

Non-intrusive load disaggregation solutions for very low-rate smart meter data

Bochao Zhao, Minxiang Ye, Lina Stankovic, and Vladimir Stankovic

*Department of Electronic and Electrical Engineering
University of Strathclyde, Glasgow, G1 1XW, UK
Email: {b.zhao, minxiang.ye, lina.stankovic, vladimir.stankovic}@strath.ac.uk*

Abstract

With the active large-scale roll-out of smart metering worldwide, details about the type of smart meter data that will be available for analysis are emerging. Consequently, focus has steadily been shifting from analysis of high-rate power readings (usually in kHz to MHz) to low-rate power readings (sampled at 1 to 60 sec) and very low-rate meter readings of the order of 15-60 minutes. This has triggered renewed research into practical non-intrusive load disaggregation of low- to very-low granularity meter readings to address challenges not addressed by existing disaggregation approaches, namely, indistinct appliance ON/OFF transitions, increased likelihood of overlapping appliance usage within a sample and noise due to unknown appliances. In this paper, focusing on smart meter readings at hourly resolution, three load disaggregation solutions are proposed based on: (i) optimisation (minimisation of error between aggregate and disaggregated loads), (ii) graph signal processing and (iii) convolutional neural network. These are benchmarked with state-of-the-art approaches, based on factorial hidden Markov model and combinatorial optimisation implemented in the NILMTK toolbox, and discriminative disaggregation sparse coding. The hourly electricity profile data is obtained from real-world active power readings from the REFIT dataset¹ over a period of longer than one year. All proposed

¹The REFIT dataset (cleaned version) used for validation in this paper can be accessed via DOI 10.15129/9ab14b0e-19ac-4279-938f-27f643078cec.

disaggregation approaches outperform benchmarking methods for labelled appliances in terms of both energy performance metrics and faster execution time. The proposed approaches succeed in disaggregating, at very low resolutions, a wide range of loads including white goods even when there are unlabelled loads contributing to the meter readings.

Keywords: Non-intrusive load monitoring, hourly load disaggregation, optimisation process, graph signal processing, convolutional neural network.

1. Introduction

With the goal of rolling out 780 million smart electricity meters worldwide [1] and 200 million in Europe [2] by the end of 2020, it is emerging that, mainly due to storage, data management, and potential privacy preservation [3], the resolution of load measurements available from national smart meters is much lower than the recently assumed 1-60sec. Technical specifications, including type and resolution of smart meter data available for analysis, differ from country to country, but are generally in the range between 15minutes and 1hour. For example, meter data is available at resolutions of 15 minutes in Italy [4], 1 hour in Spain [5], 15 minutes or 1 hour in the US [6], 30 minutes in the UK [7], 1 hour in British Columbia and Ontario, Canada [3].

Smart meter readings provide opportunities for many exciting applications that go beyond accurate and remote billing. One of the enablers of such applications, is Non-Intrusive Load Monitoring (NILM), defined as estimating algorithmically appliance-level energy consumption from aggregate meter readings. NILM has the potential to provide low-cost, efficient and fine-grained energy feedback that can potentially reduce domestic electricity consumption by 0.7%-4.5% compared to only aggregate consumption feedback [8]. However, the applications of NILM go beyond supporting energy-efficient behaviour [9, 10], as NILM has already been shown to support national surveys on energy intensity of domestic activities [11], scalable appliance modelling [12], accurate estimation of the residential consumption phase of food life-cycle assessments [13], house

maintenance and retrofit [14], and detection of anomalous appliances [15].

In general, NILM approaches can be categorized as supervised or unsupervised, depending on whether a labelled dataset is used to train the models or not. Supervised learning performs classification, where detected events (appliances being switched on or off) are assigned to appliance categories by matching extracted features and using pre-trained appliance models (e.g., [16, 17]). In unsupervised NILM methods, on the other hand, detected events are grouped into various categories through analysis of feature similarities or correlation (see, e.g., [18]).

Numerous machine learning and signal processing methods have been investigated to solve the NILM problem, including stochastic finite state machines (Hidden Markov Model and its variants) [19, 20], support vector machine (SVM) [21], decision tree [21, 22], dynamic time warping [22], k-nearest neighbours (K-NN) [16, 21, 23–25], sparse coding [26, 27], neural networks [9, 28–30], graph signal processing (GSP) [18, 31], and optimisation via learning of appliance models and occupancy information [32–36]. Furthermore, advanced hybrid approaches have been proposed for improving core NILM performance, e.g., k-means clustering based training followed by disaggregation using SVM [37], GSP with result refinement using simulated annealing [17], deep neural network utilised to learn deeper and multiple layers of sparse signal representation [38].

Motivated by national smart meter roll-outs at scale, in this paper, we focus on solving the challenging very low-rate NILM problem, where the sampling period is at most 15 minutes. Prior work in this area is limited and is reviewed in Section 2. In summary, current very low-rate disaggregation methods have several limitations, including predicting appliance-cluster consumption instead of individual appliance consumption (e.g., white goods, instead of refrigerator, washing machine, etc.), a large amount of prior information is required, e.g., dwelling and occupancy information from surveys, outdoor weather, etc. Furthermore, validation experiments tend to be limited to aggregated and labelled appliance load profiles instead of true smart meter measurements containing many loads which are unknown.

In this paper, three very low-rate (hourly) NILM approaches are proposed that utilise only smart meter power measurements and appliance manufacturer information, that can be obtained from an appliance survey. It is common practice now by utilities and energy management systems, aiming to provide energy consumption feedback, to ask customers to fill a one-off appliance survey, which includes types of appliances present in the household, make/model/wattage, and sometimes the frequency of use. Customers are informed that the more information they provide, the more accurate their energy efficiency recommendations can be. Only appliances whose metadata has been included in the appliance survey is considered known and can potentially be disaggregated. In this paper, we refer to these known appliances as labelled appliances, and remaining appliances for which we have no metadata (in the form of wattage information) are considered unlabelled appliances.

The proposed methods are evaluated on real aggregate measurements against three state-of-the-art energy disaggregation methods, namely the widely used Factorial Hidden Markov Model (FHMM) and Combinatorial Optimisation (CO), implemented within the publicly available NILM Toolkit [19], and Discriminative Disaggregation Sparse Coding (DDSC) [26], proposed for hourly data. Both FHMM (and its variants) and CO are popular NILM solutions for disaggregation of low-rate active power (1Hz or less), see, e.g., [25, 29, 39, 40].

The key contributions of this paper are:

- A detailed review of the NILM approaches shown to work for very low-rate load disaggregation and the types of loads that have been disaggregated.
- An unsupervised optimisation-based (OPT) NILM approach that aims to estimate appliance-level consumption by finding the combination of appliance models generated from manufacturer information, that minimises a cost function, after estimating and removing the baseload.
- An unsupervised GSP-based NILM approach, adapted from the higher resolution GSP-based approach of [18], where an additional graph for time-of-day feature is added.

- A supervised convolution neural network (CNN)-based approach, which is trained only on a small set of aggregate profile data, where time information is considered as a feature.
- Validation and benchmarking on the REFIT dataset [41], known to be ‘noisier’ than other publicly available datasets due to many unlabelled appliances running simultaneously with labelled appliances.
- Benchmarking with three popular NILM approaches, Factorial Hidden Markov Model (FHMM), CO implemented in NILMTK [19] and DDSC proposed in [26].
- Presentation and discussion of disaggregation results comparing commonly used metrics: Total Energy Correctly Assigned (*TECA* or *Acc.*) and best-performing match rate, and direct comparison of energy consumption of each appliance disaggregated w.r.t ground truth.
- Presentation and discussion of disaggregation results in terms of execution time for all proposed schemes w.r.t benchmarked methods.

The rest of this paper is organized as follows: Section 2 provides an in-depth review of very low-rate (15-60 minutes) NILM solutions; the problem of energy profile disaggregation and all proposed and benchmarked very low-rate NILM approaches are described in Section 3; the experimental setup including electricity profile extraction from existing high granularity datasets is presented in Section 4; the results are demonstrated and discussed in Section 5; Section 6 summarises our findings and future work.

