for many appliances as shown in [30]–[32]. Thus, an event-based disaggregation algorithm is proposed for monitoring ADLs in [33], where accuracy is 86.25%. On the other hand, there are studies that infer the room occupancies through NILM for a fine-grained activity monitoring [34]. However, this entails to map appliances to rooms and, therefore, a labeling process is previously required, which is specific for each household. The proposal here described is focused only on monitoring the activity based on the appliance usage patterns, as in [35]. This implies that the NILM algorithm should be as much generalizable as possible along households.

In summary, event detectors based on probabilistic models may work better than other approaches. Nevertheless, a training process is still required. Concerning classification algorithms, it can be observed that there is not yet a predominant load signature or energy disaggregation algorithm that performs well along all households, as noted in [15]. Besides, in most event-based NILM studies, there is not a standard metric used and, most often, data are simulated or produced from a local data set. Thence, in this context, the major contributions in this paper are as follows.

1) A novel event-based disaggregation algorithm is based on a detection module and on a classification stage, which have a high performance for activity analysis applications and can be included in actual third-party systems. In addition, it might be potentially implemented in next generations of smart meters.
2) The event-detector performance used is based on the envelope extraction and no learning process is required, only an initialization to set one parameter. After its evaluation with standard metrics and over real data sets, it has similar results than the probabilistic models that need learning processes.
3) A new type of load signature is presented: trajectories of active, reactive, and distortion powers (*PQD)*. The signature spams from the previous steady state, through the transient state of the event occurrence and up to the next steady state. Thus, more information is provided and the process of isolating transient and steady states is not needed.
4) The PQD-principal component analysis (PCA) classification algorithm learns the signatures along appliances of the same type but different brands, obtaining general appliance class models. Thereby, our proposal can be deployed in new houses with no training process. This is tested in Section III-D over a public data set. Around 102 different households and appliance brands are used. The results show a high performance with an accuracy

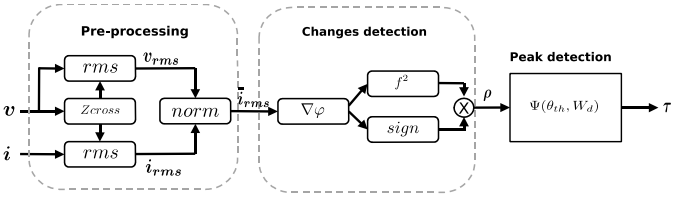

Fig. 1. Block diagram for the proposed event detector.

of 98%. Furthermore, the F0.5 score (which becomes more important for activity analysis, see discussion in Section III-D) is 98.6%.

As for the PCA technique presented in this paper, it is a widespread statistical procedure that has been used in many machine learning domains such vision computing [36] as well as in ultrasonic localization [37]. Applied to the disaggregation domain, it has been previously applied in [38] to separate the gas–liquid two-phase flow rig through electrical resistance tomography.

The rest of this paper is organized as follows. Section II describes the proposal; after a thorough evaluation of the event detector and the disaggregation algorithm, using two different data sets, is presented in Section III; finally, Section IV deals with the conclusions and future works.

II. Model Description

The proposed NILM algorithm is described hereinafter, split into two major blocks. Section II-A details the event detection process, whereas Section II-B explains the disaggregation algorithm that lies over the event detection. It is important to remark that both stages are independent, so different parameters will be defined for them: the only link between them is that the outcome of the event detector (detected event and its corresponding timestamp) becomes the income of the disaggregation algorithm. In this way, both stages can operate at different sampling frequencies and with different settings.

A. Event Detection Algorithm

The main contribution of the proposed event detection algorithm is the high performance obtained without a training process. The detected transient states will be labeled by the disaggregation algorithm (see Section II-B). This proposal is based on the detection of changes in the envelope of the acquired current signal. The analysis of this signal over time instead of its corresponding power has the advantage of discarding network noise due to voltage drops and other effects, as the electrical current is the immediate cause of appliance activities and power also includes voltage.

As can be seen in the block diagram depicted in Fig. 1, the process of the event detection is based on three stages: preprocessing, changes detection, and peak detection.

Furthermore, Fig. 2 illustrates a case of detection to facilitate the explanation.

1) Preprocessing: This proposal averages the root mean square (rms) from the current signal every cycle of the utility frequency to filter the 50- or 60-Hz component and other information or noise, keeping only the transient states

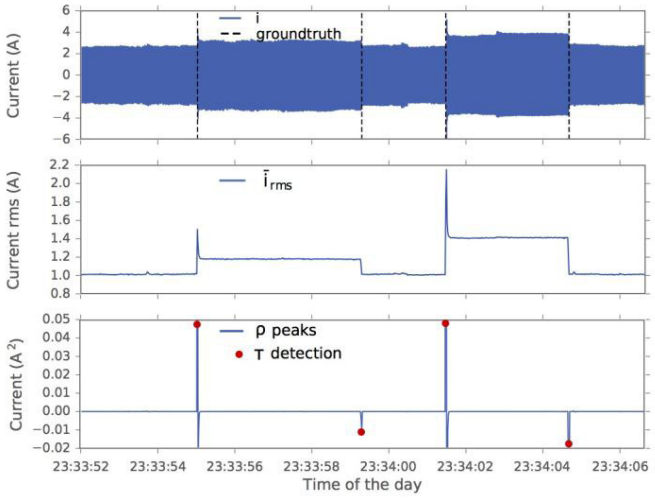

Fig. 2. Illustration of the event detection algorithm. (a) $\boldsymbol{i}$ current waveform from the mains and ground truth of events. (b) Envelope extraction: $\bar{i}_{\text{rms}}$ normalized rms value for the current. (c) $\boldsymbol{\rho}$ peak signal and $\boldsymbol{\tau}$ event detection.

information. The outcome of this block is the signal envelope that contains valuable information about the transient states. Furthermore, skews in frequency and drops in voltage are discarded using the block "Zcross" for synchronization and the block "norm" for normalization, respectively.

Let us consider the inputs described in the following as the voltage and current signals from the mains: $\boldsymbol{v}$ and $\boldsymbol{i}$, respectively:

$$\mathbf{v} = [v_0, v_1, \ldots, v_{N-1}] \tag{1}$$

$$\mathbf{i} = [i_0, i_1, \ldots, i_{N-1}] \tag{2}$$

where N is the number of samples over the monitoring time. The event detector is a real-time algorithm; thus, the length for N does not influence on the performance and the monitoring time could be from minutes to weeks. However, for the sake of computation, results on Section III-C are evaluated through a window of about 2 min. Fig. 2(a) depicts the current waveform $\boldsymbol{i}$ measured at a certain time from the mains in the BLUED data set (see Section III-A). A zooming window of 14 s is applied to facilitate the visualization of graphs. The voltage waveform $\boldsymbol{v}$ has not been plotted since it has very little changes and does not contribute to understand the process. It is a 60-Hz sinusoidal signal of 170 V of amplitude (i.e., 120 Vrms) that may have small amplitude variations due to voltage drops as well as a phase shift with respect to the current $\boldsymbol{i}$. Moreover, four events (ground truth) have been marked with vertical dashed lines. These events correspond to state changes in one appliance, in this case the refrigerator.

