POLITECNICO DI TORINO
Repository ISTITUZIONALE

A Cloud-based On-line Disaggregation Algorithm for Home Appliance Loads

Original
A Cloud-based On-line Disaggregation Algorithm for Home Appliance Loads / Mengistu, M.A., Abraha Girmay, A., Camarda, C., Acquaviva, A., Patti, E.. - In: IEEE TRANSACTIONS ON SMART GRID. - ISSN 1949-3053. - 10:3(2019), pp. 3430-3439. [10.1109/TSG.2018.2826844]

Availability:
This version is available at: 11583/2706278 since: 2019-05-06T11:47:24Z

Publisher:
IEEE

Published
DOI:10.1109/TSG.2018.2826844

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

09 November 2026

A Cloud-based On-line Disaggregation Algorithm for Home Appliance Loads

Million Abayneh Mengistu, Awet Abraha Girmay, Chirstian Camarda, Andrea Acquaviva and Edoardo Patti

***Abstract*—In this work, we address the problem of providing fast and on-line households appliance load detection in a non-intrusive way from aggregate electric energy consumption data. Enabling on-line load detection is a relevant research problem as it can unlock new grid services such as demand-side management and raises interactivity in energy awareness possibly leading to more green behaviours.**

To this purpose, we propose an On-line-NILM (Non-Intrusive Load Monitoring) machine learning algorithm combining two methodologies: i) Unsupervised event-based profiling and ii) Markov chain appliance load modelling. The event-based part performs event detection through contiguous and transient data segments, events clustering and matching. The resulting features are used to build household-specific appliance models from generic appliance models. Disaggregation is then performed on-line using an Additive Factorial Hidden Markov Model from the generated appliance model parameters. Our solution is implemented on the cloud and tested with public benchmark datasets. Accuracy results are presented and compared with literature solutions, showing that the proposed solution achieves on-line detection with comparable detection performance with respect to non on-line approaches.

***Index Terms*—Non-Intrusive Load Monitoring, load disaggregation, event detection, Hidden Markov Model, Smart Metering**

I. Introduction

Non-Intrusive Load Monitoring (NILM) is the process of identifying loads and their power consumption from an aggregate power using a disaggregation algorithm [1]. Residential and commercial buildings consume approximately 60% of worlds produced electricity [2]. Hence, energy saving in buildings will have significant impact on the reduction of overall energy demand. Research shows that real-time appliance specific feedback to households coupled with energy consumption awareness inspires green behavioral change and can save up to 12% of annual power consumption [3]. Furthermore, fast and on-line load detection can unlock new grid services for energy providers such as demand side management. In recent years, the interest of deploying novel smart meters [4] in residential buildings has grown extensively. Next generation smart meters will enable fine grain monitoring. Also, Internet-of-Things (IoT) technologies applied to electricity meters make these data available on the internet, so that they can be processed on the cloud using computational intensive machine learning approaches. Both IoT and cloud technologies will enable novel services unlocking new business models. To manage millions of smart meters in secure, reliable and scalable ways, utilities must include a distributed data-center. In this respect, cloud computing is envisaged to play a key role in designing future smart grids. As such, in the context of smart grid research, a new NILM solutions are raising as part of cloud-based energy data management IoT platforms. Thus, cloud systems are suitable to build generic appliance signature databases extracted across different homes ready to be shared for future applications and services.

Most NILM methodologies proposed in literature are based on approaches using Graph Signal Processing [5], [6], Hidden Markov Models [7], [8], [9], [10], [11] or deep neural networks [12], [13]. Based on the classification techniques they employ, NILM methods can be categorized as either event-based or non event-based [14], [2]. Depending on the data requirement for offline training to build a-priori class information, NILM can be supervised or unsupervised [14]. Due to the cost of having appliances data for training, recent focus is on achieving unsupervised methods and generating appliance models. Although NILM studies that have been conducted in recent years have advanced, the area still faces substantial challenges and limitations in its application, required training time, recognition accuracy and on-line deployment strategy within smart metering framework.

In this paper, we propose an On-line-NILM methodology by integrating an event-based appliance classification algorithm with a FHMM (Factorial Hidden Markov Model) methodology. The integration of the two approaches realizes a completely unsupervised, computationally efficient and scalable solution. The proposed solution provides fast and on-line appliance detection from 1Hz frequency aggregate electric consumption data of residential buildings. We implemented On-line-NILM in a smart metering framework [15]. Experiments conducted to characterize its detection performance and comparative results on public datasets show that the proposed solution achieves on-line recognition with similar or improved accuracy with respect to state-of-the-art solutions.

The rest of this paper is organized as follows. Section II reviews most relevant literature solutions. Section III presents the proposed On-line-NILM that leverages upon a smart metering infrastructure briefly highlighted in Section IV. Section V discusses the experimental results. Finally, Section VI offers our concluding remarks.

M.A. Mengistu, A. Acquaviva and E. Patti are with the Department of Control and Computer Engineering, Politecnico di Torino, Torino, Italy. Emails: {million.mengistu, andrea.acquaviva, edoardo.patti}@polito.it

A.A. Girmay and C. Camarda are with Midori s.r.l., Torino, Italy. Emails: {awet, christian}@midorisrl.eu

This work was partially supported by the EU project Flexmeter (Grant Agreement no. 646568)

II. Related Work

Event-based NILM systems typically are based on algorithms for detecting appliance events via edge detection followed by feature extraction, classification, appliance labelling, and energy estimation [16], [17], [5], [6], [18]. In [16], an edge detection using two thresholds for unit-time and consecutive time intervals is presented. The threshold values are set experimentally for the whole dataset to avoid false detection. On the other hand, event detection method using smoothed frequency components is proposed in [17]. They used Cepstrum smoothing to eliminate noise in aggregate power. GSP-based [5], [6] is a Graph Signal Processing (GSP) approach leveraging on piecewise smoothness of power load signal for appliance events. In GSP-based, two event-based graph techniques are discussed for low complexity classification of active power segments. For training, their method uses periods when only one appliance is running a full run from on to off.

In our proposed algorithm (hereafter referred to as On-line-NILM), we propose a completely blind event-based profiling of full day aggregate consumption sampled at 1 Hz. The method starts by identifying steady state and transient data points followed by feature extraction for active windows which are then clustered into their feature groups.