2. Prior work on very low-rate NILM

An hourly energy disaggregator is proposed in [42], based on a multi-objective genetic algorithm with pre-learnt appliance inferences, updated weekly. Appliance inferences are generated based on prior knowledge of real and reactive consumed energy, weather information, appliance ownership data, etc. The

appliance ownership data includes both unit energy consumption of typical appliance types in Canada and corresponding statistical knowledge, such as ownership and usage frequency by house type. The method is validated on hourly profile data from BC Hydro utility, resulting in more than 50% of the aggregate energy consumption disaggregated into 12 appliance categories for both real and reactive energy. The same authors proposed an optimisation method for hourly disaggregation via a study of power factors [3], which is validated on the same dataset used in [42]. Unlike [42], disaggregation results for three clusters grouped by power factor are provided in [3], achieving *F-measure* classification metric of 59%-81% for all clusters. Another disaggregation method carried out on energy profile readings collected from Canadian houses is presented in [36], based on modelling piece-wise functions of hourly real energy versus external temperature for baseload, heating and cooling devices. A model-based regression method is proposed in [35] for isolating space heating energy consumption from hourly energy profile data and tested on 470 Norwegian houses, where models are established based on hourly and daily load profile, weather data and response data from a household survey. The approach relies on the assumption that space heating consumption is weather-dependent and related to household size, number of residents, etc.

An unsupervised approach based on semi-binary non-negative matrix factorization (SBNMF) is proposed in [43] for the 30-min NILM problem, where SBNMF and its variants are used for dictionary learning. Dictionaries are labelled by a random forest classifier utilizing a pre-learnt descriptive database on the cloud. Consumer feedback for improving dictionary learning is also explored. The proposed SBNMF performs the best among all benchmarks, achieving average disaggregation match rate of 60% across four commonly used appliances: fridge, washing machine, TV and air conditioner.

Three multi-label classification algorithms, based on Decision Tree (DT), SVM and k-nearest neighbours (K-NN), are proposed in [21] and validated on energy profile data from a subset called IRISE within the REMODECE dataset [44] at sampling rates of 10 min and 1 hour. Three appliance categories, defined

based on clusters obtained using principal component analysis, contain domestic appliance types showed in Table 1. All algorithms generally achieve much lower appliance-level classification accuracy during hourly disaggregation compared to 10-min disaggregation, based on *F-measure* and area under curve (AUC) metrics. On average, various methods score 68% on 10-min and 50% on hourly data.

K-NN classification is also utilised in [23], [24] and [25]. In [23], the K-NN approach is tested for 15-min load profiles from smart metering of 187 houses in East Anglia, UK. Features are derived from both magnitude and time for modelling 10 appliances. Only classification results in the form of a confusion matrix with classification accuracy are presented, showing that their proposed K-NN and random forest for benchmarking can achieve at least 60% classification accuracy on both daily and weekly data sets. Additional optional features, extracted from reactive power and active-reactive power correlation, and core features are adaptively selected and weighted for each appliance during training in the supervised K-NN approach [24]. Results are presented for 15-min and hourly electricity profiles from REDD [45], REFIT [41] and AMPds [46] datasets, showing that up to 62% of the daily energy consumption can be disaggregated from the total noisy electricity usage profile with 15-min and 60-min granularity. Monthly electricity bills and household characteristics such as house size and occupancy are required for a transfer learning based K-NN classifier [25], where a target house is matched to similar houses in the database via K-NN and the corresponding monthly appliance-level energy consumption then estimated. 57 houses from Dataport dataset are used for evaluation demonstrating consumption accuracy around 52%, with up to 5% improvement if 15-min smart meter readings are available for feature extraction.

A supervised GSP-based power disaggregation method, based on the approach of [47], is applied to aggregated power measurements of labelled appliances downsampled to 15 minutes in [48], through iteratively identifying samples of power level similar to the labelled measurements for a particular appliance via graph total variation minimisation. Consumption accuracy of 80%-95% is

shown using aggregated data from 4 labelled appliances and a small amount of added noise to simulate energy consumed by unlabelled appliances for only 18 days. An unsupervised hybrid approach is proposed in [49] for disaggregation of activities (not loads), where Markov models are built from a time-of-use survey, requiring knowledge of appliance list and usage frequency, number of inhabitants, their age and employment status, type of heating used, nominal power per appliance and probability of an activity. FHMM and CO from NILMTK, DDSC from [26] and GSP are used for benchmarking. Note that the aggregate power signal used in [49] is not whole-house smart metering measurements like in this paper, but defined as the sum of sub-measurements. Hence, the influence of noise due to unlabelled loads is ignored in disaggregation. The conclusion is that while the performance of supervised GSP [47], implemented using the GSP Toolbox of Matlab [50], is comparable with other supervised benchmarking approaches and better than the proposed unsupervised method, the execution time for the implemented GSP is of the order of a few hours.

In the DDSC approach of [26], the hourly load profile for each appliance in unseen houses is predicted using sparse coding relying on pre-trained appliance models. The approach comprises three steps: 1) sparse coding pre-training; 2) discriminative disaggregation training; and 3) testing (see [26, 27] for details). Results are provided only for aggregated sub-metered readings of 10 labelled appliances, again not representative of real aggregate smart metering power consumption readings that include noise due to unlabelled appliances. A variant of DDSC obtains up to 55% for consumption accuracy.

Table 1 summarises the list of appliances that were disaggregated in the aforementioned review of low (1-60sec sampling interval) to very low-rate (10-60min) NILM. An important observation is that the range of disaggregated appliances decreases as the sampling interval increases from 1 sec to 1 hour. The main reason is that it becomes harder to disaggregate appliances with short operation time, such as Hair dryer, Microwave, Kettle, Toaster, as granularity decreases. We also note that the REDD dataset is used to demonstrate disaggregation of most appliances in the literature, which implies that, while

reproducible, the algorithms have been tested in a relatively low-noise (few unlabelled appliances operating simultaneously) dataset which is not the case in actual smart meter measurements, made up of 40+ appliances present in an average household. Furthermore, the table (first column) also indicates that the data for many appliances originates from non-public datasets (indicated by +), which implies that results are not reproducible and amenable for further analysis by others.

Unlike the previously discussed NILM approaches tackling very low-rate NILM, our proposed approach differs in the following ways. OPT, like CO, also minimises the tolerance between aggregate measurements and sum of appliance-level measurements. However, the difference lies in the measurements: in CO, optimisation is made on a sample-by-sample basis independently, whereas in OPT optimisation is made over a load sequence or small window of adjacent samples of aggregate and appliance-level power consumption to allow for dependency between adjacent samples and multiple same-appliance runs within a sample. Furthermore, CO appliance models are built from sub-metered appliance instantaneous power for each operational state, whilst OPT appliance models avoid difficult-to-obtain sub-metering information and rely on manufacturers wattage or energy-consumption-per-run information.

The GSP-based approach proposed in [31] is supervised and comprises a single graph based on meter readings, whereas the GSP approach proposed in this paper is unsupervised and two graphs are built for energy profile and time, respectively. Unlike [48] and [49], our proposed unsupervised GSP algorithm has a Matlab execution time of around 7 minutes to disaggregate an hourly profile of 78 weeks. We also note that the baseload removal pre-processing step proposed in [48] fails to separate baseload from noisy hourly energy profile signal and makes no improvement to disaggregation.

Additionally, unlike the majority of the aforementioned literature, we do not resort to additional environmental data. We evaluate our proposed approaches on public datasets that closely resemble real-life ‘noisy’ smart meter measurements that include many unlabelled appliances. Besides *detecting* appliance use

(classification only) we also estimate or disaggregate energy consumed and provide results for a testing period of over a year instead of a very short period, in order to capture a large range of appliance usage patterns.

Table 1: A summary of appliances disaggregated in literature at low to very low sampling rates. Each appliance is linked to the corresponding datasets, including REDD [45], REFIT [41], GREEND [51], AMPds [46], UK-DALE [52], Dataport [53], REMODECE [44] and others not publicly available (from manufacturer, supplier, self-monitoring, etc.), denoted using “+”.