The rms value of the signal is evaluated at 60 or 50 Hz; this depends on the utility frequency of the standardization. For that, the block "Zcross" detects the zero crossing of the n-D vector $\mathbf{v}$ and it passes them to the block "rms." This provides the starting indexes of $\mathbf{v}$ for every utility cycle. Afterward, the block "rms" divides $\mathbf{v}$ and $\mathbf{i}$ into smaller vectors as defined in (3) and (4), where S is the number of cycles. The utility frequency is prone to suffer from skews and small variations

due to external factors; consequently, it should be denoted that $\mathbf{v_s}$ must have the same number of samples as $\mathbf{i_s}$, but it may have a different number compared with $\mathbf{v_{s+1}}$. Finally, the block "rms" computes $\mathbf{v}_{\text{rms}}$ as follows in (5) and (6), where $v_{s,j}$ is the jth element for the vector cycle $\mathbf{v_s}$ described in (3), and N_s is the number of samples per cycle s. Likewise, $\mathbf{i}_{\text{rms}}$ is also evaluated in a similar way

$$\mathbf{v} = \{\mathbf{v_0}, \ldots, \mathbf{v_s}, \ldots, \mathbf{v_{S\text{-}1}}\} \tag{3}$$

$$\mathbf{i} = \{\mathbf{i_0}, \ldots, \mathbf{i_s}, \ldots, \mathbf{i_{S\text{-}1}}\} \tag{4}$$

$$\mathbf{v}_{\text{rms}} = [v_{\text{rms},0}, \ldots, v_{\text{rms},s}, \ldots, v_{\text{rms},S-1}] \tag{5}$$

$$v_{\text{rms},s} = \sqrt{\frac{1}{N_s}\sum_{j=0}^{N_s-1} v_{s,j}^2} \quad \forall s = 0, 1, 2, \ldots, S-1. \tag{6}$$

The voltage may vary over time and this could also imply changes in the amplitude of the current. A normalization prevents from those variations. The block "norm" carries out this normalization for each cycle s as detailed in

$$\bar{\mathbf{i}}_{\text{rms},s} = \frac{{V_{\text{norm}}}^2}{v_{\text{rms},s}^2} * i_{\text{rms},s} \quad \forall s = 0, 1, 2, \ldots, S-1. \tag{7}$$

The parameter V_{norm} is the normalized voltage (e.g., $V_{\text{norm}} = 220$ V_{rms} in Europe and $V_{\text{norm}} = 120$ V_{rms} in UE). Thus, the outcome of this block is an N-D vector $\bar{\mathbf{i}}_{\mathbf{rms}}$ plotted in Fig. 2(b). The resulting signal is the proposed envelope extraction from the current waveform $\boldsymbol{i}$. It presents step-like changes due to transient states of the refrigerator. Likewise, it is worthy to note that $\mathbf{i_{rms}}$ is susceptible to voltage drops and it may vary its amplitude for the same transient state produced at a different time. Nevertheless, the normalized $\bar{\mathbf{i}}_{\mathbf{rms}}$ signal guaranties the same amplitude always.

2) Changes Detection: The evaluation of changes in the aforementioned preprocessed signal $\bar{\mathbf{i}}_{\mathbf{rms}}$ allows the detection of connections and disconnections of appliances in the mains. By performing a derivation, only changes in the signal are kept. This process filters steady states, leaving spikes that correspond to transient states. Then, the derived signal $\nabla\bar{\mathbf{i}}_{\mathbf{rms}}$ is squared to increase the gap between small changes due to noise, and significant changes due to transient states. This is a nonlinear transformation (squaring) used in signal processing that flattens small values and emphasizes larger ones (greater than $1A^2$) [39]. Although this is not a critical step, in many cases, this improves detection while keeping the performance for the rest.

It is worth noting that the aforementioned derived signal $\nabla\bar{\mathbf{i}}_{\mathbf{rms}}$ draws positive or negative spikes to reflect the type of event. In the case of positive events, such as turning on or passing to a state of higher consumption, the rms value of the current signal $\bar{\mathbf{i}}_{\mathbf{rms}}$ increases and, therefore, a positive spike is obtained. Exactly the opposite happens with negative events such as turning OFF appliances, where negative spikes are reflected. In order to keep track of the type of event (positive or negative), the sign of the derived signal $\nabla\bar{\mathbf{i}}_{\mathbf{rms}}$ is included after the squaring in the resulting signal $\boldsymbol{\rho}$. This process is shown in Fig. 1 and described in (8). For instance, Fig. 2(c) shows the signal $\boldsymbol{\rho}$ in blue line for the case of ON/OFF connections of the refrigerator. Positive spikes in $\boldsymbol{\rho}$ are due to ON switching events, whereas OFF events provide negative ones. It can be observed that $\boldsymbol{\rho}$ keeps only the transient states information from the current waveform presented in $\boldsymbol{i}$

$$p_{\mathbf{s}} = \begin{cases} -(\bar{\imath}_{\text{rms},s} - \bar{\imath}_{\text{rms},s-1})^2, & \text{if } (\bar{\imath}_{\text{rms},s} - \bar{\imath}_{\text{rms},s-1}) < 0 \\ (\bar{\imath}_{\text{rms},s} - \bar{\imath}_{\text{rms},s-1})^{\mathbf{2}}, & \text{if } (\bar{\imath}_{\text{rms},s} - \bar{\imath}_{\text{rms},s-1}) \geq 0 \\ & \forall s = 2, 3, \ldots, S-1. \end{cases} \tag{8}$$

3) Peak Detector: The output $\boldsymbol{\rho}$ presents small peaks produced by noise and high peaks due to events (transient states). The algorithm described in [40] has been applied to detect those peaks from events. The algorithm requires two parameters: 1) a minimum distance between peaks W_d and 2) a threshold θ_{th} that filters out any peak below it.

In this case, transient states may generate oscillations around the main peak and, therefore, a minimum value for W_d should be adjusted. It is worth noting that the smaller the value of W_d, the higher the time resolution. Likewise, the threshold θ_{th} filters out noise from the steady states and allows only those peaks coming from events. There is a significant gap between the amplitude of peaks generated by events and those due to noise, so the threshold θ_{th} should be set according to those intervals. This value θ_{th} has been empirically optimized, based on the receiver operating characteristic (ROC) curve (see Section III-C).

The corresponding output signal $\boldsymbol{\tau}$ is a vector where every element t_k is the timestamp at which events take place, where K is the number of events

$$\boldsymbol{\tau} = [t_0, t_1, \ldots, t_{K-1}]. \tag{9}$$

Fig. 2(c) shows the peak detection (red dots) where $\theta_{\text{th}} = 5(A^2)$ and $W_d = 5$ cycles of the utility frequency (i.e., 83 ms in mains of 60 Hz). In this case, the four events have been detected and the recorded timestamps for the detection compound the signal $\boldsymbol{\tau}$, defined in (9).

B. Event Classification Algorithm

The algorithm previously described in Section II-A tracks the events and registers the timestamp. The next step is to classify them into appliance classes. For this paper, we pursue a high accuracy and, therefore, active, reactive, and distortion powers are used as signatures to avoid overlapping (see Section I). Furthermore, delta values (single appliance signatures) of *PQD* curve trajectories are characterized during the transient state and previous and next steady states. The use of a mix of transient/steady states as load signatures is a novel approach that, to the best of our knowledge, has been presented only in [41]. Hence, the proposed approach involves a preprocessing stage, followed by the proposed classification algorithm PQD-PCA, PCA of active, reactive, and distortion (PQD) powers. This procedure is shown in Fig. 3 and detailed in the following sections.

1) Signal Preprocessing: Extraction of SPQD Powers Signatures: This block extracts signatures and sends them to the PQD-PCA classification block. The proposal deals with "delta form" signatures as explained in Section I. As only the

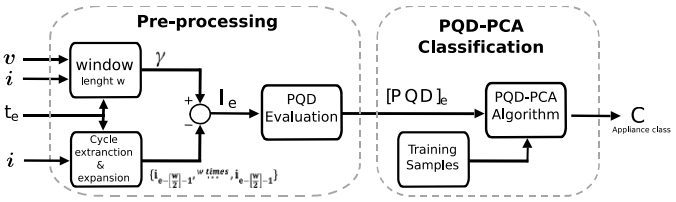

Fig. 3. Block diagram for the proposed PCA-PQD Classification.

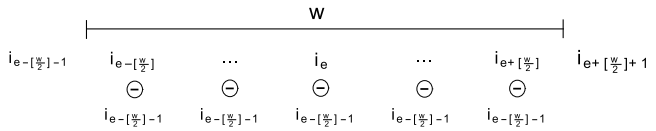

Fig. 4. Extracting delta values from the aggregated current.

aggregated current is provided, it is necessary to preprocess the signal to isolate the current due to the appliance that is causing the event from the rest. Other approaches deal with incremental values of the active and reactive powers (ΔP-ΔQ) as in [3]. Nevertheless, the distortion power D cannot be easily added or subtracted, as it presents nonlinear components. Then, we present an isolation process to extract the current waveform produced for the specific appliance that produces the event; note that the voltage waveform can be considered not affected by the number of appliances connected at each instant as all of them are connected to the same voltage.