Hidden Markov Model (HMM) is machine learning algorithm for probabilistically modeling states data over a length of time [19], [20]. It has been applied widely by researchers for energy disaggregation because its structure makes it well suited for modeling appliance consumption. Factorial HMM (FHMM) is an extension of HMM where there are multiple independent Markov chains of hidden variables per number of states [21]. This property makes FHMM well suited in NILM for classification of multiple appliances. NILM-TK [22] is an open source toolkit for non-intrusive load monitoring designed specifically to enable comparison of energy disaggregation algorithms in a reproducible manner. It gives a complete pipeline from datasets to accuracy metrics. NILM-TK [22] includes benchmark implementations for the Combinatorial Optimization and FHMM disaggregation algorithms. The authors provided one of the first comparison between multiple approaches across multiple publicly available datasets. In [7], Kolter et al. used unsupervised learning procedure for devices with relatively short time scales. Their focus was on the inference problem and applied a combination of two FHMM variants (additive and difference FHMM). They demonstrated that the additive Factorial HMM model has large representation power. However, its applicability has likely been greatly reduced by the difficulty of inference in models with a substantial number of HMMs. SparseNILM [8] is an on-line NILM method that works on an embedded processor for low sampling rates. In their approach, authors modeled aggregate consumption using FHMM. To preserve the dependencies between loads, they proposed a super state Viterbi algorithm. Their methodology analyzes sub-metered data from load priors and creates probability mass function for each appliance. Kong et al. [10] present an extensible supervised approach exploiting HMMs to model appliances. This solutions uses Segment Integer Quadratic Constraint Programming (SIQCP) to disaggregate power. Authors provide a framework for their approach to work within smart meter infrastructure. Neural-NILM [12] is a deep learning implementation of NILM. With Neural-NILM, authors demonstrated three neural network architectures with supervised training in known houses. The methodology performs very well on a house, unseen during training, but as any deep learning approach, requires a large training-set with high computational complexity. In [12], authors provide detailed accuracy test results using public data sets.

Approach	Unsupervised	Methodology	On-line	Cloud-based
On-line-NILM	✓	Events + HMM	✓	✓
GSP-Based [5], [6]	✓	Events	✓	×
NILM-TK [22]	×	FHMM	×	×
Kolter et al. [7]	✓	HMM	×	×
SparseNILM [8]	×	Sparse HMM	✓	×
SIQCP [10]	×	HMM	×	✓
Neural-NILM [12]	×	Deep Neural Network	×	✓

TABLE I: Summary of Our On-line-NILM approach and other literature solutions

In our cloud based On-line-NILM, we use event-based appliance detection integrated with HMM to model appliances and perform inferences. Once On-line-NILM completes the device signature extraction, it works in event less mode. This allows our on-line solution to be engaged on short duration of aggregate input data neglecting the need to have every time all information about rising and falling events of appliances (which are requirements for clustering, matching steps of completely event-based algorithms). In addition, we implemented our own new and robust event detector which is capable to perform well with noisy aggregate input and fluctuations that detects well events even with growth and decay trends. These are the main novelties of our On-line-NILM's event detection strategy compared with other event-based approaches, such as GSP-Based [5], [6]. Unlike previous works based on HMM, such as [8] and [10], our solution does not rely on sub-metered data or on prior knowledge to generate appliance models. Instead, On-line-NILM summarizes the results of the event-based detection to build specific appliance models using HMM. Then for inference, it exploits these appliance models to generate aggregate consumption models dynamically at run-time for short time-windows using FHMM. Our method promises to solve the scalability limitation highlighted in SparseNILM [8] through an automatic appliance model generation for new devices by enabling unsupervised event-based appliance detection. Table I reports the main features of the related literature solutions. The limitation of presented solutions consists of missing a sustainable unsupervised approach for automatically modelling appliances and realizing an on-line NILM for smart metering infrastructures. As such, our contribution with respect to literature, as detailed in Table I, includes the following: i) a completely blind event-based appliance signature extraction methodology that makes use of reactive power as an additional feature; ii) a new appliance modelling technique based on results of the event-based methodology (this technique neglects sub-metered data or prior knowledge from households); iii) generating minimized aggregate power consumption model dynamically at run time by using modelled

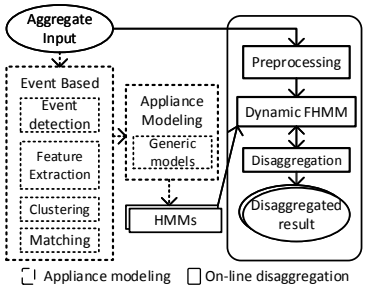

Fig. 1: Architecture of the proposed On-line-NILM.

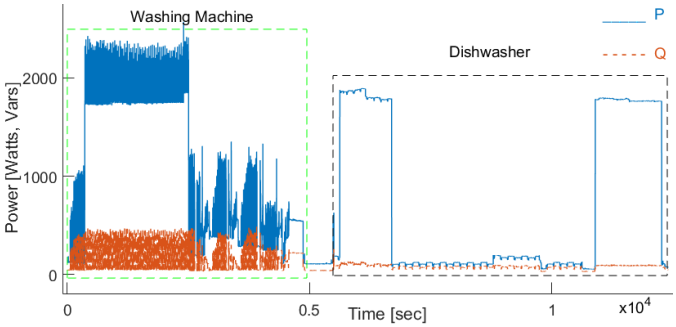

Fig. 2: Real and reactive power trends for a washing machine and a dishwasher

appliances for short disaggregation time-windows; iv) on-line disaggregation for short time-windows (e.g. from 15 to 60 minutes) feasible for near-real-time feedback to end-users; v) a cloud based implementation of our NILM algorithm within a smart metering infrastructure for generic smart-meter data; vi) a scalable design to easily include new appliances.

III. Methodology

We now introduce the proposed On-line-NILM system. The basic modules of the proposed system are shown in the architecture diagram in Figure 1. The approach starts from *Event Based* block performing load profiling of one-day aggregate consumption through event-based profiling method. The inputs to this block are real and reactive power consumption sampled at 1 Hz by a smart meter deployed at home premises. For this purpose, we are also developing our own smart meter (whose presentation is beyond the scope of this paper). However, the proposed On-line-NILM can work also with third-party devices. Next, the results of the event-detection (classified appliances and features associated with them) are passed to the *Appliance modeling* block, where appliances are modeled using *HMMs* specific to appliances and households. These parameters are then exported to data-store. Events, that are matched but not associated with particular appliance, are stored separately to be labeled manually by consulting with residents. After running event-based disaggregation method with appliance modeling for few days until high-power appliances have been modeled, the event detection block is discontinued. Then the on-line disaggregation procedure starts by taking aggregate data sampled at 1 Hz in chunks of short time windows (window lengths of 15 to 60 minutes). The *Preprocessing* block loads the input data and prepares it for on-line disaggregation. The prepared data is fed to a *Dynamic FHMM* block where an aggregate consumption is modeled for the passed data. The *Disaggregation* block performs inference on the input using the generated aggregate model. The result of the inference is fed back to the dynamic appliance modeling to improve the estimation accuracy of appliances in the subsequent time windows. Finally, the *Disaggregated results* are exported to data-store to estimate energy consumption and generate accuracy metrics. The following sub-sections describe in-depth each block in our architecture.

A. Event-Based Non-Intrusive Detection

The proposed Event-Based NILM system relies mainly on the detection and classification of events within the aggregate electrical signal. In the proposed solution, we implemented our own robust event detector that is capable to perform well with noisy fluctuations and detect events with growth and decay trends which is a bottleneck and inherent problem to other event detectors. This event detector enhances our previous work [18]. The *Event-based* block depicted in the block diagram of the proposed approach (see Figure 1) entails *Event Detection*, *Feature Extraction*, *Event Clustering* and *Events Matching* stages.