Appliance	Aggregate signal granularity		
	1 hour	10-30 min	1-60 sec
Bathroom GFI [45]			[16, 18]
Clothes dryer [44, 45, 53]+	[21, 26, 42]	[21, 25, 48]	[16, 18, 20, 22, 27]
Clothes washer [41, 44, 46] [45, 51-53]+	[21, 24, 26, 42]	[21, 23, 25, 43]	[9, 16, 22, 28-30] [17, 18, 37]
Dishwasher [41, 44-46] [51, 52]	[21, 24, 26, 42]	[21, 23, 25, 48]	[9, 20, 22, 27-29] [17]
Electronics [45, 46, 53]+	[24, 26]	[25]	[16-18, 27, 30, 37]
Fridge/freezer/ fridge-freezer [41, 45, 52, 53]+	[24, 26, 42]	[23-25] [43, 48, 49]	[9, 16, 22, 27-29] [17, 18, 30, 37, 48]
Geyser +		[48]	[48]
Hair dryer [41]			[22]
Heat pump [45, 46]	[24]		[20]
Hot water unit/ furnaces [41, 44-46]+	[21, 24, 42]	[21, 23]	[17, 22, 27]
HVAC [41, 45, 46]+	[24, 26, 35, 42]	[23, 24, 43, 49]	[16-18, 20, 27]
Kettle [41, 52]+		[23]	[9, 17, 22, 28, 29]
Kitchen outlets [41, 44, 45]+	[24, 26, 42]	[21, 24]	[16-18, 22, 27]
Lighting [45]+	[26, 42]	[49]	[16, 18]
Microwave [41, 44, 45, 52]+	[21]	[21, 23]	[9, 16, 22, 27-29] [17, 18, 30, 37, 48]
Oven [41, 44, 45]+	[21]	[21, 23]	[16, 18, 20, 22]
PC/ICT +	[26]	[23, 49]	
Stove [45]+	[42]		[17, 18, 27, 37]
Toaster [41, 45]			[22, 37, 48]
TV [41, 51]+	[26]	[23, 43]	[17, 18, 22, 37]

3. Very low-rate load disaggregation

In this section, we first introduce our notation and set the problem of very low-rate NILM. Then, we present three proposed solutions that will be evaluated in the next section.

3.1. Notation and Problem Formulation

Vectors and matrices are denoted by bold letters, e.g., \mathbf{E} , and their entries by single-value variables with italic letters, e.g., E_i . For an n -length vector \mathbf{E} , $\mathbf{E}_{p:q}$, $1 \leq p \leq q \leq n$, denotes a vector $[E_p, \dots, E_q]$. Similarly, for an $n \times m$ matrix \mathbf{P} , $\mathbf{P}_{a:b,c:d}$ is a submatrix containing Rows a to b and Columns c to d of \mathbf{P} . For a matrix \mathbf{A} , $\mathbf{A}^\#$ denote pseudo inverse of \mathbf{A} . The sets are denoted using calligraphy fonts, e.g., \mathcal{N} , and their cardinality is $|\mathcal{N}|$. \emptyset is an empty set.

Let $E_i \geq 0$ be the aggregate consumed energy at time instance i . It can be expressed as

$$E_i = \sum_m E_i^m + n_i, \quad (1)$$

where $E_i^m \geq 0$ is the energy consumed by Appliance m at time instance i and n_i is measurement noise. The very low-rate disaggregation task is to estimate E_i^m given E_i .

We next describe, one by one, the three proposed approaches to solve the above very low-rate load profile disaggregation problem.

3.2. Proposed optimisation-based approach (OPT)

Our proposed OPT approach does not require any training of the model. It requires no other information besides smart meter consumption readings, i.e., E_i , and the appliance manufacturer information, i.e., typical energy consumption per operation cycle, or alternatively, rated power and operation duration of the appliance.

The very low-rate NILM problem can be reformulated by splitting it into two steps: 1) estimating the consumption of always-on appliances and removing

their contribution from the aggregate; 2) estimating the consumption of other appliances. To proceed in this way, we rewrite (1) as

$$E_i = \sum_{m \in \mathcal{M}} E_i^m + \sum_{m \in \mathcal{N}} E_i^m + n_i, \quad (2)$$

where \mathcal{M} refers to the set of appliances that are always on (including baseload, stand-by, and appliances with short operation cycles, such as refrigerators) and \mathcal{N} is the set of all other appliances. Then, the very low-rate NILM problem can be formulated as:

$$\arg \min_{\{E_i^m\}, m \in \mathcal{N}} \sum_{i=1}^N \left| \hat{E}_i - \sum_{m \in \mathcal{N}} E_i^m \right|, \quad (3)$$

where N is the total number of samples to be disaggregated, and \hat{E}_i is obtained by removing the estimated always-on load profile, that is:

$$\hat{E}_i = E_i - \sum_{m \in \mathcal{M}} E_i^m. \quad (4)$$

OPT directly solves the optimisation problem in Eq. (3), after the always-on load profile is estimated and removed from aggregate profile signal as in Eq. (4).

The algorithm consists of three steps: removal of always-on loads (Alg. 1); appliance modelling based on appliance power rating; and solving the optimisation problem in Eq. (3) (Alg. 2), which are discussed in the following subsections.

3.2.1. Always-on load estimation and removal

The first task is to estimate the always-on loads and remove them from the aggregate signal. The proposed steps are shown in Alg. 1. Similarly to [54], we estimate always-on loads by assuming that the electricity consumption between 12AM and 5AM is mainly due to the always-on appliances. Let ω be the number of days considered for disaggregation and τ the number of samples collected in a single day during the period between 12AM and 5AM. E.g., for hourly sampling rate, τ is 5, as it corresponds to the 12AM-5AM period of a day. Let $r_{i,j} \in \mathbf{R}$ be the aggregate measurement during sampling period j of Day i , where $i = 1, \dots, \omega$ and $j = 1, \dots, \tau$, and \mathbf{R} is a $\omega \times \tau$ matrix containing all such measurements.

Algorithm 1: Proposed always-on load estimation.

Input: Measurements \mathbf{E} for ω days;

Output: $\sum_{m \in \mathcal{M}} \mathbf{E}^m$; $\hat{\mathbf{E}}$;

- 1 **Form** $\omega \times \tau$ matrix \mathbf{R} by extracting daily 12AM-5AM usage from \mathbf{E} ;
- 2 **Initialise** $\Delta\mathcal{R} \leftarrow \{\emptyset\}$; $\mathcal{R}^{max} \leftarrow \{\emptyset\}$; $\mathcal{R}^{min} \leftarrow \{\emptyset\}$;
- 3 **for** $i = 1$ to ω **do**
- 4 **for** $j = 1$ to $\tau - 1$ **do**
- 5 $|\Delta R_{i,j}| \leftarrow |R_{i,j+1} - R_{i,j}|$; $\Delta\mathcal{R} \leftarrow \Delta\mathcal{R} \cup \{|\Delta R_{i,j}|\}$;
- 6 **for** $j = 2$ to $\tau - 1$ **do**
- 7 **if** $R_{i,j} > R_{i,j+1}$ and $R_{i,j} > R_{i,j-1}$ **then**
- 8 $\mathcal{R}^{max} \leftarrow \mathcal{R}^{max} \cup \{|\Delta R_{i,j}|\}$;
- 9 **else if** $R_{i,j} < R_{i,j+1}$ and $R_{i,j} < R_{i,j-1}$ **then**
- 10 $\mathcal{R}^{min} \leftarrow \mathcal{R}^{min} \cup \{|\Delta R_{i,j}|\}$;
- 11 $T \leftarrow \text{median}(\Delta\mathcal{R})$; $L_{max} \leftarrow \text{median}(\mathcal{R}^{max})$; $L_{min} \leftarrow \text{median}(\mathcal{R}^{min})$;
- foreach** $E_i \in \mathbf{E}$ **do**
- 12 **if** $E_i - L_{max} \leq \gamma$ **then**
- 13 **set** $\sum_{m \in \mathcal{M}} E_i^m$ to E_i ;
- 14 **if** $|E_i - L_{max}| \leq |E_i - L_{min}|$ **then**
- 15 **set** $\sum_{m \in \mathcal{M}} E_i^m$ to L_{max} ;
- 16 **else**
- 17 **set** $\sum_{m \in \mathcal{M}} E_i^m$ to L_{min} ;
- 18 **else if** $\sum_{m \in \mathcal{M}} E_{i-1}^m$ is set above to L_{min} **then**
- 19 **set** $\sum_{m \in \mathcal{M}} E_i^m$ to L_{max} ;
- 20 **else**
- 21 $\sum_{m \in \mathcal{M}} E_i^m \leftarrow L_{min}$;
- 22 **calculate** $\hat{\mathbf{E}}$ using Eq. (4);
- 23 **return** $\sum_{m \in \mathcal{M}} \mathbf{E}^m$, $\hat{\mathbf{E}}$;

Note that always-on load is not constant as it fluctuates over time (e.g., refrigerator has cooling and stand-by stages, whose duration depends on its usage) containing local minima and maxima. Therefore we attempt to mitigate the influence of rare appliance usage by residents during 12AM-5AM on always-on load removal. To do that, we set a threshold γ , as the median of all elements in $\Delta\mathcal{R}$, where $\Delta\mathcal{R}$ is a set of differences between consecutive samples in the same row in \mathbf{R} (signal fluctuations), defined in Lines 4 – 5 in Alg. 1.