As shown in Fig. 3, the current and voltage waveforms are the inputs as well as the occurrence time of a specific event e detected in (9). First, a w-length window is applied to the current waveform $\boldsymbol{i}$ and $\boldsymbol{\gamma}$ (10), a collection of vector-cycle vectors, is obtained, where w is the number of cycles and $\mathbf{i_e}$ is the vector cycle where the event e occurs. This window is centered at the event occurrence and should be large enough to hold a previous steady state, the transient state, and the next steady state. From the experiments, it is fixed $w = 60$ cycles. This is equivalent to a window of 1 s whenever the utility frequency is 60 Hz, which is enough time to complete the transient state by the appliances to be disaggregated. Nevertheless, it is tunable and could be changed regarding requirements. Then, the "cycle extraction and expansion" block filters the current waveform keeping only the vector cycle previous to the windowing ($\mathbf{i_{e-[w/2]-1}}$) and it creates a collection of repeated vectors. Then, the latter is subtracted from $\boldsymbol{\gamma}$ (11). This, windowing, expansion and subtraction process is depicted in Fig. 4

$$\boldsymbol{\gamma} = \left\{\mathbf{i}_{\mathbf{e}-\left[\frac{\mathbf{w}}{2}\right]}, \ldots, \mathbf{i_e}, \ldots, \mathbf{i}_{\mathbf{e}+\left[\frac{\mathbf{w}}{2}\right]}\right\} \tag{10}$$

$$\mathbf{I_e} = \boldsymbol{\gamma} - \left\{\mathbf{i}_{\mathbf{e}-\left[\frac{\mathbf{w}}{2}\right]-\mathbf{1}}, \overset{w \text{ times}}{\ldots}, \mathbf{i}_{\mathbf{e}-\left[\frac{\mathbf{w}}{2}\right]-\mathbf{1}}\right\}. \tag{11}$$

The resulting collection of vectors $\mathbf{I_e}$ approximates the current waveform drawn by the appliance that triggers the event, during the previous and next steady states and the transient state. Then, *PQD* trajectories are obtained in the "PQD evaluation" block, where every vector cycle of $\mathbf{I_e}$ draws a point value on the *PQD* power cube depicted in Fig. 5. The latter also gives the power relations that they are expressed in (10).

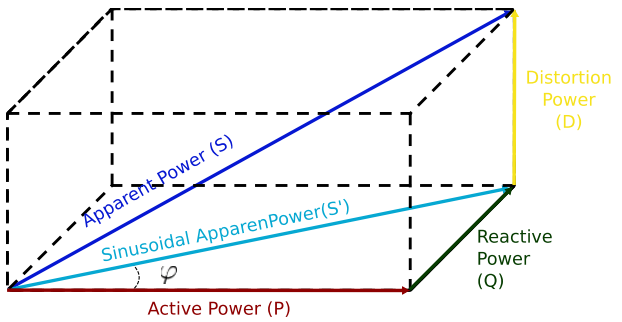

Fig. 5. Representation of the PQD power cube used to model load behavior.

Then, for every vector cycle, the parameters I_{rms} and V_{rms} are evaluated according to (16), where T is the fundamental period that corresponds to the utility frequency. Hence, the apparent power S can be deducted as (13). A fast Fourier transform analysis is applied to obtain the first harmonic of the electrical current I_{rms}, also denoted as the lineal component since it presents the fundamental utility frequency. The rms of the current first harmonic $I_{\text{rms},1}$ is used in (14) and (15) to obtain the active and reactive powers, P and Q, respectively. Finally, the distortion power D is evaluated in (12). This results in three vectors $\boldsymbol{P}$, $\boldsymbol{Q}$, *and* $\boldsymbol{D}$ with length w. These are the incomes for the PQD-PCA classification

$$S^2 = P^2 + Q^2 + D^2 \tag{12}$$

$$S = V_{\text{rms}} * I_{\text{rms}} \tag{13}$$

$$P = V_{\text{rms}} * I_{\text{rms},1} * \cos(\varphi) \tag{14}$$

$$Q = V_{\text{rms}} * I_{\text{rms},1} * \text{sen}(\varphi) \tag{15}$$

$$I_{\text{rms}} = \sqrt{\frac{1}{T}\int_0^T i\,(t)^2\,dt}. \tag{16}$$

2) PQD-PCA Classification Algorithm: The goal is to convert a set of observations for correlated variables into a set of values for linearly uncorrelated variables, called principal components. This is carried out by performing an orthogonal transformation. The first principal component has the most possible variance, whereas every following component is optimized to achieve the maximum allowed variance, always orthogonal to the previous components.

As has been already described, we locate the occurrence of events over the time (9). Each event corresponds to a change on a certain appliance and, to identify that appliance, powers P, Q, and D have to be evaluated just for that one (12)–(16). This set of observation *PQD* within the windowed current $\mathbf{I_e}$ (11) provides significant information about the appliance type. Thus, before and after the event, the obtained powers *PQD* are steady and characteristic for the several operation modes that cause them. Moreover, within a few cycles around the event occurrence, there is the transient state, such as startups, that are strongly influenced by the circuit configuration of every appliance. The features of these transient states are less susceptible of overlapping [21]. In contrast to other event-based energy disaggregation algorithms focused on either steady or transient states (see Section I), the proposal described here pursues to classify events using as signatures both steady

and transient states. PCA transforms these tree correlated variables *PQD* into noncorrelated ones, which simplify the classification described below. This classification process was introduced for multisensory ultrasonic localization in [37].

Following the diagram in Fig. 3, the PQD-PCA algorithm needs to be previously trained with *PQD* instances of those appliance classes susceptible to be detected. Therefore, for a certain appliance, let us consider a training set of M samples, where every sample m is a w-D vector of current measures $\mathbf{I_e}$, as described in (16). Let us also consider the following matrix as the evaluated powers P, Q, and D in (12)–(16) for a certain sample m (indicated as a subindex in the following expressions):

$$[\mathbf{P} \quad \mathbf{Q} \quad \mathbf{D}]_\mathbf{m} = \begin{bmatrix} P_0 & Q_0 & D_0 \\ \vdots & \vdots & \vdots \\ P_{w-1} & Q_{w-1} & D_{w-1} \end{bmatrix}. \tag{17}$$

The matrix above can be rearranged as a $3w \times 1$ column vector $\boldsymbol{\tau}$ as follows:

$$\boldsymbol{\tau_m} = \{\mathbf{P}, \mathbf{Q}, \mathbf{D}\}^\mathbf{T} \tag{18}$$

Then, the averaged vector $\boldsymbol{\Psi}$ for the set of training can be evaluated as

$$\boldsymbol{\Psi} = \frac{1}{M} \sum_{m=0}^{M-1} \boldsymbol{\tau}_\mathbf{m} \tag{19}$$

where $\Psi \in \Re^{3w}$. Subtracting the averaged vector $\boldsymbol{\Psi}$ from the training set $\boldsymbol{\tau}_\mathbf{m}$, only the variance $\boldsymbol{\Phi}$ remains

$$\boldsymbol{\Phi}_\mathbf{m} = \boldsymbol{\tau}_\mathbf{m} - \boldsymbol{\Psi} \quad \forall m = 0, 1, \ldots, M-1. \tag{20}$$

And the scatter matrix $\mathbf{S_T}$ is evaluated as

$$\mathbf{S_T} = \sum_{m=0}^{M-1} \boldsymbol{\Phi}_\mathbf{m} \boldsymbol{\Phi}_\mathbf{m}^\mathbf{T}. \tag{21}$$

Considering a linear transformation where $3w$ column vectors of the training set above are mapped into a l-D space, such as $l < 3 \cdot w$, a new set of features $\boldsymbol{\Omega}_m \in \Re^l$ is obtained

$$\boldsymbol{\Omega}_\mathbf{m} = \mathbf{U}^\mathbf{T} \boldsymbol{\Phi}_\mathbf{m} \quad \forall m = 0, 1, \ldots, M-1 \tag{22}$$

where the matrix transformation $\mathbf{U} \in \Re^{3w \times l}$ has orthogonal columns.