It is worth highlighting that in event-based frameworks in literature accuracy are greatly dependent on the power features used. This is also for our *Event Detection*, *Event Clustering* and *Events Matching* modules. With more features, appliances model is more precise because some appliances can have similar active power curves but completely different reactive power trends (especially for non-linear loads). Hence, an advantage of using the reactive power is to better distinguish characteristic features of devices. For example, Figure 2 shows the power trends of a Washing Machine and a Dishwasher. The heating elements of the Dishwasher has a real power trend comparable with that of the Washing Machine. However, Washing Machine has very frequent and high variation oscillations of reactive power due to heating and spin cycles. Whereas, the power trend of the Dishwasher shows little or no significant variation in reactive power. Therefore, using reactive power enables good cluster separation and better pairing during events matching. However, if information on reactive power is not available, our methodology still works and it is still able to build the appliance models.

1) Event Detection: This block detects transient states and start-up features of appliances from the aggregated power signal that are used to extract further signatures. Real and reactive power, P(t) and Q(t) respectively, sampled at 1 Hz are fed to the inputs. The power signals are transformed into contiguous and transient data segments and obtain the time of occurrence for each transition interval as follows:

i) Contiguous level detection: In this stage, P and Q are first filtered using median filtering. The filtering process is intended to remove noise like spiky events from the aggregated signal but it is able to keep relevant short duration events. First, we

compute the difference, $\Delta p/\Delta t$, of the aggregate consumption denoted by δX in order to segment the signal into steady and transient state sections. Given a time series of both aggregate real and reactive power consumption $X(1:T)$ of length T, δX is obtained as shown in Equation 1.

$$\delta X(t) = \{X(t+1) - X(t), \quad \text{for } t = 1:T-1 \tag{1}$$

Then, we label the data points in $\delta X(t)$ as contiguous and transient sections based on a specified threshold. For example, events can be change in average active power exceeding a certain threshold able to capture significant events. In the following, the binary vector a(t) defines the labelling as:

$$a(t) = \begin{cases} 1, & \text{if } -P_{Thr} \leq \delta X(t) \leq P_{Thr} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

In Equation 2, 1 indicates the steady state portion of the time series data, while 0 corresponds to transient portions due to rising and falling edges. The contiguous level detector obtains the starting and ending time of each steady state power segments.

ii) Active power segment labeling: This stage will label the filtered P and Q components of the power data segments obtained as either active power segments or background power. A reading is referred to background power when either standby or permanently on devices are operating and there are no noticeable changes in power levels due to devices activation.

iii) Active window detection: Once the steady state segments are labeled, the start and end time of active window periods are obtained by looking for portion of aggregate signal surrounded by two background levels. Figure 3 shows a plot of aggregate power consumption of one day where active windows of a single appliance are represented by an active segment between two background segments. The power activities outside the highlighted boxes indicate power segments labeled as no device activities. Whereas, the enclosed power variations correspond to active windows where the power curve belong to appliances operation. The background power segments are shown marked by a horizontal dashed line, whereas the boxes highlighted in gray indicate the active windows segments comprising either multiple or single appliance activities.

2) Feature Extraction: In this stage, once active windows are marked features are extracted, from the transient portion of original unfiltered signal. Relevant features are: power change $\Delta(P,Q)$ and amplitude features of the transient spikes on both real and reactive power signals. The amplitude features are determined by estimating the positive and negative peak power amplitude, spike width, gradient, peak to peak amplitude and peak amplitude to $\Delta(P,Q)$ ratios.

3) Event Clustering: In the clustering stage, events are grouped into separate clusters according to their extracted features. In this section, a non-parametric clustering algorithm, mean-shift clustering [23], [24], is used because we do not have knowledge of number of existing appliances in advance. The great benefit of the mean-shift clustering algorithm is that it is non-parametric which is independent of the underlying

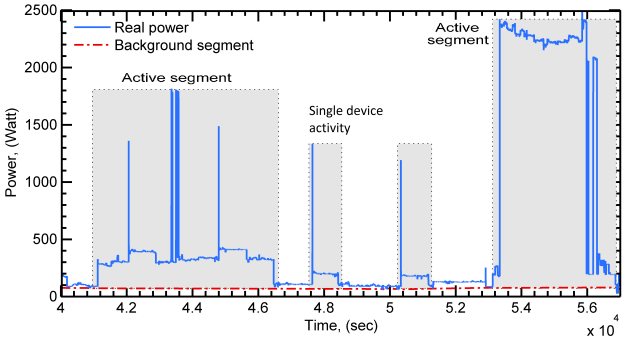

Fig. 3: Detected windows of single or multiple appliance activities and background power levels

distribution and entails a mode-seeking algorithm [25]. We utilize a simple kernel function such as:

$$K(\theta) = \begin{cases} 1, & \text{if } ||\theta|| \leq \lambda \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where λ is the kernel bandwidth.

4) Events Matching: In the matching process, all rising and falling events are checked for matching pairs so as to infer the usage interval of each appliance. The matching process in the proposed system is based on the background level detection and active window approach. Hence, this processing stage is performed on each identified active window times. It supports self-learning in operation to create a pool of possible appliance power states and build update-able appliance database.

B. Appliance Modeling

This section presents how we construct models for appliances consumption using HMM from the output of the event detection. The resulting models are generated specifically for each household and will be used to generate aggregate consumption model using the Factorial HMM.

During their operation, appliances usually go through their operational modes/states (e.g. ON, Standby, Heat, Spin, OFF depending on the appliance). These modes are observable through the appliances power demand. To describe appliance consumption sequence as a process with HMM, the two crucial steps are determining the number of possible states with interconnection between them and parameter estimation which is estimating transition and emission probabilities. Considering a particular appliance (a) the sequence of transition between its operation modes for a length of N time steps can be represented by $Z^a = \{z^a(1), ..., z^a(N)\}$ and taking into account appliance consumption real power component P $\in \mathbb{R}$. Its continuous observed power demand is $P^a = \{p^a(1), ..., p^a(N)\}$. The value of each discrete state variables corresponds to one of the states of the appliance and are hidden while the power values are the emissions of an HMM. The parameter governing a complete HMM is a three parameter vector represented by $\theta^a = \{\pi^a, A^a, B^a\}$. Where π, is the probability of the starting state of the hidden

variable at first time step t=1. A is transition probabilities between consecutive states. B is the emission distribution of the hidden states for power values (P) from the states. The emission distributions are commonly described to be Gaussian distributed in different states, given by $B_i = \{\mu_i, \sigma_i^2\}$ where μ and σ are the mean and variance of the Gaussian distribution describing the appliances power at state (z). From Section III-A, we extracted: i) The duration of appliance operation and individual states ($T_{On}, T_{Off}, T_{Heat}, T_{Spin}, etc.$); ii) The consumption of each operational states in the first step (i) ${P_{state}}^a$; iii) Appliance linearity based on reactive power (resistive & non-resistive); iv) Appliance category based on frequency of usage (not frequent, high frequent, medium frequent, low frequent, and always on). Features (i) and (ii) are used to decide the HMM parameter vector values (θ). Whilst, (iii) and (iv) are stored together with the HMM of appliances and are used to optimize aggregate consumption modeling. We start from generic models of appliances with priors for each parameter in appliance's mode as in [9] and [26]. The priors contain appliance type labels with state transition matrix between states and expected values of an appliances emission in terms of mean and variance from domain knowledge. Hence, the modeling process goes from parameters of an appliance initialization models, to specific appliance instance per household per appliance. For all labeled and unlabeled appliances discovered by event-detection within household, we construct consumption and states sequence using the power levels and operational states. Then by employing K-means clustering [27] matching the number of states in generic model and taking the average of returned values, we obtain mean consumption for each state in the real devices. We applied the most widely used Expectation-Maximization algorithm for HMM parameter estimation from the pre-determined hidden states and power demand. We generate instances of this models by replacing actual values for states of the appliance with its mean consumption. This will allow the modeling of appliances with specific characteristics to all households we have to do disaggregation.