We define L_{max} and L_{min} , as median values of all local maxima and minima of \mathbf{R} calculated as shown in Lines 6 – 11, where \mathcal{R}^{max} and \mathcal{R}^{min} are the sets containing all local maxima and minima, respectively. For each element in \mathbf{E} , if $E_i - L_{max} \leq \gamma$, we estimate that only always-on appliances were operating during the sampling period i . Thus we estimate such load for always-on appliances, $\sum_{m \in \mathcal{M}} E_i^m$, as E_i , and therefore, from Eq. (4), $\hat{E}_i = 0$. Based on this assumption, if $|E_i - L_{max}| \leq |E_i - L_{min}|$, we define always-on load during sampling period i as a local maximum, otherwise, it is identified as a local minimum.

For the case $E_i - L_{max} > \gamma$, namely, E_i is already higher than usual maximum, it is likely loads other than always-on appliances were in operation. Thus, always-on appliance load $\sum_{m \in \mathcal{M}} E_i^m$ will be estimated as L_{max} if the always-on load during last sampling period $i - 1$ is estimated as a local minimum. Otherwise, if the last always-on load estimation is a local maximum, $\sum_{m \in \mathcal{M}} E_i^m$ will be set to L_{min} . Hence, $\hat{E}_i = E_i - L_{max}$ or $\hat{E}_i = E_i - L_{min}$.

For a particular case when the initial l samples in \mathbf{E} all comply with $E_i - L_{max} > \gamma$ for $i = 1, 2, \dots, l$, while there is no previous sample for determining $\sum_{m \in \mathcal{M}} \mathbf{E}_{1:l}^m$, the always-on load estimation for such period remains blank pending the identification of the next $E_{l+1} \leq L_{max} + \gamma$. Then, based on whether $\sum_{m \in \mathcal{M}} E_{l+1}^m$ is a local maximum or a local minimum, we set $\sum_{m \in \mathcal{M}} E_l^m$ to L_{min} or L_{max} , respectively. Similarly, each value in $\sum_{m \in \mathcal{M}} \mathbf{E}_{1:l-1}^m$ can be estimated based on the estimation of the next sampling period.

3.2.2. Appliance modelling

We build a model of each appliance $m \in \mathcal{N}$ using manufacturer information on appliance power rating. We split these appliances into two categories: The first category $\mathcal{N}_1 \subseteq \mathcal{N}$ contains appliances whose energy consumption per use does not fluctuate much. Such appliances either have pre-set running mode options, such as washing machines, or consume a more or less constant amount of energy per use, such as kettles. For each $m \in \mathcal{N}_1$, \bar{W}^m represents the total energy consumption during its typical use. The corresponding $\bar{T}^m \in \mathbb{Z}^+$ is the estimated maximum duration, in units of samples, for each run of Appliance m . E.g., for an appliance with typical use less than one hour, such as kettle, $\bar{T}^m = 2$ (the kettle might have started during the previous sampling period and continued during the current period); while for a washing machine which is on for more than one hour but less than two, $\bar{T}^m = 3$.

The second appliance category $\mathcal{N}_2 \subseteq \mathcal{N}$ refers to a group of appliances with constant rated power but variable usage duration, e.g., computer and television. For $m \in \mathcal{N}_2$, since no specific \bar{T}^m is defined, \bar{W}^m is set to the product of rated power and sampling period. Thus, for hourly load profile, \bar{W}^m for $m \in \mathcal{N}_2$ refers to the total energy consumed by Appliance m running for the whole hour.

The consumed energy $\hat{\mathbf{E}}^m$ is represented by:

$$\hat{\mathbf{E}}^m = \begin{cases} \boldsymbol{\alpha}^m \bar{W}^m, & \text{for } m \in \mathcal{N}_1, \\ \boldsymbol{\beta}^m \bar{W}^m, & \text{for } m \in \mathcal{N}_2, \end{cases} \quad (5)$$

where the j -th element in N -length vector $\boldsymbol{\alpha}^m$ is within $[0, 1]$ representing the percentage of energy consumed during sampling period j , relative to \bar{W}^m ; j -th element of a vector $\boldsymbol{\beta}^m$, on the other hand, is 0 or 1 depending whether Appliance m was on or off during the sampling period j .

To clarify the definition of $\hat{\mathbf{E}}^m$ for $m \in \mathcal{N}_1$ in (5), we give the following example:

$$\hat{\mathbf{E}}^m = \left[\underbrace{\alpha_1^m, \alpha_2^m, \alpha_3^m}_{Run1}, \alpha_4^m, \alpha_5^m, \underbrace{\alpha_6^m, \alpha_7^m}_{Run2}, \dots \right] \times \bar{W}^m. \quad (6)$$

Eq. (6) indicates Appliance m runs two times within the selected period. Hence,

based on the rules we defined, $\alpha_1^m + \alpha_2^m + \alpha_3^m = 1$; $\alpha_6^m + \alpha_7^m = 1$ and $\alpha_4^m = \alpha_5^m = 0$. Then,

$$\hat{\mathbf{E}}^m = \left[\underbrace{\alpha_1^m \bar{W}^m, \alpha_2^m \bar{W}^m, \alpha_3^m \bar{W}^m}_{Run1}, 0, 0, \underbrace{\alpha_6^m \bar{W}^m, \alpha_7^m \bar{W}^m}_{Run2}, \dots \right], \quad (7)$$

where $\hat{E}_1^m + \hat{E}_2^m + \hat{E}_3^m = \bar{W}^m$ and $\hat{E}_6^m + \hat{E}_7^m = \bar{W}^m$.

3.2.3. Disaggregation via optimisation

The OPT disaggregation algorithm is given in Alg. 2. The optimisation problem in (3) is NP-complete [34]. We find an approximate solution using CVX [55] in Matlab, where the infeasible path-following algorithm, as a solver for semidefinite quadratic linear programming, is used for searching a non-negative solution based on two Newton steps per iteration [56]. Constraints are heuristically set to reduce the number of candidates and optimisation complexity.

We split the entire sequence to be disaggregated into windows, where each window contains a consecutive non-zero profile segment $\hat{\mathbf{E}}_{seq}$, shown in Line 1 of Alg. 2. Alg. 2 relies on the assumption that each appliance $m \in \mathcal{N}_1$ runs for up to \hat{k}_m times within the window, where \hat{k}_m is heuristically set based on expected appliance usage patterns, as shown in Lines 4 – 7, to reduce the number of candidates and trade-off algorithm performance and complexity. That is, in our implementation, we fix $\sum_{i=1}^n \alpha_i^m$, i.e., the number of runs of Appliance m , to an integer number between 0 and \hat{k}_m , for $m \in \mathcal{N}_1$, and repeat the optimisation steps for all possible values. This is achieved by iterating through a set \mathcal{F} , that contains all possible combinations of the values of $\sum_{i=1}^n \alpha_i^m, m \in \mathcal{N}_1$, where each $\sum_{i=1}^n \alpha_i^m$ can take an integer value between 0 and \hat{k}_m . Note that each vector in the set \mathcal{F} , \mathbf{F} , is of length N_0 and contains as its m -th element the value of $\sum_{i=1}^n \alpha_i^m$.