If the scatter matrix $\mathbf{S_T}$ in (21) is considered for the training set (20), then, for the new feature vectors $\boldsymbol{\Omega}_\mathbf{m}$ expressed in (22), the new scatter matrix is $\mathbf{U^T S_T U}$. In PCA, this scatter matrix in the projected space should be chosen in order to maximize its determinant. This is done as follows (23), where $\{\mathbf{u_l} | l = 0, 1, \ldots, L\}$ are the set of the l largest eigenvectors from $\mathbf{S_T}$. Thus, the vector $\mathbf{u_1}$ is the first principal component and it has the largest variance

$$\mathbf{U_{opt}} = \arg\max_{U} |\mathbf{U^T S_T U}| = [\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_l}]. \tag{23}$$

So far, the training process for a certain appliance type has been described, resulting in the matrix $\mathbf{U_{opt}}$. Then, assuming a new sample $\boldsymbol{\tau_e} = [\mathbf{P\ Q\ D}]_\mathbf{m}$ arrives for classification (see Fig. 3), first, it is translated into the new feature space as

$$\boldsymbol{\Omega_e} = \mathbf{U_{opt}^T} \boldsymbol{\Phi_e} = \mathbf{U_{opt}^T} (\boldsymbol{\tau_e} - \boldsymbol{\Psi}) \tag{24}$$

where $\boldsymbol{\Psi}$ is the mean vector evaluated in (19). Then, the reconstruction matrix $\hat{\boldsymbol{\Phi}}_e$ is evaluated according to the following, using the inverse transformation:

$$\hat{\boldsymbol{\Phi}}_\boldsymbol{e} = (\hat{\tau}_e - \boldsymbol{\Psi}) = \mathbf{U_{opt}} \boldsymbol{\Omega_e}. \tag{25}$$

Since the linear transformation is $\mathbf{U_{opt}} \in \Re^{3w \times l}$, the k smallest eigenvalues (i.e., those with the smallest variance), where $k = 3 \cdot w - l$, are discarded. This results in energy losses during the reconstruction $\hat{\boldsymbol{\Phi}}_e$, as shown in

$$\varepsilon_e = \|\boldsymbol{\Phi_e} - \hat{\boldsymbol{\Phi}}_\boldsymbol{e}\|. \tag{26}$$

Consequently, if the arrival sample $\boldsymbol{\tau_e}$ has a similar variance to the training set $\boldsymbol{\Omega_m}$ in (22), then, the linear transformation $\mathbf{U_{opt}}$ arranges its variance on the l first eigenvalues leaving out just a small quantity of variance. As result, the reconstruction in (25) barely losses energy and the error reconstruction ε_i (26) should be small. Otherwise, a sample $\mathbf{e}$ belonging to a different appliance class does not have similar variance distribution, resulting in a bad rearranging after the linear transformation and larger losses, i.e., big error in the reconstruction (26).

With concern to the classification, the sample $\boldsymbol{\tau_e}$ can be classified into a certain appliance class if the reconstruction error is $\varepsilon_e < \gamma_a$, where γ_a is a certain heuristic threshold and a is a certain appliance class. Let us then consider a set of A appliance classes that have been trained to obtain their respective linear transformation as described before: $\mathbf{U_{opt,a}}, \forall a = 0, 1, 2, \ldots, A-1$. Thus, the reconstruction error ε_e for the sample $\boldsymbol{\tau_e}$ is evaluated for every appliance class a in order to obtain the corresponding errors $\varepsilon_{e,a}, \forall a = 0, 1, 2, \ldots, A-1$, and the sample $\mathbf{e}$ is classified as appliance type C, if it satisfies (27). Otherwise, this sample remains as unclassified (i.e., it does not belong to any of the trained classes).

The threshold γ_a is used to discard outliers, which are situations that do not correspond with any of the classes considered, and it can be set automatically. Hence, its value is high compared with the reconstruction error ε_e. Let us say the average error for the appliance class a during the training process is 0.5. Then, we set γ_a ten times higher (i.e., $\gamma_a = 5$). On the arrival of a new sample $\boldsymbol{\tau_e}$, if the minimum reconstruction error obtained from (27) is $\varepsilon_{e,a} = 8$, the sample $\boldsymbol{\tau_e}$ is likely to be an outlier and, consequently, it remains unlabeled. Therefore, once the training process is carried out [as performed in Section II-B2 with the Plug-Level Appliance Identification data set (PLAID) data set], the parameter γ_a is set automatically. Eventually, it could be updated with new correctly classified samples from new houses. As for the results presented in Section II-B2, no outlier has been found

$$C = \arg\min_{\forall \boldsymbol{a} \in \boldsymbol{A}} \ \varepsilon_{e,a} \text{ and } \varepsilon_{e,a} < \gamma_a. \tag{27}$$

III. Experimental Results

A. Data Sets

Two different data sets have been used to evaluate the performance of the proposal: 1) the BLUED (for the event detectors) and 2) the PLAID (for the classifier) data sets [19], [42].

This is due to the fact that they are independent processes and they require different scenarios to test their performance: high event density for the event detector, whereas heterogeneity about appliance classes for the classifier.

The BLUED data set contains voltage and current measurements from a single household in the United States during a week. It includes a large number of events (2845 events) from 50 different appliances and 30 different appliance types. Thus, this data set is very suitable for the analysis of the event detector thanks to its high event density. Nevertheless, there is a lack of heterogeneity in terms of learning features as only one household is monitored and, basically, there is only one instance per appliance class.

The PLAID consists of current and voltage snapshots of a few seconds (so it contains both steady and transient states) for 1074 instances from 11 different appliance types presented in 55 households. For instance, there are 176 instances of different laptops. This is a very convenient scenario for the PQD-PCA algorithm, since it can be trained with heterogeneous samples to achieve universal classifying models. However, there are not events available.

In terms of potential implementations, the BLUED and the PLAID data sets are sampled at 12 and 30 kHz, respectively, but the current transducers have an upper cutoff frequency of 300 Hz and 1 kHz, respectively, thus implying those bandwidths in processing.

In these data sets, the voltage and current transducers used imply an uncertainty of about 2%. Then, the algorithms have to deal with this level of uncertainty in the voltage and current measurements.

B. Considerations About the Computational Cost

The event detector is the most sensitive part in terms of a real-time implementation. A rough evaluation of the number of computations to be carried out can be extracted from Section II-A. The voltage $\boldsymbol{v}$ and current $\boldsymbol{i}$ in (1) and (2) may be acquired at 3 kHz (i.e., ten times higher than the bandwidth). Then the values $\boldsymbol{v_{rms}}$ and $\boldsymbol{i_{rms}}$ are calculated at a frequency of 60 Hz (i.e., the mains frequency in the United States) as in (6); every rms computation implies one multiplication and one addition per sample, and one square root and one division per cycle (60 Hz). The signal $\bar{\mathbf{i}}_{\mathbf{rms}}$ performs two multiplications and one division every mains cycle. The signal $\boldsymbol{\rho}$ requires a subtraction, multiplication, and comparison per every fundamental cycle. Finally, the value $\boldsymbol{\tau}$ comprises a comparison with a threshold and with its neighbors per fundamental cycle (60 Hz).

On the other hand, the PQD-PCA classifier algorithm operates only whenever an event has been detected; hence, it does not significantly affect the real-time performance of the system. It may be also acquired at 3 kHz (three times higher than the bandwidth) or higher, depending on the disaggregation accuracy that is needed. Its iteration is sparse and there is more time to process the signal. For instance, in the BLUED data set, 2845 events are detected in a week. Thus, the PQD-PCA classifier runs every 212 s approximately ((604 800/2845)((seconds per week)/(number of events))). Moreover, since it is an unsupervised method, the training process is carried out offline for all smart meters and it can be downloaded as a new firmware periodically. Therefore, this aspect does not affect the real-time computational cost either.