The learning approach in our methodology is different from other unsupervised training approaches like [7] and [9]. In [7], HMMs of individual appliance signatures extracted from aggregate data are manually labeled. In [9], a-priori models of general appliances are tuned using signatures extracted from the aggregate exploiting the periods during which a single appliance turns ON/OFF. We obtain appliance labels and individual signatures from unsupervised event detection. Instead of using generalized appliance model, we store customized models with specific characteristics for each household. This builds unique appliance signature model database for households. This is an effort to solves the challenge in difficulty of forming a generic appliance model due to high variability of features within appliances [28].

C. On-line disaggregation

1) Pre-processing: This is the starting point of the on-line disaggregation. It takes as input active power readings of aggregate consumption sampled at 1 Hz to 1 minute. The data length is decided by disaggregation window setting (10 to 60 minutes). We employ median filter for 1Hz inputs to smooth the signal by removing spikes and outliers. Then we estimate the local-background power based on local-minimum

2) Dynamic FHMM for Aggregate consumption modeling: Before combining HMMs to generate aggregate model for households, we propose an algorithm that estimates the likelihood of appliance activation within the disaggregation time window. This is because, when using FHMM for inference, the accuracy decreases when the number of involved HMM models increases [7], [11]. To combine resulting HMMs and generate an aggregate power consumption model, we adopted the additive Factorial HMM (FHMM) [7]. The algorithm to dynamically generate aggregate consumption is shown in the listing 1. Its inputs (see line 1 in listing 1) are: i) aggregate readings in time window (AG); ii) list of all HMM models in household i.e. HMM parameters with minimum and maximum activation duration (Appliances); iii) estimated background power of the household (BG) and iv) the time-step of the aggregate (STP). First, maximum consumption is marked for the aggregate input passed using background power (see line 2). Then using mean activation duration from the appliance model, app.minDuration, app.maxDuration and the time-step of aggregate, possible continuous activation of appliance is marked to estimate numSamplesMin and numSamplesMax (see lines 4 - 6). Finally, for the appliances that match the duration and mean consumption, their HMM parameters are used to form aggregate power model using GaussianHmm probability density function (see lines 7 - 9).

Algorithm 1 Dynamic Fhmm building

1: **procedure** BUILDMODEL(AG, $Appliances$, BG, STP)
2: $Max \leftarrow max(AG) - BG$ ▷ local maximum peak consumption
3: **for** $app\ in\ Appliances$ **do**
4: $numSamplesMin \leftarrow app.minDuration * STP * 60$
5: $numSamplesMax \leftarrow app.maxDuration * STP * 60$
▷ group aggregate into continuous ON and OFF using mean power as threshold
6: $onDuration \leftarrow \{max\ number\ of\ ON\ samples\}$
▷ If appliance on duration is within duration include model in aggregate fhmm
7: **if** $Max \geq app.mean$ & $duration[app] \geq numSamplesMin \leq numSamplesMax$ **then**
8: $aggregate.add(app)$
9: $model \leftarrow GaussianHmm(app)$
10: **return** $model$

3) Disaggregation: Aggregate model (from Section III-C2) applied to smart meter power reading decodes appliance states and power emission for the input sequence. Using duration property from the appliance model in the household, we post-process decoded results by removing short spike-like inferences and merging adjacent events. However, in performing on-line disaggregation for short duration, it is evident that some appliances could be active for duration longer than the length of disaggregation window (such as washing machines, dishwashers and lights). Figure 4 shows an example of washing machine and a dishwasher operation patterns with respect to a possible disaggregation window (set at 30 minutes). The washing machine in Figure 4 takes at least two time windows to fully capture its operation cycle (indicated as heater followed by frequent spin cycles). Similarly, the dishwasher in Figure 4 needs about four disaggregation slots to completely detect its operation pattern states (indicated as

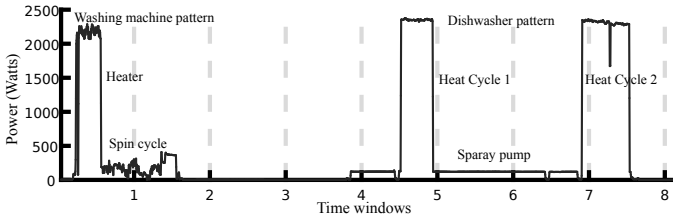

Fig. 4: Washing machine and Dishwasher operation patterns

heat cycle 1, followed by spray-pump and finally heat cycle 2). To keep the distinctive features of long lasting appliances within the visibility of disaggregation windows, we propose to exploit a sliding memory of partially identified appliances with signature patterns across disaggregation windows. For instance, in the case that the heater of a dishwasher is confused with heating element of a washing machine or other device (see Figure 4), it can be corrected with the time windows that follows. In addition, the intermediate states (Spin and Spray pump of washing machine and dishwasher respectively) usually comes following the heating elements and this unique feature enables to include their intermediate states in the aggregate model. Finally, to scale the on-line methodology for new appliances introduced in households, appliance model construction through event detection needs to be activated by enabling the *event-detection* block (discussed in Section 4).

IV. Flexmeter Architecture

In realizing a successful on-line NILM software, it is essential to evaluate it with real-world data in a practical platform. For this reason, we designed our On-line-NILM solution as a service leveraging upon Flexmeter [29], [30]. Flexmeter is our proposed cloud-based Smart Metering architecture to foster general purpose services in the smart grid scenario. It follows both multi-tenant and microservice design patterns. In brief, Flexmeter provides: i) an hardware independent interoperability among heterogeneous devices and novel smart meters (e.g. sampling at 1 Hz); ii) fast bidirectional communication; iii) (near-) real-time data transmission and data collection; iv) Web Services and API (Application Programming Interface) to access data using open and web-oriented standard data-formats. Hence, thanks to Flexmeter's Web Services and API, the proposed On-line-NILM will retrieve the needed data from smart meters deployed at customers premises. In this context, the On-line-NILM is one of the services enabled by Flexmeter. Example of other services are: i) State Estimation, ii) Network Reconfiguration, iii) User Awareness and iv) Demand Response.

V. Experimental Results

In this section, we present experiments conducted by using the proposed approach to support the validity of our claims discussed in Section III. We present accuracy estimations for load classification and energy consumption using three low-frequency public datasets. We tested the proposed On-line-NILM using the Flexmeter architecture (see Section IV). The public datasets we used are: the Reference Energy Disaggregation Dataset (REDD) [31] , UK Domestic Appliance-Level Electricity (UK-DALE) [32] and the REFIT Electrical Load Measurements dataset [33]. After a brief introduction to the datasets and the accuracy metrics, we show the estimation on time analysis for on-line disaggregation, the evaluation of accuracy and the comparison of our results with literature solutions.