Then, Line 11, corresponding to Eq. (3), solves the optimisation problem with the solution denoted by α^{m*} and β^{m*} . In Lines 12 – 16, the constraints are set based on the definition of the variables presented in Subsection 3.2.2, where each $\sum_{i=1}^n \alpha_i^m$ for $m \in \mathcal{N}_1$ is a fixed integer and $\alpha_i^m \in [0, 1]$. Finally, the

Algorithm 2: Proposed OPT load profile disaggregation.

Input: $\hat{\mathbf{E}}$; k ; Model parameters \bar{W}^m for each appliance $m \in \mathcal{N}$ where

$\mathcal{N} = \mathcal{N}_1 \cup \mathcal{N}_2$; Model parameters \bar{T}^m for each appliance $m \in \mathcal{N}_1$;

Output: Disaggregated load profile $\hat{\mathbf{E}}^m$ for all appliances in \mathcal{N} ;

1 **Split** $\hat{\mathbf{E}}$ into sequences of consecutive non-zero profile samples $\hat{\mathbf{E}}_{seq}$;

2 **foreach** n -length sequence $\hat{\mathbf{E}}_{seq}$ **do**

3 $T_{opt} \leftarrow +\infty$;

4 **foreach** $m \in \mathcal{N}_1$ **do**

5 $\hat{k}_m \leftarrow k$;

6 **if** $n/\bar{T}^m \leq k$ **then**

7 $\hat{k}_m \leftarrow n/\bar{T}^m$;

8 $N_0 \leftarrow |\mathcal{N}_1|$

9 $\mathcal{F} \leftarrow [\{0, \dots, 0\}, \{0, \dots, 0, 1\}, \dots, \{\hat{k}_1, \dots, \hat{k}_{N_0}\}]$;

10 **foreach** $\mathbf{F} \in \mathcal{F}$ **do**

11 **solve** $\{\alpha^{m*}, \beta^{m*}\} =$

$$\arg \min_{\alpha^m, \beta^m} \left\| \hat{\mathbf{E}}_{seq} - \sum_{m \in \mathcal{N}_1} \alpha^m \bar{W}^m - \sum_{m \in \mathcal{N}_2} \beta^m \bar{W}^m \right\|;$$

12 **subject to**

13 $\{\sum_{i=1}^n \alpha_i^m\} = F_m; 0 \leq \alpha_i^m \leq 1$;

14 **foreach** $m \in \mathcal{N}_2$ **do**

15 $\beta_i^m \in \{0, 1\}$;

16 $T_{opt}^* \leftarrow \left\| \hat{\mathbf{E}}_{seq} - \sum_{m \in \mathcal{N}_1} \alpha^{m*} \bar{W}^m - \sum_{m \in \mathcal{N}_2} \beta^{m*} \bar{W}^m \right\|$;

17 **if** $T_{opt}^* < T_{opt}$ **then**

18 $T_{opt} \leftarrow T_{opt}^*$; $\alpha_{opt}^m \leftarrow \alpha^{m*}$; $\beta_{opt}^m \leftarrow \beta^{m*}$;

19 **calculate** $\hat{\mathbf{E}}_{seq}^m$ using Eq. (5) with α_{opt}^m and β_{opt}^m ;

20 **return** $\hat{\mathbf{E}}^m$

solution with the lowest optimisation loss, denoted by α_{opt}^m and β_{opt}^m , among all the solutions for various values of $\sum_{i=1}^n \alpha_i^m \in [0, \hat{k}_m]$, is used to calculate

appliance load profile by Eq. (5).

3.3. Approach based on Graph signal processing

We build on the GSP-based unsupervised approach of [18] by generating an additional graph for the time-of-day feature, motivated by NILM approaches [24, 48, 57], where time-of-day usage pattern is exploited.

Using notation of [18, 31], $\mathcal{G} = (\mathcal{V}, \mathbf{A})$ is used to denote an undirected graph. $\mathcal{V} = \{v_0, \dots, v_{N-1}\}$ is a set of vertices and $\mathbf{A} \in \mathbb{C}^{N \times N}$ represents a weighted $N \times N$ adjacency matrix, where $A_{i,j}$ corresponds to the weighted edge from Vertex v_i to Vertex v_j and weight depends on the relationship between vertices v_i and v_j . A vector $\mathbf{s} \in \mathbb{R}^N$ is then defined as the *graph signal* that maps $\mathcal{V} \rightarrow \mathbb{R}$ [58], where each element s_i represents a signal value at Vertex v_i .

The proposed GSP clustering steps are shown in Alg. 3. Two graphs are generated, namely, the energy profile graph \mathcal{G}^E , where adjacency matrix \mathbf{A}^E is defined by the Gaussian kernel weighting function in Line 8 and the time-of-day graph \mathcal{G}^T , defined in Lines 9–12, which is used to capture routine or correlation in appliance patterns of use at similar times across different days. ρ^E and ρ^T are scaling factors. The combined adjacency matrix is then defined in Line 13, where λ is a trade-off factor. The remaining steps are the same as the clustering algorithm proposed in [18], based on total variation regularization upon graphs, where $\hat{\cdot}$ denotes the concatenation of two vectors.

The overall disaggregation algorithm is shown in Alg. 4, and consists of repeating the clustering steps until all samples are grouped into clusters, and then labelling the clusters and calculating disaggregated load profile. RSD_f in Line 9 is defined as *relative standard deviation* for quality evaluation of Cluster \mathbf{C}_f :

$$RSD_f = \left| \frac{\sigma_f}{\mu_f} \right|, \quad (8)$$

where σ_f and μ_f refer to standard deviation and mean values of all elements in Cluster \mathbf{C}_f , respectively. For the rejected clusters with RSD_f greater than a heuristically set threshold T_{RSD} , we enhance clustering by halving ρ_E as in [18]. Once ρ_E is halved, we repeat clustering until ρ_E becomes very small (regulated

Algorithm 3: GSP-based clustering

Input: l -length aggregate profile measurements \mathbf{E} ; ρ_E ; ρ_T ; q ; λ

Output: Cluster set \mathbf{C} of all qualified samples; updated aggregate profile measurements \mathbf{E} ;

```
1 Initialize  $\mathbf{C} \leftarrow []$ ;  $s_1^* \leftarrow 1$ ;  
2 Split  $\mathbf{E}$  into  $(n - 1)$ -length sequences  $\tilde{\mathbf{E}}$ ;  
3 foreach  $\tilde{\mathbf{E}}$  do  
4   Initialize  $n \times 1$  graph signal vector  $\mathbf{s}$ , with  $s_1 \leftarrow 1$  and  
    $s_i \leftarrow 0, \forall i \in [2 : n]$   
5   for  $i = 1$  to  $n$  do  
6     for  $j = 1$  to  $n$  do  
7        $A_{i,j}^E \leftarrow \exp\left(-\frac{\|\tilde{E}_i - \tilde{E}_j\|_2^2}{\rho_E^2}\right)$ ;  
8       if  $|i - j| \bmod 24 \leq 12$  then  
9          $A_{i,j}^T \leftarrow \exp\left(-\frac{\| |i-j| \bmod 24 \|_2^2}{\rho_T^2}\right)$ ;  
10      else  
11         $A_{i,j}^T \leftarrow \exp\left(-\frac{\|24 - (|i-j| \bmod 24)\|_2^2}{\rho_T^2}\right)$ ;  
12         $A_{i,j} \leftarrow \lambda A_{i,j}^E + (1 - \lambda) A_{i,j}^T$ ;  
13         $D_{i,i} \leftarrow \sum_{j=1}^n A_{i,j}$ ;  
14       $\mathbf{L} \leftarrow \mathbf{D} - \mathbf{A}$ ;  
15       $\mathbf{s}^* \leftarrow \mathbf{L}_{2:n,2:n}^\# (-s_1) \mathbf{L}_{1,2:n}^T \cap \mathbf{s}^*$ ;  
16 for  $i = 1$  to  $l$  do  
17   if  $s_i^* \geq q$  then  
18      $\mathbf{C} \leftarrow \mathbf{C} \cup E_i$ ;  
19     remove  $E_i$  from  $\mathbf{E}$ ;  
20 return  $\mathbf{C}, \mathbf{E}$ 
```

by a parameter κ close to zero). In [18], feature matching is required for pairing positive and negative resulting clusters and ON/OFF transition events in paired