C. Event Detection Performance

This section presents the performance of the proposed event detection algorithm using BLUED data set. The performance is evaluated according to two metrics: 1) *true positive rate* (*TPR*) and 2) *false positive rate* (*FPR*) [17]. Furthermore, an ROC curve can be drawn out from pairs of *TPR* and *FPR* values to optimize the detector. *TPR* and *FPR* are defined as

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \in [0, 1] \tag{28}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \in [0, 1] \tag{29}$$

where TP is the number of true positives, FN is the number of false negatives (also known as missed), FP is the number of false positives, and TN is the number of true negatives. Hence, the best detector is such that TPR = 1 and FPR = 0. The optimal detector ψ_{opt} is such that minimizes the Euclidean distance as in (30). Each pair of settings for parameters θ_{th} and W_d gives a detector ψ. The set of all settings in the test gives the set of detectors Ψ

$$\psi_{\mathrm{opt}} = \arg \min_{\forall \boldsymbol{\psi} \in \boldsymbol{\Psi}} \|(0, 1) - (\mathrm{FPR}, \mathrm{TPR})\|^2. \tag{30}$$

The event detection algorithm detailed in Section II-A has been carried out several times for different configurations of θ_{th} and W_d, over the whole BLUED data set for both phases. Then, for every iteration, the *TPR* and the *FPR* as in (28) and (29) have been evaluated.

Due to the nature of the occurrence of events in a household (i.e., there are sparse over the day), the minimum distance W_d between peaks does not need to be optimized. It only needs to last more than transient states to avoid *FP*s coming from peak fluctuations. In the experimental results, it has been set $W_d = 5$ cycles, which guaranties the transient states have already occurred.

If $W_d = 5$ cycles, the variation of θ_{th} provides different pairs of *TPR* and *FPR* values, which can be represented in an ROC curve. This curve is presented in Fig. 6 for both phases A and B in BLUED data set, as well as for a random guess detector that draws a straight line from $(0, 0)$ to $(1, 1)$. Note that the optimal configuration would be the nearest one to the point $(0, 1)$ in the ROC curve (top-left corner in Fig. 6). The zoomed-in area gives more details about the performance for phase A in the area of interest. Table I shows the pair (*FPR*, *TPR*) achieved by an optimal configuration of the parameter θ_{th} according to (30), as well as the area under the curve considered as a quality factor. Although the performance is better in phase A, it is still high in B, where 88% of true positives are correctly detected with less than 12% of false positives. The outperforming in phase A can be explained in Table II, since there is a larger number of events in phase B (1578) than in phase A (907). Furthermore, there

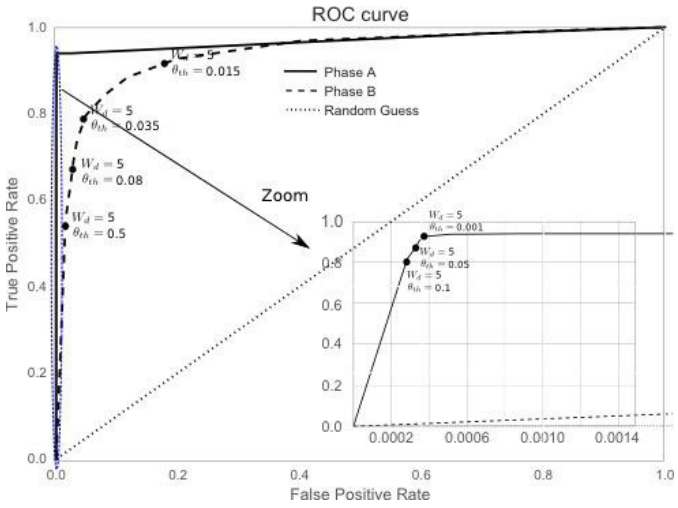

Fig. 6. Event-detection algorithm performance evaluated on an ROC curve.

TABLE I

EVENT-DETECTION ALGORITHM PERFORMANCE. METRICS: FALSE POSITIVE RATE AND TRUE POSITIVE RATE FOR THE OPTIMAL DETECTOR (FPR AND TPR) AND AREA UNDER CURVE

Metric	Phase A	Phase B
(FPR, TPR) *Optimal detector*	(0.88e-3, 0.94)	(0.12, 0.88)
AUC	0.97	0.94

TABLE II

OCCURRENCE OF EVENTS IN THE BLUED DATA SET PER PHASE AND APPLIANCE TYPE

Phase	Appliance	Events	Phase	Appliance	Events
A	Air Compressor	20	B	Basement Receiver/DVR/Blueray Player	34
	Backyard Lights	16		Basement lights	39
	Bathroom upstairs lights	98		Circuit 10	99
	Bedroom lights	19		Circuit 11	394
	Circuit 7	38		Circuit 4	46
	Hair Dryer	8		Circuit 9	46
	Kitchen Aid Chopper	16		Closet light	22
	Refrigerator	619		Computer 1	45
	Unknown appliances	67		Dining Room overhead light	32
	Washroom Light	6		Garage Door	24
				Hallway stairs light	58
				Iron	40
				Kitchen hallway light	6
				Kitchen overhead light	56
				LCD Monitor 1	77
				Laptop 1	14
				Living Room A/V System	8
				Living Room Desk Lamp	26
				Living Room Empty Socket	2
				Living Room Tall Desk Lamp	25
				Monitor 2	150
				Office lights	54
				Printer	150
				TV	54
				Unknown appliances	60
				Upstairs hallway light	17
	Total	907		Total	1578
TOTTAL (A+B)					2485

is also more appliances plugged to phase B than to phase A, and also those appliances are characterized by a smaller step change. All this makes phase B a more complex scenario for detecting events.

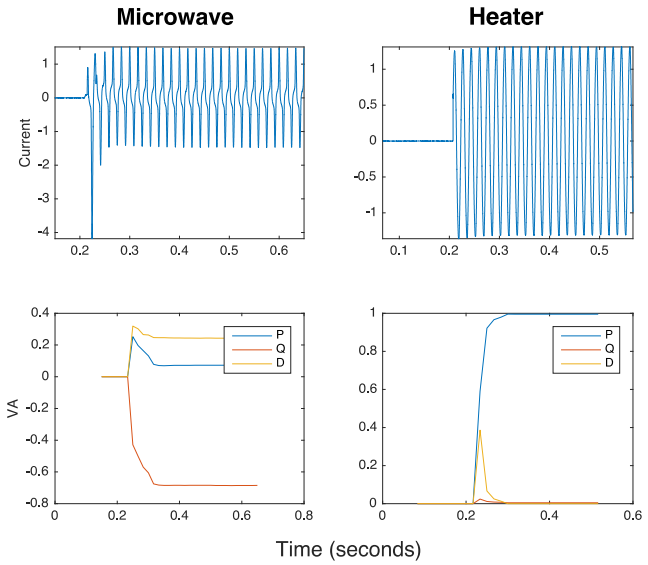

Fig. 7. Two instances from a microwave and a heater. (a) Current waveform for microwave. (b) PQD signatures for microwave. (c) Current waveform for heater. (d) PQD signatures for heater.

This optimization process, which is still required for each household, is much less complex than in previous works where a training process and a strong parameterization are involved.

D. PQD-PCA Algorithm Performance

In order to apply the procedure detailed in Section II-B to the PLAID data set, first, the evaluation of powers P, Q, and D for every single instance is required.

For instance, the signature extraction process for two appliances (a microwave and a heater) is shown in Fig. 7. The *PQD* power curves have valuable information: during transient states, they are unique for the circuit configuration, whereas during the steady states, capacitive loads (e.g., the microwave) show different power than resistive loads (e.g., the heater). Nevertheless, the *PQD* curves may be correlated with each other and this is why they need to be processed as described in Section II-B2, in order to rule out redundancies and compress the variance on the principal components.