The REDD dataset contains low-frequency power data for 6 US houses including the mains and individual labelled circuits in the houses. Plug-level monitors in REDD took measurements at a frequency of 1/3 Hz. To match these readings with the mains which is sampled at 1 Hz, we used 4 seconds interval on both the main and plug levels for on-line disaggregation. This dataset is most widely used by NILM researchers to evaluate varies approaches. The UK-DALE dataset has power demand records from 5 UK houses where in each house the mains are sampled at 1 Hz and the appliances data samples are taken at every 6 seconds. We down sampled real power measurements of the main to match with sub-metered data. The REFIT dataset is collected at 8 second resolution from 20 houses in UK. As highlighted in [6], REFIT reports various unknown appliances that make challenging the disaggregation.

From each dataset, we choose a period of two weeks of data for appliance model generation using the *Event Detection* block (see Section III-A). To evaluate the On-line-NILM performance and the generalizability of appliance models in the different seasons, we disaggregated a one-year worth of data of House-1 in UK-DALE. To compare our results with Neural-NILM [12] and NILM-TK [22], we disaggregated all houses in UK-DALE taking one week worth of consecutive data. Whilst, to compare our results with GSP-Based [6], we disaggregated House-1, House-2 and House-6 from REDD and House-8 from REFIT. In this case, we disaggregated two weeks worth of consecutive data. It can be noticed that we chose the same houses and the same time periods to obtain a fair comparison with other literature solutions. To obtain a consistent comparison, the time periods are the same as in [12], [22], [6].

Table II presents the device modeling parameters constructed from the results of the proposed event-detection procedure described in Section III-A applying it on the UK-DALE dataset House-1. Table IIa shows the number of states of appliances, mean power emissions for these states (the states are OFF, Intermediate and ON represented as S1, S2 and S3 respectively), the minimum and maximum ON activation duration in minutes. Table IIb reports the state transition probability matrix for appliances in Table IIa (e.g. for the washing machine estimated probability of staying in first state (S1/ON) is 0.8 as reported in the first entry; and the transition probability from S1 to S2 is 0.1). These parameters in Tables IIa and IIb are loaded by the on-line algorithm to dynamically build aggregate model.

A. Accuracy metrics

We adopted classification metrics shown in Equations (4) - (7) based on True Positive (TP), True Negative (TN), False

Appliance	States	Emission/State			Duration (min)	
		S1	S2	S3	Min	Max
Washing Machine	3	0	1850	200	10	-
Microwave	2	0	1550	-	0	2
Refrigerator	2	0	88	-	6	-
Kettle	2	0	2500	-	1	4
Dishwasher	3	0	123	2300	6	20

(a) Appliance states

	Washing machine			Microwave		Fridge		Kettle		Dishwasher		
	S1	S2	S3	S1	S2	S1	S2	S1	S2	S1	S2	S3
S1	0.80	0.10	0.10	0.95	0.05	0.95	0.05	0.95	0.05	0.88	0.12	0.00
S2	0.00	1.00	0.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	0.9	0.1
S3	0.10	0.00	0.90	-	-	-	-	-	-	0.00	0.00	1

(b) State transition parameters

TABLE II: Appliance model parameter values

Positive (FP) and False Negative (FN). These metrics analyze how well the algorithm can identify appliance switching ON or OFF. TP refers to the number of times a device is correctly identified as ON, while TN is the count of correctly captured OFF events. In the contrary FP stands for the case when an ON state is reported while the appliance is not consuming power, while FN is the count of incorrectly assigned OFF events.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \tag{7}$$

The precision in Equation (5) denotes the ratio of TP in the universe of all the examples assigned as positive while the recall in Equation (6) is the ratio of TP in the universe of all positive examples in the dataset. F_1-score is a measure of test's accuracy and is obtained by calculating the weighted average of the precision and recall. It calculates the percentage of energy correctly assigned to each appliance in the dataset. A higher F_1-score value indicates a better identification of appliance. The mean absolute error and fraction of energy correctly assigned in Equations (8) and (9) are non-event based metrics indicative of how well a load-disaggregation system is able to compute and assign power consumption.

$$MAE = 1/len(T) * \sum_{t=1}^{T} |\hat{y}_t^{(i)} - y_t^{(i)}| \tag{8}$$

$$FECA = 1 - \frac{\sum_{t=1}^{T} \sum_{i=1}^{n} |\hat{y}_t^{(i)} - y_t^{(i)}|}{2 \sum_{t=1}^{T} \bar{y}_t} \tag{9}$$

Where $\hat{y}_t^{(i)}$ and $y_t^{(i)}$ are the estimated and ground-truth power for the i^{th} device at time step t respectively, and $\bar{y}_t$ is the aggregated power at time t. Mean Absolute Error Equation (8) is calculated by taking the absolute energy difference in the total predicted and in the ground-truth. The Fraction of Energy Correctly Assigned (FECA), taken from [31], is used to evaluate the performance accuracy.

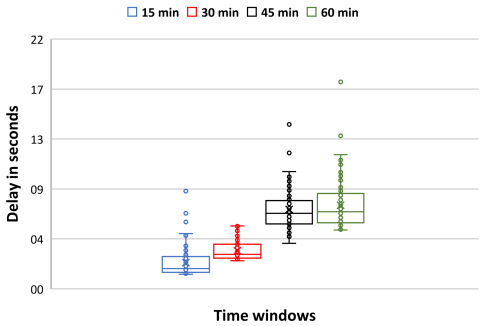

Fig. 5: Overall time-delay for on-line disaggregation

B. Time analysis for on-line disaggregation

As previously mentioned, the proposed On-line-NILM algorithm computes energy disaggregation in near-real-time. To compute such disaggregation, it needs an initial leaning process to build the appliances models. This is a one-time and off-line procedure that does not affect the subsequent on-line and near-real-time disaggregation process. The learning process is handled by the *Event-Based Non-Intrusive Detection* (see Section III-A) at the initial set-up of the algorithm and needs less than 15 min (considering a data-set of about two weeks).

With "near-real-time", we refer to the time needed for both data transmission and data processing. According to the algorithm initial set-up, the smart meter can send a payload with 1 Hz measurements every 15, 30, 45 or 60 min (our disaggregation-time-window). We carried out laboratory tests to calculate the time needed by both data transmission and data processing with the following set-up: (i) a smart meter emulator sends real measurements through MQTT protocol exploiting a conventional ADSL Internet connection; (ii) the On-line-NILM algorithm is running in a server with an Intel Xeon 3.50 GHz processor and 32 GB of RAM. In this context, the smart meter emulator works as publisher, the Flexmeter platform provides the Message Broker and the On-line-NILM algorithm works as subscriber. The median value of data transmission delay is between 2s and 6s for disaggregation-time-windows between 15 and 60 min, respectively. Whilst, the median value of data processing delay is between 40 and 100 ms. Hence, as shown in Figure 5, the total on-line disaggregation delay in the worst-case scenario (i.e. 60 min disaggregation-time-window) will not be over 7s.

C. Performance evaluation

This section presents the disaggregation results of our solution exploiting REDD, UK-DALE and REFIT datasets. Table III details the performance indicators achieved by running our tests on these datasets. As shown in Table III, across the three datasets, F_1-score gives the best results for refrigerators, air-conditioner, freezer, TV and Washing Machine with values higher than 0.70. However, toaster and electronics have a lower F_1-score of about 0.25 due to misclassification that occurred because of non-uniform pattern of these devices.