Algorithm 4: The proposed GSP load profile disaggregation

Input: Aggregate profile measurements \mathbf{E} ; ρ_E ; ρ_T ; q ; T_{RSD} ; κ

Output: Disaggregated load profile $\hat{\mathbf{E}}^m$

```
1 Initialize  $f \leftarrow 1$ ;  $g \leftarrow 1$ ;  $\mathcal{C}^* = \emptyset$ 
2 while  $\rho_E > \kappa$  do
3   while  $\mathbf{E} \neq []$  do
4     generate Cluster  $\mathbf{C}_f$  and update  $\mathbf{E}$  by calling Alg. 3 with inputs
        $\mathbf{E}$ ,  $\rho_E$ ,  $\rho_T$  and  $q$ ;
5      $f \leftarrow f + 1$ ;
6   foreach  $\mathbf{C}_f$  do
7     calculate  $RSD_f$  using Eq. (8);
8     if  $RSD_f \leq T_{RSD}$  then
9        $\mathbf{C}_g^* \leftarrow \mathbf{C}_f$ ;  $g \leftarrow g + 1$ ;
10       $\mathcal{C}^* = \mathcal{C}^* \cup \{\mathbf{C}_g^*\}$ 
11    else
12       $\mathbf{E} \leftarrow \mathbf{E} \setminus \mathbf{C}_f$ ;
13   $\rho_E \leftarrow \rho_E/2$ ;  $f \leftarrow 1$ ;
14 foreach  $\mathbf{C}_h^* \in \mathcal{C}^*$  do
15   label  $\mathbf{C}_h^*$  as in [18], by comparing its signature with an existing
     database of appliance signatures;
16 calculate  $\hat{\mathbf{E}}^m$  for all disaggregated appliances using appliance power
     ratings as in [18];
17 return  $\hat{\mathbf{E}}^m$ 
```

clusters since signal changes are used to build the graph. However, since we use hourly aggregate profile $\mathbf{E} \in \mathbb{R}^+ \cup \{0\}$ to build the graph in this paper, feature matching is not needed. Finally, as in [18], each disaggregated cluster is labelled as one category by comparing with appliance profile, and corresponding appliance-level load is estimated. Note that time (not time-of-day) is used as a

feature in [18], firstly for pairing ON/OFF events in feature matching, but not for clustering, and secondly, edges of the graph in [18] represent the correlation of duration among ON/OFF candidates, while in this paper each edge in graph \mathcal{G}^T represents time-of-day difference of two corresponding samples.

3.4. Convolution neural network approach

Motivated by recent deep neural network architectures [9], [28] and [30] for low-rate power disaggregation, we implement a CNN architecture for estimating appliance hourly profile after training on very low-rate aggregate profile data. To enhance disaggregation performance on hourly profile, we explicitly use time-of-day information as an additional feature.

As shown in Fig. 1, our proposed CNN architecture aims to disaggregate the hourly power consumption per appliance per sample. The proposed network takes two inputs: (1) hourly aggregate power consumption in a Q -hour sliding window; (2) encoded cyclical continuous absolute time features of the fourth hour in the window. In this paper, we heuristically set window length $Q = 7$ hours to cover at least two working periods of washing machine. Given a sequence of 7 hours aggregate power consumption, we employ 3 CNN blocks to extract the spatial features correlated with the target hour (the fourth hour in the sliding window). To explore the non-uniform distribution of appliance usage, we encode the hour of the day ($H = [0, 23]$, $H \in \mathbb{R}^+ \cup \{0\}$) into two cyclical continuous variables $\sin(2\pi H/24)$ and $\cos(2\pi H/24)$ as the additional time features (encoded hour). Instead of feeding two time features into the network at the first layer, we merge the target hour time features with the corresponding spatial features extracted by the 3 CNN blocks by downsampling in the time domain. To effectively train the network, we adopt ‘skip connection’ to merge the outputs of the 3 CNN blocks with the time features that accelerates the training, enabling an ensemble of 7, 5, 3- hour profile feature maps with increasing CNN filters. The overall merged features are fed into the last four fully connected layers (128-256-64-1 neurons) to estimate the power consumption of the target hour. Moreover, we also adopt batch normalization [59], dropout [61], leaky

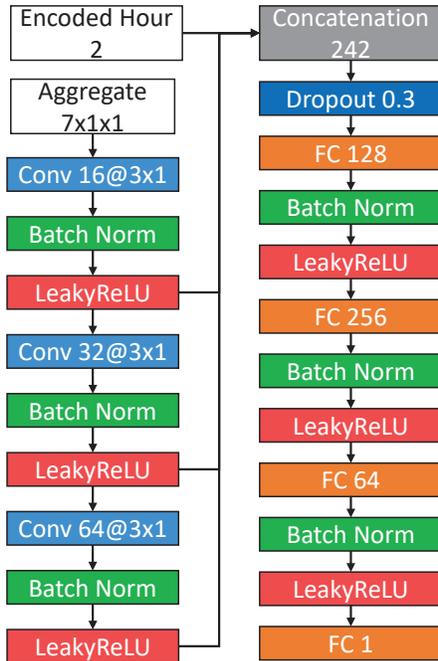


Figure 1: Proposed CNN architecture for very low-rate disaggregation. Both windows of 7-hour long aggregate data and the fourth hour’s encoded time information are fed into the proposed network. ‘Conv x@y’ refers to a 2D convolutional layer with x filters each with kernel size y and fixed stride size=1x1. ‘FC x’ means the fully connected layer with x=number of neurons. ‘Concatenation x’ refers to a flattening operation that reshapes a matrix into a vector array with x values. ‘Batch Norm’ is a batch normalisation layer [59]. ‘LeakyReLU’ refers to a leaky version of a rectified linear unit activation function [60], with a fixed slope coefficient=0.3. ‘Dropout x’ means that the dropout layer [61] randomly sets a fraction rate=x of input units to zero at each update during the training phase.

ReLU [60] and l_2 -norm penalty on weights [62] to obtain better regularization performance regardless of weight initialization.

4. Experimental Setup

4.1. Hourly energy presentation calculation

Due to the difference in the literature in the way downsampling is implemented on higher rate datasets to obtain a lower rate sample, first we clarify

how hourly energy profile data can be obtained from existing public datasets of power measurements with resolution of higher than 1 minute.

For a daily window, we denote $\mathbf{E}_{1:24}$ to represent 24 hourly load profile samples, e.g., E_{10} denotes total power consumption from 9:00AM to 10:00AM. Thus, the load profile is the integral of power over time.

4.1.1. Downsampling assuming fixed sampling intervals

In [48] and [49], very low-rate power consumption, in Watts, is calculated as the average of mean power values over N samples:

$$E_T = \frac{1}{N-1} \sum_{i=2}^N P_i. \quad (9)$$

Note that we start at t_2 , i.e., $i = 2$, because t_1 is outside the sampling period, as shown in Fig. 2.

4.1.2. Downsampling allowing variable sampling intervals

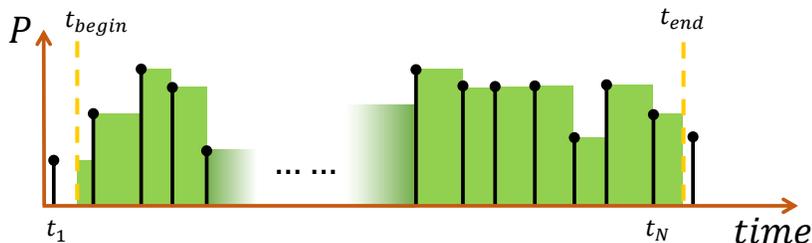


Figure 2: Hourly profile generation example.

Downsampling by averaging, however, assumes that the sampling rate is always fixed. However, smart meters, in practice, do not always sample at fixed intervals and vary slightly, e.g., within a range of 6-8 seconds [41]. Therefore, in order to account for variability in sampling rates and ensuring accurate estimation of power consumption over a period comprising non-uniform sampling rates, we calculate hourly consumption as follows. Referring to Fig. 2, the aim is to calculate the hourly profile E_T between t_{begin} and t_{end} , as shown in Eq. (10).

$$E_T = P_1 \times (t_2 - t_{begin}) + \sum_{i=2}^{N-1} P_i \times (t_{i+1} - t_i) + P_N \times (t_{end} - t_N). \quad (10)$$

The load profile unit is empirically chosen as *Watt · hour (Wh)* instead of *kW · h* or *J*. The appliance-level load profile is calculated in the same way as in Eq. (10) for validation purposes.