The number of eigenvalues chosen for the inverse transformation of each appliance class keeps with the 97% of the variance. This results in about 3% energy losses for instances from the same appliance class and much more energy losses for instances from different classes.

Furthermore, to prevent the PQD-PCA algorithm from learning abrupt changes on the PQD signature curve due to errors in measurements, the evaluation of power through (12)–(16) is done every four cycles of the utility frequency with an overlapping of three cycles. Also, in order to avoid voltage drops, the normalization presented in the following is used:

$$P_i = \text{sgn}(P_i) * \frac{P_i^2}{S_i^2} \quad \forall i = 0, 1, \ldots, W-1 \tag{31}$$

where W is the number of samples for the signature curve P and S is the apparent power. This is also evaluated, in a similar manner, for Q and D.

TABLE III
METRICS

Appliance Class	Recall	Precision	F1	F0.5	Accuracy	Test instances	Training instances
Air conditioner	0.923	0.985	0.953	0.984	0.967	26	38
Compact Fluorescent Lamp	0.994	1	0.997	1	0.992	67	100
Fan	0.825	0.981	0.896	0.981	0.958	44	66
Hairdryer	0.815	0.996	0.896	0.995	0.959	61	91
Heater	0.477	1	0.646	0.997	0.971	13	20
Incandescent Light bulb	0.795	1	0.887	0.999	0.973	44	67
Laptop	0.931	1	0.964	0.999	0.986	68	102
Microwave "preheat"	0.984	1	0.992	1	0.996	49	74
Microwave "heat"	1	1	1	1	0.998	50	74
Vacuum Cleaner	0.7	0.934	0.8	0.934	0.986	15	23
Washing Machine	0.778	0.95	0.854	0.946	0.992	9	14
Average	***0.838***	***0.986***	***0.906***	***0.985***	***0.98***		
Total						***446***	***669***

The learning process has been carried out for all appliance types in the PLAID data set, except for the fridge that has been discarded due to the lack of instances (only 35). The microwave has been divided into "preheat" and "heat" classes. For every appliance class, the 60% of instances are used for training and the rest for test.

The performance is evaluated using metrics presented in Table III for each appliance class. Furthermore, to achieve more reliable results, a Monte Carlo cross validation has been performed where ten experiments have been conducted and the mean value of metrics in those experiments is presented in Table III. For every experiment, training instances have been selected randomly and metrics figured out. Likewise, overfitting problems during the training process are avoided. Finally, an average value for metrics along all appliances classes is presented. Also, it is worthy to remark that the "Test instances" column indicates only the number of instances left for testing in every appliance class. The classification performance has been computed for the whole test instance population (446).

Regarding metrics, recall, precision, F_1 score, and accuracy are commonly used to evaluate the performance of NILM algorithms and are defined in [26]. Recall (also known as sensibility) represents the ratio of positives detected between the total number of positives. For instance, the total number of Laptop instances is 67 and our detector is able to correctly detect and classify 63 of them (93.1%) discerning from a population of 446. Likewise, precision represents the ratio of true positive between false positive. Therefore, the precision on the example before is 100%, since no instance from another appliance class was classified as Laptop. Both recall and precision are sufficient to evaluate the performance of our detector. It can be seen that the proposed algorithm presents a high performance on both metrics with an average along all appliances of 83% and 98.6% for recall and precision, respectively. These two metrics are often represented as one on F_1 score, which is the harmonic mean between them. The average of the F_1 score appliance along all appliances is 90.6%. In general, it can be observed that appliances classes with more training instances are best detected than others. For instance, the F_1 score is 100% (ideal) for the microwave on "heat" mode (74 training instances), whereas the heater (20 training instances) draws a 64.7%. However, some appliances classes need less training examples than others because their singularity of their signatures overlaps less with other classes. In this way, the washing machine score is higher than the heater with less training instances: F_1 score is 85.4% with 14 training instances. Precision and recall often have the same importance on the performance; however, one might be more important that the other. Specifically, the use of an NILM algorithm for activity recognition applications, where the pattern usage needs to be learned, precision becomes more important than recall. Otherwise, having a high sensibility and a poor precision could lead to a bad pattern usage learning that feeds the model with many false positives. Therefore, we also present the F_β (32) score, which is the weighed mean between recall and precision, where $\beta = (w_{\text{recall}}/W_{\text{precision}})$ is the ratio between the weight of recall (w_{recall}) and the weight of precision ($w_{\text{precision}}$)

$$F_\beta = (1+\beta^2) * \frac{\text{precision} * \text{recall}}{\beta^2 * \text{precision} + \text{recall}}. \tag{32}$$

Thus, $\beta = 1$ is the harmonic mean. For instance, Table III shows $F_{0.5}$ score to illustrate the prominent importance of precision against recall on applications where the pattern usage needs to be learned. Here, the precision weights double and it can be seen now how the performance increases. Specially for the heater that has a recall of 47.7%, the $F_{0.5}$ score draws a performance of 99.7%. Although less relevant, the accuracy metrics (the ratio between true negatives and true positives between positives and negatives) is also evaluated that gives information about the true negatives. The averaged accuracy of this proposal is 98%. In general, it can be seen that the PCA-PQD algorithm performs quite well along all metrics, being slightly better on precision (98.6% on average). The best performance is for compact fluorescent lamps (98% in F_1 score); hence, this propose could be beneficial in smart lightning systems for large smart buildings where accuracy is required and the impact of wireless interference is intended to be reduced [43].

Although results are promising, there is a significant decrease in the classification performance for the heater regarding the recall metric. Hence, a new metric, commonly used in computer vision, has been employed: the cumulative ranking curve (CRC) or cumulative matching characteristics [44].

For instance, for a set of 11 different appliance classes, there will be 11 different CRC curves, each one corresponding to a class. The x-axis corresponds to the ranking, whereas the y-axis represents the correct classification rate, ranking one is equivalent to recall. Thus, having 100 instances from a certain appliance class, if it is obtained a ranking value of 3 and a classification rate of 0.5, it means that, for the 50 instances out of the 100, the correct appliance class was chosen as first, second, or third option. Thus, the ranking is the number of possible appliance options to select and the y-axis indicates

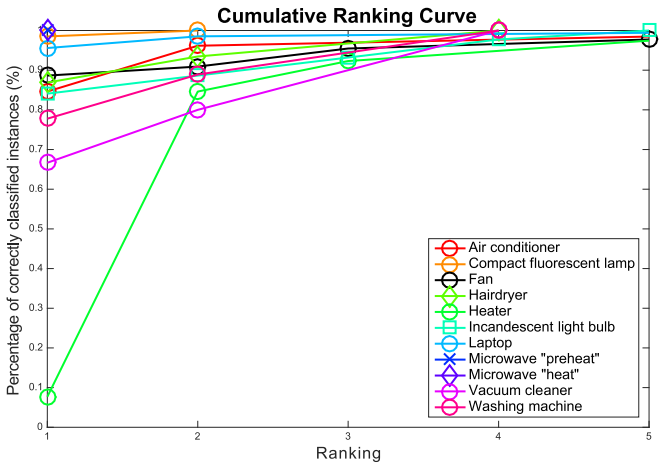

Fig. 8. Cumulative Ranking Curve for the classification.

the instance rate where the appliance class was considered as an option within the ranking interval. For instance, if a CRC curve has these sequence of values: $[(1, 0.5), (2, 0.7), (3, 0.8)$ $(4, 1)]$; this means that the appliance class was correctly classified 50% of cases (also known as recall); 70% was selected as first and second options; 80% as first, second, and third options; and, at the worst case, it was selected between the fourth best option. This gives an idea about how similar this class is with regard to other classes and the number of classes alike as well.

Fig. 8 presents the worst case scenario obtained from experiments. All the appliances are correctly classified with a higher rate than 65%, except for the heater whose ranking is 7.6%. Whether ranking 2 is considered, the classification rate for the heater significantly increases, reaching the 80%. After a manual review of the classification for the CRC, it is possible to observe that this other appliance class is the fan: 10 out of 13 heater test instances are classified as fan class as first option. This itself slightly affects the fan precision by decreasing its performance.