Table IV shows the estimated and true (or real) energy consumed for the three datasets averaged over the houses

and over the evaluation duration. Such "aggregated" charts have significant value for user feedback [31]. As shown in the Table IV, nine types of appliances are identified in the REDD dataset. On-line-NILM correctly estimates 79% of the total energy compared with the ground-truth contribution for all devices excepting the oven. The energy consumption estimation of the oven is over estimated of about 11% in House-1 due to high amount of false positives at 4000 watts as modeled by our method. On-line-NILM estimates the energy contributions of refrigerator and microwave with a difference of about 5%. Whilst, the lowest percentage estimation is given by stove, dishwasher and bath at about 1%. Regarding the five houses in UK-DALE, the highest energy demands mostly come from the refrigerator and dishwasher at about 38% and 30%. On-line-NILM is able to infer the energy consumption of these devices at 32% and 22% respectively. As shown in the Table IV, seven appliances were identified in House-8 on REFIT. The energy contribution of Washing Machine and Freezer to the overall consumption is 42% and 17%, respectively. On-line-NILM underestimates the disaggregation of the Washing Machine of about 7% due to the false positive of heat cycles. Whilst, it overestimates the disaggregation of the Freezer of about 7%. Looking at the overall performances for all the seven appliances, On-line-NILM estimates their energy consumed with an accuracy of 78%.

Finally, Table V reports the performance of our solution applied to UK-DALE dataset House-1 for the four seasons in 2013. All the six metrics highlight that the disaggregation performance of our On-line-NILM is consistent across the seasons. Indeed, F_1-score ranges: i) between 0.70 and 0.81 for the kettle, ii) between 0.18 and 0.46 for the lighting; iii) between 0.71 and 0.81 for the refrigerator and iv) between 0.83 and 0.91 for the washing machine. Whilst, microwave and dishwasher performance are almost constant across the seasons and F_1-score is about 0.34 and 0.50, respectively. Table VI shows estimated and true energy consumed averaged over each season in House-1. The energy estimation between the seasons for the Kettle, Microwave and Washing Machine and Dishwasher is almost constant and the error ranges between 0% and 3%. Whilst, the energy estimation error for the Refrigerator ranges between 7% in springer and 13% in winter. The estimation of the Light gives the largest consumption difference at 16% in winter.

D. Comparison with literature solution

In this section, we compare the accuracy metrics of results with three literature solutions: i) NILM-TK [22], an FHMM implementation; ii) Neural-NILM [12], a deep neural network adaptation for energy estimation and iii) GSP-Based [6]. In [12], the authors provide both disaggregation performance and disaggregated data for both NILM-TK and Neural-NILM. In Neural-NILM, three deep neural network architectures are adopted: i) *long short term memory*, ii) *denoising autoencoders* and iii) *rectangles*. In particular, *rectangles* network regresses the start-time, end-time and average power of appliances activations. We reported the results of *rectangles* network because it has the best performance with respect to the other two neural networks as reported by the authors.

	Appliance	Precision	Recall	Accuracy	MAE (W)	F_1 - Score	FECA
REDD	Air conditioner	0.93	0.94	0.92	48.28	0.94	0.99
	Electronics	0.18	0.45	0.82	62.60	0.26	0.98
	Lighting	0.53	0.76	0.88	35.11	0.60	0.97
	Refrigerator	0.85	0.89	0.89	3.28	0.87	0.99
	Stove	0.38	0.95	0.96	58.93	0.53	0.94
	Microwave	0.73	0.21	0.95	57.31	0.33	0.95
	Dishwasher	0.88	0.53	0.80	28.43	0.66	0.99
	Bath	0.37	0.25	0.93	21.54	0.30	0.99
	Oven	0.23	0.86	0.76	894.34	0.37	0.81
UK-DALE	Dishwasher	0.78	0.44	0.71	27.72	0.56	0.99
	Kettle	0.42	0.85	0.92	165.97	0.53	0.94
	Microwave	0.60	0.38	0.67	22.16	0.35	0.99
	Refrigerator	0.73	0.87	0.82	4.34	0.79	1.00
	Washing Machine	0.60	1.00	0.60	118.11	0.70	0.88
REFIT	Freezer	0.87	0.90	0.93	3.10	0.89	0.99
	kettle	0.74	0.44	0.97	46.79	0.55	0.96
	Microwave	0.69	0.47	0.98	12.19	0.56	0.98
	Refrigerator	0.71	0.84	0.91	1.34	0.77	1.00
	Toaster	0.20	0.34	0.98	6.47	0.25	0.99
	TV	1.00	0.75	0.83	22.34	0.86	0.98
	Washing Machine	1.00	0.59	0.72	366.89	0.74	0.87

TABLE III: Disaggregation performance indicators

	Contribution	Refrigerator	Air conditioner	Lighting	Microwave	Dishwasher
REDD	Ture	30%	28%	20%	7%	5%
	Estimated	25%	24%	22%	2%	4%
		Bath	Stove	Oven	Electronics	
	Ture	2%	0%	4%	1%	
	Estimated	1%	1%	15%	3%	
UK-DALE	Contribution	Refrigerator	Dishwasher	Kettle	Washing Machine	Microwave
	Ture	38%	30%	16%	6%	10%
	Estimated	32%	22%	29%	6%	11%
REFIT	Contribution	Washing Machine	Freezer	TV	Kettle	Refrigerator
	Ture	42%	17%	14%	13%	6%
	Estimated	35%	24%	13%	10%	8%
		Microwave	Toaster			
	Ture	6%	2%			
	Estimated	5%	4%			

TABLE IV: True and estimated energy contributions

	Appliance	Precision	Recall	Accuracy	MAE (W)	F_1 - Score	FECA
Spring	Dishwasher	0.90	0.33	0.62	102.15	0.48	0.96
	Kettle	0.67	0.80	0.97	24.80	0.73	0.98
	Lighting	0.31	0.90	0.51	108.31	0.46	0.91
	Microwave	0.24	0.42	0.97	19.37	0.31	0.99
	Refrigerator	0.66	0.77	0.74	3.64	0.71	0.99
	Washing Machine	0.84	1.00	0.84	792.26	0.91	0.75
Summer	Dishwasher	0.98	0.33	0.57	105.54	0.50	0.94
	Kettle	0.78	0.83	0.98	8.05	0.81	0.99
	Lighting	0.11	0.62	0.45	138.43	0.18	0.86
	Microwave	0.31	0.38	0.97	8.05	0.34	0.99
	Refrigerator	0.82	0.80	0.82	5.45	0.81	0.99
	Washing Machine	0.84	1.00	0.84	940.41	0.91	0.65
Autumn	Dishwasher	0.96	0.33	0.62	87.74	0.49	0.96
	Kettle	0.70	0.81	0.97	19.04	0.75	0.98
	Lighting	0.27	0.79	0.43	157.20	0.40	0.86
	Microwave	0.27	0.40	0.97	15.04	0.32	0.99
	Refrigerator	0.72	0.77	0.76	0.95	0.74	1.00
	Washing Machine	0.71	1.00	0.71	969.89	0.83	0.66
Winter	Dishwasher	0.94	0.36	0.66	63.71	0.52	0.97
	Kettle	0.63	0.80	0.97	33.04	0.70	0.98
	Lighting	0.32	0.80	0.44	160.89	0.46	0.87
	Microwave	0.29	0.36	0.97	10.57	0.32	0.99
	Refrigerator	0.70	0.75	0.75	0.21	0.73	1.00
	Washing Machine	0.80	1.00	0.80	805.92	0.89	0.74