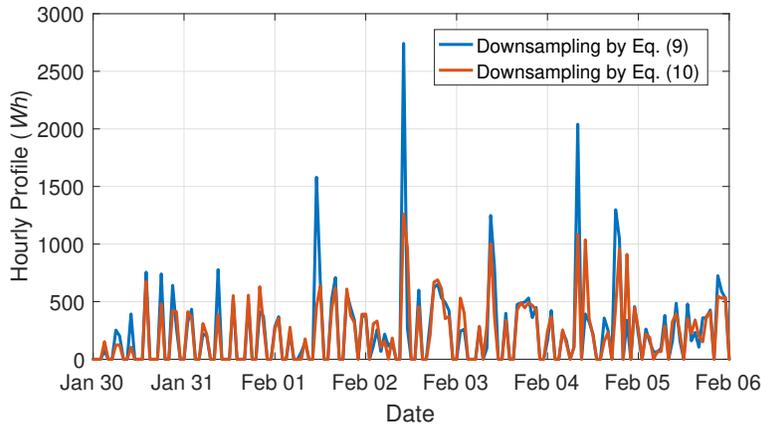


Figure 3: Hourly profile of House 4 from REFIT dataset calculated for a typical week.

Fig. 3 demonstrates that there is a difference between hourly electricity profile signals E_T calculated using the two downsampling approaches over one week (30/01/2014–05/02/2014) of smart meter readings from REFIT House 4.

4.2. Metrics

Since our NILM problem here is how reliably we can estimate the relative contribution of individual loads contributing to the aggregate consumption, we use consumption accuracy, as used in [18], [28] and [40], and calculated as in Eq. (11), where \hat{E}_d^m is defined as Eq. (12).

$$Acc^m = 1 - \frac{\sum_{d=1}^N |\hat{E}_d^m - E_d^m|}{2 \sum_{d=1}^N E_d^m}, \quad (11)$$

$$\hat{E}_d^m = \sum_{h=1}^{24} \hat{E}_{(d-1)*24+h}^m, \quad (12)$$

where Acc^m , also referred to as *TECA* (Total Energy Correctly Assigned) in the NILM literature, is the total error in assigned energy, normalized by the actual energy consumption in each time slice averaged over all appliances, that is, Acc^m demonstrates the error between actual daily load profile for Appliance m , E_d^m , and its disaggregated estimate \hat{E}_d^m , for $d \in Z$ and $d \in [1, N]$, where N is the total number of days of testing data.

We also utilise match rate, a metric where the evaluation is based on the overlapping rate of true and estimated energy, and stated to have the best overall performance in [63] and references therein. It was also used in [43] for very low-rate disaggregation. We denote our daily disaggregation match rate as MR^m , defined in Eq. (13), which shows the overall matching accuracy between actual and estimated daily energy consumption for Appliance m .

$$MR^m = \frac{\sum_{d=1}^N \min\{\hat{E}_d^m, E_d^m\}}{\sum_{d=1}^N \max\{\hat{E}_d^m, E_d^m\}}. \quad (13)$$

Solutions for the very low-rate disaggregation problem are unlikely to identify sample-by-sample load as easily and accurately as those at higher granularity, due to unavailability of sharp power change features, numerous simultaneous operation of multiple loads, significant negative influence of noise, etc. Based on the definition of metrics and the nature of this problem, MR^m is regarded as a better metric than Acc^m [63]. From [18, 28, 40], Acc^m is a good metric for demonstrating the error between estimation and actual measurements when the error is generally small. For very low-rate disaggregation where overestimation is common, especially for real-world noisy datasets, Acc^m could be negative. E.g., assume that there exist two disaggregators: Disaggregator 1 estimates total load consumption for Appliance m close to the actual consumption but not correctly assigned sample-by-sample in each time instance, with $Acc^m < 0$; Disaggregator 2 fails to disaggregate Appliance m and returns nothing, but the resulting $Acc^m = 0.5$ by Eq. (11). Such Acc^m results intuitively mean Disaggre-

gator 1 performs much worse than Disaggregator 2 on Appliance m . However, at least Disaggregator 1 is able to offer appliance-level total energy usage feedback, while Disaggregator 2 feedback is meaningless. On the contrary, $MR^m \in [0, 1]$ acts as a better overall evaluation presentation, where this value tending to 0 indicates a poor match between estimated and actual energy consumed.

4.3. Experimental data

The REFIT dataset [41] is used for evaluation. This dataset was chosen for the following reasons: (i) supported by NILMTK and used in recent literature, [18, 39, 40, 64], to facilitate benchmarking with other NILM solutions; (ii) large dataset with aggregate and sub-metering data from 20 houses over a continuous period of 2 years; (iii) this dataset was collected in multiple households with numerous unlabelled appliances, while inhabitants carried out their daily routines under no test conditions, and is therefore more challenging but also more representative of the average household.

The hourly experimental load profile data is generated using Eq. (10). For all results presented, experiments were carried out for REFIT Houses 4 and 8, with a low-level and a higher-level of unlabelled appliance noise [20], 0.02 and 0.24, respectively. The experimental period of testing and training (where applicable) is presented in Table 2.

Table 2: Experimental data selection for training and testing

	House 4	House 8
Training period	12/10/2013 – 03/01/2014 (12 weeks)	02/11/2013 – 25/01/2014 (12 weeks)
Testing period	04/01/2014 – 03/07/2015 (78 weeks)	26/01/2014 – 09/05/2015 (67 weeks)

4.4. Benchmark Setup

While the NILMTK toolbox [19] has an embedded resampling tool to generate hourly power samples by picking the last sample in each hour, this does not

result in a true hourly load profile and aggregated hourly consumption is lost. Instead, we import our hourly profile with corresponding timestamps into the NILMTK for benchmarking against FHMM and CO. All parameters in DDSC implementation in this paper are chosen as per [27].

4.5. Appliances disaggregated

The following domestic loads are disaggregated and denoted in abbreviated form in the rest of this paper: F for Fridge; FZ for Freezer; FFZ for Fridge-freezer; WM for Washing machine; M for Microwave; K for Kettle; WD for Washer dryer; T for Toaster.

5. Experimental results

In this section, experimental results of all proposed and benchmarked hourly NILM approaches are presented for REFIT Houses 4 and 8, for periods shown in Table 2, and using the following evaluation metrics: daily Acc^m and daily MR^m .

5.1. OPT Parameters

For OPT, we split the sequence into 17-hour windows, that is, $\hat{E}_{seq} \leq 17$, and set $k = 4$ for all appliances in the experiments below, i.e., OPT will assume that each appliance will not run more than four times in each 17-hour window to trade-off complexity and performance.

In terms of metadata required for OPT, only wattage or energy-per-use-per-run is needed. Make and model are only used if wattage and energy-per-use-per-run are unknown. For appliances with more or less constant operational power range, such as M and K, the energy-consumption-per-run, \bar{W}^m , can be estimated as the product of wattage and average duration per use. Otherwise, for multi-state appliances with preset programmes, such as WM, the energy-consumption-per-run is usually available from the manufacturer. If the wattage of an appliance is not available or the energy-consumption-per-run cannot be estimated for an older model with varying operational power, the wattage and

Table 3: Example of metadata needed for OPT disaggregation, obtained from manufacturer or user manual for REFIT House 4. Note that make and model are only needed to determine Wattage or Energy per use, if they are not known to the user.

Appliance type	WM		TV	M	K
Make	Servis	Zanussi	Sony	Matsui	Swan
Model	6065	Z917	KDL-32W706B	170TC	Unknown
Average duration	2-3hours		-	280sec	130sec
Wattage:	2200W	2730W	39W/80W	650W	2000W
Energy-consumption-per-run	760Wh	-	-	-	-

\bar{W}^m can be estimated according to the manufacturer information from make and model information. For example, the metadata obtained via an appliance survey for the appliances being disaggregated in House 4 are shown in Table 3.