Finally, an extended comparative study based on the one in [45] has been carried out in Table IV, thus including the PCA reconstruction method proposed here using the *PQD* signatures. For that purpose, it is necessary to define the accuracy *ACC* as the total number of correct predictions divided by the total number of predictions

$$\text{ACC} = \frac{\text{number of correct predictions}}{\text{number of total predictions}}. \tag{33}$$

In [45], a thorough analysis of classifiers and load signatures is performed. It provides the performance for the best five classifiers depending on certain signatures: k-nearest neighbors, Gaussian Naïve Bayes, logistic regression classifier, decision tree, and random forest (RForest); where the RForest is the one achieving the best performance (ACC = 81.75%) using VI image signatures. Furthermore, combining all signatures, RForest achieves up to ACC = 86.03%. It can be observed that the proposal explained here, together with the *PQD* power trajectories, outperforms the best classifier (RForest), even when all signatures are combined, with an *ACC* of 88%.

TABLE IV
EXTENDED COMPARISON BASED ON [45] AND INCLUDING THE PQD-PCA PERFORMANCE

Signature	kNN(1)	GNB	LGC	DTree	RForest	Proposed PQD-PCA
Current	75.98 %	61.73 %	69.83 %	70.67 %	76.26 %	–
Real/Reactive	55.4%	27.19 %	29.14 %	49.07 %	51.58 %	–
Harmonics	45.25 %	18.72 %	30.45 %	42.18 %	49.63 %	–
Quantised	60.06 %	57.17 %	60.06 %	73.09 %	80.63 %	–
VI image	78.96 %	51.96 %	74.49 %	76.07 %	81.75 %	–
PCA Current	44.13 %	52.14 %	46.37 %	48.14 %	45.07 %	–
PCA Quantized	24.30 %	18.06 %	11.08 %	25.98 %	27.28 %	–
PCA IV	69.93 %	60.34 %	64.53 %	70.67 %	77.65 %	–
Combined	69.93 %	59.22 %	49.44 %	74.49 %	86.03 %	–
PQD power trajectories	–	–	–	–	–	**88%**

This high performance is due to the fact that the *PQD* power trajectories also contain information about the transient states and the proposed classifier successfully identifies the general signature of the appliance class.

IV. CONCLUSION

This paper presents a novel event-based NILM algorithm that has a high performance and scalability for activity monitoring. The event detector does not require a learning stage, only the set of one parameter, which simplifies the process and increases scalability. Furthermore, a thorough evaluation of the performance has been conducted obtaining suitable results, similar to those from previous works (94% of detected events with less than 1% of false positives).

The classification PQD-PCA algorithm presents the novelty of using the trajectories of power PQD as signatures and the compression of the PQD variables over the principal components to obtain generic appliance models. The PQD trajectories have proved to be a suitable choice, as they do not overlap among different appliances types. The training process can be carried out offline, using a data set, thus allowing the algorithm to be run for new households without any further training. The PQD-PCA algorithm has also been successfully evaluated according to standard metrics over the PLAID data set.

It is important to note that the performance of both algorithms (even detector and PQD-PCA classifier) has been obtained with an uncertainty in the voltage and current measurements of about 2% (according to the data sets).

REFERENCES

[1] D.-M. Han and J.-H. Lim, "Smart home energy management system using IEEE 802.15.4 and ZigBee," *IEEE Trans. Consum. Electron.*, vol. 56, no. 3, pp. 1403–1410, Aug. 2010.

[2] G. Gelle, M. Colas, and C. Serviere, "Blind source separation: A new pre-processing tool for rotating machines monitoring?" *IEEE Trans. Instrum. Meas.*, vol. 52, no. 3, pp. 790–795, Jun. 2003.

[3] G. W. Hart, "Nonintrusive appliance load monitoring," *Proc. IEEE*, vol. 80, no. 12, pp. 1870–1891, Dec. 1992.

[4] M. Sanduleac, M. Albu, J. Martins, M. D. Alacreu, and C. Stanescu, "Power quality assessment in LV networks using new smart meters design," in *Proc. 9th Int. Conf. Compat. Power Electron.*, Jun. 2015, pp. 106–112.

[5] V. V. Terzija, V. Stanojević, M. Popov, and L. van der Sluis, "Digital metering of power components according to IEEE standard 1459-2000 using the Newton-type algorithm," *IEEE Trans. Instrum. Meas.*, vol. 56, no. 6, pp. 2717–2724, Dec. 2007.

[6] S. K. Jain and S. N. Singh, "Fast harmonic estimation of stationary and time-varying signals using EA-AWNN," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 2, pp. 335–343, Feb. 2013.

[7] M. Music, N. Hasanspahic, A. Bosovic, D. Aganovic, and S. Avdakovic, "Upgrading smart meters as key components of integated power quality monitoring system," in *Proc. IEEE 16th Int. Conf. Environ. Elect. Eng. (EEEIC)*, Jun. 2016, pp. 1–6.

[8] F. Adamo, F. Attivissimo, G. Cavone, A. Di Nisio, and M. Spadavecchia, "Channel characterization of an open source energy meter," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 5, pp. 1106–1115, May 2014.

[9] R. Bonfigli, S. Squartini, M. Fagiani, and F. Piazza, "Unsupervised algorithms for non-intrusive load monitoring: An up-to-date overview," in *Proc. IEEE 15th Int. Conf. Environ. Elect. Eng. (EEEIC)*, Jun. 2015, pp. 1175–1180.

[10] J. Froehlich, E. Larson, S. Gupta, G. Cohn, M. Reynolds, and S. Patel, "Disaggregated end-use energy sensing for the smart grid," *IEEE Pervasive Comput.*, vol. 10, no. 1, pp. 28–39, Mar. 2011.

[11] O. Parson, S. Ghosh, M. Weal and A. Rogers, "Using hidden markov models for iterative non-intrusive appliance monitoring," in *Proc. 25th Annu. Conf. Neural Inf. Process. Syst. (NIPS) Workshop Mach. Learn. Sustain.*, Granada, Spain, 2011, pp. 1–4.

[12] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial HMMs with application to energy disaggregation," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2012, pp. 1472–1482.

[13] D. Egarter, V. P. Bhuvana, and W. Elmenreich, "PALDi: Online load disaggregation via particle filtering," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 2, pp. 467–477, Feb. 2015.

[14] J. Kelly and W. Knottenbelt, "Neural NILM: Deep neural networks applied to energy disaggregation," in *Proc. 2nd ACM Int. Conf. Embedded. Syst. Energy-Efficient Built Environ.-BuildSys*, vol. 15. 2015, pp. 55–64.

[15] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," *IEEE Trans. Consum. Electron.*, vol. 57, no. 1, pp. 76–84, Feb. 2011.

[16] S. Gupta, M. S. Reynolds, and S. N. Patel, "ElectriSense: Single-point sensing using EMI for electrical event detection and classification in the home," in *Proc. 12th ACM Int. Conf. Ubiquitous Comput.*, 2010, pp. 139–148.

[17] K. D. Anderson, M. E. Bergés, A. Ocneanu, D. Benitez, and J. M. F. Moura, "Event detection for non intrusive load monitoring," in *Proc. IECON (Ind. Electron. Conf.*, 2012, pp. 3312–3317.

[18] L. Farinaccio and R. Zmeureanu, "Using a pattern recognition approach to disaggregate the total electricity consumption in a house into the major end-uses," *Energy Buildings*, vol. 30, no. 3, pp. 245–259, 1999.

[19] K. Anderson, A. Ocneanu, D. Benitez, D. Carlson, A. Rowe and M. Berges, "BLUED: A fully labeled public dataset for event-based non-intrusive load monitoring research," in *Proc. 2nd KDD Workshop Data Mining Appl. Sustain. (SustKDD)*, Beijing, China, 2012, pp. 1–5.

[20] L. Pereira, F. Quintal, R. Gonçalves, and N. J. Nunes, "SustData: A public dataset for ICT4S electric energy research," in *Proc. Conf. ICT Sustain. (ICTS)*, 2014, pp. 359–368.