TABLE V: Seasonal Disaggregation for UK-DALE's House-1

Seasons	Contribution	Refrigerator	Dishwasher	Kettle	Washing Machine	Lighting	Microwave
Spring	True	42.79%	17.51%	14.37%	12.80%	8.52%	4.00%
	Estimated	36.17%	10.98%	13.47%	17.27%	15.96%	6.14%
Summer	True	63.11%	15.67%	12.63%	2.45%	1.19%	4.95%
	Estimated	52.69%	12.24%	12.77%	4.73%	11.32%	6.25%
Autumn	True	53.66%	18.81%	12.76%	4.95%	4.52%	5.28%
	Estimated	41.31%	12.40%	11.77%	8.20%	19.47%	6.86%
Winter	True	44.03%	16.01%	11.24%	17.09%	6.59%	5.04%
	Estimated	30.80%	9.85%	10.25%	21.29%	22.89%	4.91%

TABLE VI: Seasonal energy contributions for UK-DALE's House-1

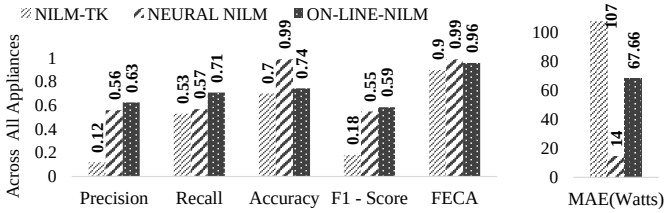

Fig. 6: Accuracy metrics across appliances in uk-dale dataset

Table VII compares the accuracy metrics of our solution with NILM-TK and Neural-NILM for 5 appliances (i.e. fridge, washing machine, dishwasher, microwave and kettle) in UK-DALE dataset. This Table highlights that the worst scores for the three solutions are given by the microwave. NILM-TK reports MAE and F_1-score of about 195 Watts and 0.01 respectively. Neural-NILM has the best MAE of 6 Watts and F_1-score of 0.21. Our solution performs better than the other two with F_1-score of about 0.35. Whilst, MAE is 22.16 Watts. In this case, we learned from disaggregated data that NILM-TK's inferring model for the microwave has multiple states. Instead, we use two state models for the microwave and hence achieved less False Positive inferences. On the other hand, the best scores for all the three methods are given for the refrigerator. NILM-TK reports MAE and F_1-score of about 67 Watts and 0.55 respectively. Neural-NILM has the MAE of 18 Watts and F_1-score of 0.82. In this case, our solution performs better than the other two with MAE of 4.34 Watts, while the F_1-score is about 0.79.

Table VII shows that our method outperformed the benchmark implementation of FHMM on every appliance by the given accuracy metrics. This is also highlighted by Figure 6 that shows histograms of the accuracy indicators summed across all appliances. It reports that NILM-TK has a higher MAE over all devices, about 107 Watts, and the lowest F_1-score, about 0.18. Whilst, our solution, that also exploit FHMM algorithms, performs better with MAE of about 67.66 Watts and F_1-score of about 0.59. Our results are comparable with Neural-NILM performance with MAE of about 14 Watts and F_1-score of about 0.55. In general, Neural-NILM scored better than our proposed method in energy estimation especially for the complex device like dishwasher and washing machine. However, the time and computation effort for training the neural network to obtain the models is extensive and requires appliance level data. Whereas, our approach can construct appliance models from the aggregate data without the need of appliance level sub-metered data.

	Precision	Recall	Accuracy	MAE (W)	F_1 - Score	FECA
Dishwasher						
On-line-NILM	0.78	0.44	0.71	27.72	0.56	0.99
Neural NILM	0.98	0.64	0.99	30.00	0.74	0.98
NILMT-TK	0.05	0.49	0.33	110.00	0.05	0.91
Kettle						
On-line-NILM	0.42	0.85	0.92	165.97	0.53	0.94
Neural NILM	0.70	0.71	1.00	7.00	0.70	0.99
NILMT-TK	0.14	0.29	0.99	98.00	0.19	0.94
Microwave						
On-line-NILM	0.60	0.38	0.67	22.16	0.35	0.99
Neural NILM	0.14	0.40	0.99	6.00	0.21	1.00
NILMT-TK	0.01	0.34	0.91	195.00	0.01	0.84
Refrigerator						
On-line-NILM	0.73	0.87	0.82	4.34	0.79	1.00
Neural NILM	0.79	0.88	0.87	18.00	0.82	0.99
NILMT-TK	0.40	0.86	0.50	67.00	0.55	0.94
Washing Machine						
On-line-NILM	0.60	1.00	0.60	118.11	0.70	0.88
Neural NILM	0.29	0.24	0.98	11.00	0.27	0.98
NILMT-TK	0.04	0.64	0.79	67.00	0.08	0.88

TABLE VII: Disaggregation performance of our solution compared with Neural NILM [12] and NILM-TK [22]

	House 1		House 2		House 6	
Appliance	GSP-Based	Online-NILM	GSP-Based	Online-NILM	GSP-Based	Online-NILM
Lighting	0.49	0.98	0.74	0.57	0.36	1.00
Refrigerator	0.84	0.85	0.80	0.92	0.82	0.84
Dish Washer	0.38	0.55	0.49	0.76	-	-
Oven	0.64	0.37	-	-	-	-
Bath	0.72	0.30	-	-	-	-
Microwave	0.72	0.26	0.94	0.39	-	-
Electronics	-	-	-	-	0.42	0.26
Stove	-	-	0.44	0.65	0.58	0.41
Air Conditioner	-	-	-	-	0.90	0.94

TABLE VIII: F_1-score of our solution compared with GSP-Based [6] on REDD [31]

Table VIII and Table IX report disaggregation performance of On-line-NILM compared with GSP-Based [6] on REDD and REFIT datasets, respectively. With the only exception of the refrigerator in REFIT's House-8, On-line-NILM performs better than the GSP-Based [6] for high energy consuming devices, i.e. refrigerator, dishwasher, air conditioner and washing machine. For example in House-6, F_1-score of On-line-NILM is 0.94, 0.84 for air conditioner and refrigerator. Whilst with GSP-Based, It is 0.9 and 0.82, respectively. For the dishwasher in House-2 and House-1, On-line-NILM scores 0.76 and 0.55, respectively, while GSP-Based [6] gives an F_1-score of 0.49 and 0.38. However, GSP-Based performs better than On-line-NILM for short duration appliances like microwave, toaster and oven. This is due to the advantage of effective

	House 8	
Appliance	GSP-Based	Online-NILM
Freezer	0.32	0.89
TV	0.04	0.86
Refrigerator	0.95	0.77
Washing Machine	0.40	0.74
Microwave	0.52	0.56
kettle	0.84	0.55
Toaster	0.67	0.25

TABLE IX: F_1-score of our solution compared with GSP-Based [6] on REFIT [33]

clustering method in event-based approaches for short-duration appliances, where the rising/ON and falling/OFF events are very close to each-other.

SparseNILM [8] gave estimation accuracy and F_1-score respectively in their paper for selected houses in REDD [31] dataset. Estimation accuracy are reported by SparseNILM [8] exploiting Equation 9 - FECA. It is about 98.8% for houses 1, 2, 3 and 6 in REDD dataset, where authors modeled the difference between appliances ground truth power and aggregate consumption as unknown device. Our solution for the same houses and appliances in REDD dataset scored FECA of 98.65%. This is an improved score with respect to SparseNILM's Noisy Test in which case authors score is 94.9%. Unlike SparseNILM [8], our On-line-NILM does not rely on training using ground-truth measurement from household appliances. Furthermore, our cloud-based disaggregation architecture enables support for smart grid services in contrast with SparseNILM [8], which can run on embedded hardware.

VI. Conclusion

In this work, we presented an on-line and cloud-based NILM methodology for disaggregating residential building electric consumption in near-real-time. It overcomes the limitations of literature solutions through a training-less combined approach of event-based and state-based algorithms. Experimental results were demonstrated over three datasets with a range of appliances and the results show the approach is competitive with respect to literature solutions. The disaggregation results of our solution can be used by applications for user-awareness or for demand side management policies.

References

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

[2] A. Faustine, N. H. Mvungi, S. Kaijage, and K. Michael, "A survey on non-intrusive load monitoring methodies and techniques for energy disaggregation problem," *arXiv preprint arXiv:1703.00785*, 2017.

[3] C. Klemenjak and P. Goldsborough, "Non-intrusive load monitoring: A review and outlook," *arXiv preprint arXiv:1610.01191*, 2016.

[4] A. Bahmanyar, S. Jamali, A. Estebsari, E. Pons, E. Bompard, E. Patti, and A. Acquaviva, "Emerging smart meters in electrical distribution systems: opportunities and challenges," in *Electrical Engineering (ICEE), 2016 24th Iranian Conference on*. IEEE, 2016, pp. 1082–1087.

[5] K. He, L. Stankovic, J. Liao, and V. Stankovic, "Non-intrusive load disaggregation using graph signal processing," *IEEE Transactions on Smart Grid*, 2017.

[6] B. Zhao, L. Stankovic, and V. Stankovic, "On a training-less solution for non-intrusive appliance load monitoring using graph signal processing," *IEEE Access*, vol. 4, pp. 1784–1799, 2016.

[7] J. Z. Kolter and T. S. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation." in *AISTATS*, vol. 22, 2012, pp. 1472–1482.

[8] S. Makonin, F. Popowich, I. V. Bajić, B. Gill, and L. Bartram, "Exploiting hmm sparsity to perform online real-time nonintrusive load monitoring," *IEEE Transactions on Smart Grid*, vol. 7, no. 6, pp. 2575–2585, 2016.

[9] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Non-intrusive load monitoring using prior models of general appliance types," in *AAAI Conference on Artificial Intelligence, Toronto, ON, Canada, 2012*, 2012, p. 356362.

[10] W. Kong, Z. Y. Dong, J. Ma, D. Hill, J. Zhao, and F. Luo, "An extensible approach for non-intrusive load disaggregation with smart meter data," *IEEE Transactions on Smart Grid*, 2017.

[11] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han, "Unsupervised disaggregation of low frequency power measurements," in *Proceedings of the 2011 SIAM International Conference on Data Mining*. SIAM, 2011, pp. 747–758.

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

[13] P. P. M. do Nascimento, "Applications of deep learning techniques on nilm," Ph.D. dissertation, Universidade Federal do Rio de Janeiro, 2016.

[14] Y. F. Wong, Y. A. Şekercioğlu, T. Drummond, and V. S. Wong, "Recent approaches to non-intrusive load monitoring techniques in residential settings," in *Computational Intelligence Applications In Smart Grid (CIASG), 2013 IEEE Symposium on*. IEEE, 2013, pp. 73–79.

[15] E. Patti, E. Pons, D. Martellacci, F. B. Castagnetti, A. Acquaviva, and E. Macii, "multiflex: Flexible multi-utility, multi-service smart metering architecture for energy vectors with active prosumers," in *Smart Cities and Green ICT Systems (SMARTGREENS), 2015 International Conference on*. IEEE, 2015, pp. 1–6.

[16] A. Al Imran, M. A. Syrus, and H. A. Rahman, "An improved event detection algorithm for non-intrusive load monitoring system for low frequency smart meters," in *3rd International Workshop on Non-Intrusive Load Monitoring*, 2016.

[17] L. De Baets, J. Ruyssinck, D. Deschrijver, and T. Dhaene, "Event detection in nilm using cepstrum smoothing," in *3rd International Workshop on Non-Intrusive Load Monitoring*, 2016, pp. 1–4.

[18] A. A. Girmay and C. Camarda, "Simple event detection and disaggregation approach for residential energy estimation," in *Workshop on Non-Intrusive Load Monitoring (NILM), 2016 Proceedings of the 3rd International*, 2016.

[19] L. Rabiner and B. Juang, "An introduction to hidden markov models," *IEEE assp magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[20] Z. Ghahramani, "An introduction to hidden markov models and bayesian networks," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 01, pp. 9–42, 2001.

[21] Z. Ghahramani, M. I. Jordan, and P. Smyth, "Factorial hidden markov models," *Machine learning*, vol. 29, no. 2-3, pp. 245–273, 1997.

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

[23] W. Zhenyu and Z. Guilin, "The application of mean-shift cluster in residential appliance identification," *in Control Conference (CCC), 2011 30th Chinese*, 2011.

[24] M. P. Comaniciu D, "Mean shift: A robust approach toward feature space analysis." *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 5, pp. 603–619, 2002.

[25] C. YZ., "Mean shift, mode seeking, and clustering." *IEEE Trans. on Pattern Analysis and Machine Intelligence,1995*, vol. 17, no. 8, pp. 790–799, 1995.

[26] D. Egarter, V. P. Bhuvana, and W. Elmenreich, "Paldi: Online load disaggregation via particle filtering," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 2, pp. 467–477, 2015.

[27] H. Altrabalsi, V. Stankovic, J. Liao, and L. Stankovic, "Low-complexity energy disaggregation using appliance load modelling," *AIMS Energy*, vol. 4, no. 1, pp. 884–905, 2016.

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

[29] M. Pau, E. Patti, L. Barbierato, A. Estebsari, E. Pons, F. Ponci, and A. Monti, "Low voltage system state estimation based on smart metering infrastructure," in *Applied Measurements for Power Systems (AMPS), 2016 IEEE International Workshop on*. IEEE, 2016, pp. 1–6.

[30] M. Pau, E. Patti, L. Barbierato, A. Estebsati, E. Pons, F. Ponci, and A. Monti, "A cloud-based smart metering infrastructure for distribution grid services and automation," *Sustainable Energy, Grids and Networks*, 2017.

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

[32] J. Kelly and W. Knottenbelt, "The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes," *Scientific Data*, vol. 2, no. 150007, 2015.

[33] D. Murray, L. Stankovic, and V. Stankovic, "An electrical load measurements dataset of united kingdom households from a two-year longitudinal study," *Scientific data*, vol. 4, p. 160122, 2017.