[21] J. M. Alcalá, J. Ureña, and Á. Hernández, "Event-based detector for non-intrusive load monitoring based on the Hilbert transform," in *Proc. Emerg. Technol. Factory Autom.*, Sep. 2014, pp. 14–17.

[22] S. R. Shaw and C. R. Laughman, "A Kalman-filter spectral envelope preprocessor," *IEEE Trans. Instrum. Meas.*, vol. 56, no. 5, pp. 2010–2017, Oct. 2007.

[23] T. Hassan, F. Javed, and N. Arshad, "An empirical investigation of V-I trajectory based load signatures for non-intrusive load monitoring," *IEEE Trans. Smart Grid*, vol. 5, no. 2, pp. 870–878, Mar. 2014.

[24] J. Liang, S. K. K. Ng, G. Kendall, and J. W. M. Cheng, "Load signature study—Part I: Basic concept, structure, and methodology," *IEEE Trans. Power Del.*, vol. 25, no. 2, pp. 551–560, Apr. 2010.

[25] M. Weiss, A. Helfenstein, F. Mattern, and T. Staake, "Leveraging smart meter data to recognize home appliances," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, Mar. 2012, pp. 190–197.

[26] D. Srinivasan, W. S. Ng, and A. C. Liew, "Neural-network-based signature recognition for harmonic source identification," *IEEE Trans. Power Del.*, vol. 21, no. 1, pp. 398–405, Jan. 2006.

[27] J. Liang, S. K. K. Ng, G. Kendall, and J. W. M. Cheng, "Load signature study—Part II: Disaggregation framework, simulation, and applications," *IEEE Trans. Power Del.*, vol. 25, no. 2, pp. 561–569, Apr. 2010.

[28] H. Song, G. Kalogridis, and Z. Fan, "Short paper: Time-dependent power load disaggregation with applications to daily activity monitoring," in *Proc. IEEE World Forum Internet Things (WF-IoT)*, Mar. 2014, pp. 183–184.

[29] F. Chen, J. Dai, B. Wang, S. Sahu, M. Naphade, and C.-T. Lu, "Activity analysis based on low sample rate smart meters," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining-(KDD)*, vol. 11. 2011, pp. 240–248.

[30] J. Alcalá, O. Parson, and A. Rogers, "Detecting anomalies in activities of daily living of elderly residents via energy disaggregation and Cox processes," in *Proc. 2nd ACM Int. Conf. Embedded Syst. Energy-Efficient Built Environ. BuildSys*, 2015, pp. 225–234.

[31] J. Alcalá, J. Ureña, and Á. Hernández, "Activity supervision tool using non-intrusive load monitoring systems," in *Proc. IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2015, pp. 1–4.

[32] N. Noury, M. Berenguer, H. Teyssier, M. J. Bouzid, and M. Giordani, "Building an index of activity of inhabitants from their activity on the residential electrical power line," *IEEE Trans. Inf. Technol. Biomed.*, vol. 15, no. 5, pp. 758–766, Sep. 2011.

[33] S. Rahimi, A. D. C. Chan, and R. A. Goubran, "Nonintrusive load monitoring of electrical devices in health smart homes," in *Proc. IEEE IMTC-Int. Instrum. Meas. Technol. Conf.*, May 2012, pp. 2313–2316.

[34] W. Kleiminger, C. Beckel, T. Staake, and S. Santini, "Occupancy detection from electricity consumption data," in *Proc. 5th ACM Workshop Embedded Syst. Energy-Efficient Buildings*, 2013, pp. 10:1–10:8.

[35] J. M. Alcalá, J. Ureña, Á. Hernández, and D. Gualda, "Assessing human activity in elderly people using non-intrusive load monitoring," *Sensors*, vol. 17, no. 2, p. 351, 2017.

[36] M. Turk and A. Pentland, "Eigenfaces for face detection/recognition," *J. Cognit. Neurosci.*, vol. 3, no. 1, pp. 1–11, 1991.

[37] J. A. Jimenez *et al.*, "Using PCA in time-of-flight vectors for reflector recognition and 3-D localization," *IEEE Trans. Robot.*, vol. 21, no. 5, pp. 909–924, Oct. 2005.

[38] Y. Xu, H. Wang, Z. Cui, F. Dong, and Y. Yan, "Separation of gas–liquid two-phase flow through independent component analysis," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 5, pp. 1294–1302, May 2010.

[39] R. Gutiérrez, C. Spagnol, J. J. García, L. Marnane, and E. Popovici, "Low complexity QRS detectors for performance and energy aware applications," in *Proc. IEEE-EMBS Int. Conf. Biomed. Health Inf. (BHI)*, Jun. 2014, pp. 256–259.

[40] M. Duarte, "Notes on scientific computing for biomechanics and motor control," in *GitHub Repository*. San Francisco, CA, USA: GitHub, 2015.

[41] A. Cole and A. Albicki, "Nonintrusive identification of electrical loads in a three-phase environment based on harmonic content," in *Proc. 17th IEEE Instrum. Meas. Technol. Conf.*, May 2000, vol. 1. no. 7, pp. 24–29.

[42] J. Gao, S. Giri, E. C. Kara, and M. Bergés, "PLAID: A public dataset of high-resoultion electrical appliance measurements for load identification research: Demo abstract," in *Proc. 1st ACM Conf. Embedded Syst. Energy-Efficient Buildings-BuildSys*, vol. 14. 2014, pp. 198–199.

[43] M. Li and H. J. Lin, "Design and implementation of smart home control systems based on wireless sensor networks and power line communications," *IEEE Trans. Ind. Electron.*, vol. 62, no. 7, pp. 4430–4442, Jul. 2015.

[44] V. Cevher, A. Sankaranarayanan, M. F. Duarte, D. Reddy, R. G. Baraniuk, and R. Chellappa, "Compressive sensing for background subtraction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Marseille, France, 2008, pp. 155–168.

[45] J. Gao, E. C. Kara, S. Giri, and M. Bergés, "A feasibility study of automated plug-load identification from high-frequency measurements," in *Proc. IEEE Global Conf. Signal Inf. Process. Global*, Dec. 2015, pp. 220–224.

José M. Alcalá received the B.S. degree in telecommunications engineering from the University of Granada, Granada, Spain, in 2009, and the M.Sc. degree in advanced electronics system and the Ph.D. degree in electronics from the Universidad de Alcalá, Madrid, Spain, in 2013 and 2017, respectively.

His current research interests include nonintrusive load monitoring, machine learning, big data, and signal processing.

Jesús Ureña (M'06–SM'15) received the B.S. degree in electronics engineering and the M.S. degree in telecommunications engineering from the Universidad Politécnica de Madrid, Madrid, Spain, in 1986 and 1992, respectively, and the Ph.D. degree in telecommunications from the Universidad de Alcalá, Madrid, in 1998.

Since 1986, he has been with the Department of Electronics, Universidad de Alcalá, where he is currently a Professor. He has collaborated in several educational and research projects in the area of electronic control and sensorial systems for mobile robots and wheelchairs and in the area of electronic distributed systems for railways. His current research interests include ultrasonic signal processing, local positioning systems and activity monitoring, and sensory systems for railway safety.

Álvaro Hernández (M'06–SM'15) received the Ph.D. degree from the Universidad de Alcalá, Madrid, Spain, and Blaise Pascal University, Clermont-Ferrand, France, in 2003.

He is currently an Associate Professor of Digital Systems and Electronic Design with the Electronics Department, Universidad de Alcalá. His current research interests include multisensor integration, electronic systems for mobile robots, and digital and embedded systems.

David Gualda received the B.S. degree in electronics systems and the M.Sc. degree in advanced electronics system from the Universidad de Alcalá, Madrid, Spain, in 2009 and 2011, respectively, and the Ph.D. in Electronics from the University of Alcala in 2016.

He is currently a Post-Doctoral Researcher with the Department of Electronics, Universidad de Alcalá. His current research interests include ultrasonic indoor location, signal processing, and information fusion.