The parameters for the appliance models, defined in Section 3.2.2, are shown in Tables 4 and 5. As discussed in Section 3.2.2, PC and TV belong to appliance category \mathcal{N}_2 , where \bar{W}^m for PC or TV refers to the average total energy consumption over a period of one hour.

Table 4: Appliance models for REFIT House 4.

Appliance	WM	PC	TV	M	K
\bar{W}^m (Wh/cycle)	766	76	69	51	73
\bar{T}^m	3	n/a	n/a	2	2

Table 5: Appliance models for REFIT House 8.

Appliance	WD	WM	T	PC	TV	M	K
\bar{W}^m (Wh/cycle)	521	874	61	55	132	97	111
\bar{T}^m	3	3	2	n/a	n/a	2	2

For each appliance $m \in \mathcal{N}_1$, \bar{T}^m represents the maximum duration in samples. E.g., WM and DW last 1-2 hours during one operation, and thus \bar{T}^m is set to 3, as shown in Tables 4 and 5.

As explained in Section 3.2.1, OPT estimates energy consumed by F, FZ, FFZ and baseload via always-on consumption estimation steps and therefore results show always-on load performance. Results for GSP are also presented by grouping F, FZ and FFZ due to similarity of F, FZ and FFZ profile values obtained during GSP clustering.

5.2. GSP parameters

The scaling factors for weighting graph edges: ρ_E is initialised to 10 and ρ_T is fixed to 0.005. We set κ to $\rho_E/2^{10}$. Trade-off factor λ is empirically set to 0.5. We define $T_{RSD} = 10\%$ and $q = 0.98$ as in [18], for maintaining high cluster quality. In order to mitigate long execution time, as reported in [49], we set the upper limit of window size in GSP clustering to 1344, which is equivalent to a period of 8 weeks for hourly profile measurements.

5.3. CNN parameters

The proposed CNN network is trained by Adam optimiser via mean square error (MSE) loss for up to 60 epochs with a learning rate of 0.0001. The hourly power consumption and individual measurements are standardized by subtracting the mean, then dividing by the standard deviation. During each epoch, all training data are fed into the network with a batch size of 128. Each training batch is sampled with a distribution of 50/50 ON/OFF states for each appliance to reduce the effect of biased predictions made for infrequently used appliances.

5.4. Daily consumption accuracy and match rate performance

Daily disaggregation accuracy Acc^m performance for Houses 4 and 8 are presented in Tables 6 and 7, respectively. Daily match rate MR^m results can be observed in Tables 8 and 9 for Houses 4 and 8, respectively.

DDSC performance is slightly worse than reported results in [26] but this is expected as results in Tables 6 to 9 were tested on houses containing many unlabelled appliances as opposed to aggregate of labelled appliances in [26]. Since DDSC attempts to learn appliance dictionaries by mitigating the difference

Table 6: Daily Acc^m results for REFIT House 4.

App.	F	FZ	FFZ	WM	PC	TV	M	K
FHMM	0.91	0.91	0.86	0.2	-	0.48	0.17	-
CO	0.92	0.87	0.88	-	0.27	0.59	-	-
DDSC	0.33	0.3	-	0.39	-	-	0.1	0.08
GSP	0.87			0.63	0.5	-	0.59	0.65
OPT	0.94			0.41	0.66	0.68	0.67	0.68
CNN	0.93	0.94	0.93	0.43	0.71	0.73	0.55	0.67

Table 7: Daily Acc^m results for REFIT House 8.

App.	F	FZ	WD	WM	T	PC	TV	M	K
FHMM	0.72	0.54	0.17	-	-	0.65	-	-	-
CO	0.51	0.69	0.5	-	-	0.56	0.24	-	-
DDSC	-	-	0.46	0.5	0.5	-	0.42	0.5	-
GSP	0.85		0.27	0.59	0.56	0.54	0.52	0.75	0.54
OPT	0.73		0.4	0.57	0.28	0.56	0.81	0.72	0.81
CNN	0.88	0.85	-	0.58	0.46	0.92	0.85	0.51	0.87

between the weekly profile and the production of dictionaries and appliance-level weekly activation, it is sensitive to noise caused by unlabelled appliances. The results explain why the sum of sub-metering measurements is used instead of real aggregate readings in [26], as discussed in Section 2. DDSC is also observed to have the worst performance of all benchmarks and proposed algorithms.

Table 6 shows that FHMM performs well in estimating consumption accuracy for F, FZ and FFZ, which is in line with NILM results reported previously [17], [18] and [19]. However, in noisier House 8, as shown in Table 7, FHMM is not so robust. A similar observation is made for CO. However, GSP is the most robust algorithm against noise for these always-on appliances. OPT and CNN perform as well as FHMM and CO on average for these always-on appliances. However, as observed in Tables 6 and 7, FHMM and CO cannot disaggregate

all appliances compared to OPT and CNN.

GSP is also good at disaggregating all appliances, except TV in House 4 because TV has far higher usage frequency than other appliances. Clusters labelled as TV contain a large amount of unclassified or unlabelled loads, resulting in overestimation for TV but underestimation for other appliances, as shown in Fig. 6. However, in House 8, the usage frequencies for different appliances are more balanced. Hence, fewer events are falsely clustered as TV and more loads can be correctly identified in House 8 than in House 4, such as M. As PCs are always-on for both houses with low hourly consumption values, GSP fails disaggregating such small loads, also shown in Figs. 6 and 7. However, FHMM and CO achieve good performance for PC in both houses as expected, in both metrics and Figs. 6 and 7.

As shown in Tables 6 and 7, CNN outperforms others in estimating daily consumption accuracy for always-on and long-lasting appliances (F, FZ, FFZ, TV, PC) in both houses. CNN is especially good at estimating consumption of PC and TV, that operate in a house-dependant hourly usage pattern, because it incorporates time-of-day information.

We observe that WM and WD Acc^m performance is relatively poorer than for other appliances. Similarly, daily match rate MR^m results can be observed from Tables 8 and 9, respectively. As defined in Subsection 3.2.2, the load profile of such appliances $m \in \mathcal{N}_1$ varies based on sampling instances and is split into consecutive aggregate samples. WM and WD have operation cycles longer than one sampling period. As loads are identified per non-zero sequence in OPT instead of per sample as in other benchmarking methods, although OPT does not perform so well with Acc^m and MR^m for WM in House 4, its energy consumption estimation is the closest to the energy actually consumed.

For most appliances, MR^m results are inline with Acc^m results. Both Acc^m and MR^m results show the reliability of disaggregation for always-on appliances such as F, FZ and FFZ compared to other appliances. Recall that Acc^m focuses on estimation tolerance *per sample* or per window defined as per day in this paper, whereas MR^m focuses on *overall* estimation tolerance. The latter

Table 8: Disaggregation daily MR^m results (%) for REFIT House 4.

App.	F	FZ	FFZ	WM	PC	TV	M	K
FHMM	85	84	76	7	21	44	34	13
CO	85	74	76	26	38	54	13	12
DDSC	3	24	16	1	6	6	5	3
GSP	80			36	0	29	21	34
OPT	87			26	49	56	49	54
CNN	86	90	87	41	55	64	50	55

Table 9: Disaggregation daily MR^m results (%) for REFIT House 8.

App.	F	FZ	WD	WM	T	PC	TV	M	K
FHMM	63	52	12	13	16	58	24	24	18
CO	50	61	0	12	6	53	40	12	29
DDSC	3	5	5	9	0	5	9	0	5
GSP	72		15	47	18	8	5	56	7
OPT	65		9	30	38	11	64	61	63
CNN	79	70	17	46	43	85	75	49	79

demonstrates a better metric to assess disaggregation performance as shown by Figs. 4 to 7, which show estimated energy consumed vs ground truth. E.g., for House 4, MR^m of GSP for PC is 0, inline with Fig. 6 - these indicate that GSP fails to disaggregate PC. However, if considering only Acc^m : GSP for PC in House 4 is similar to those of OPT or CNN. However, the total PC consumption estimation from OPT and CNN is close to the actual ground truth.

5.5. Analysis of estimated vs. actual energy consumption to explain metrics

First, we look at always-on appliances, as shown in Figs. 4 and 5, which illustrate the estimated disaggregated energy with respect to the ground truth. Note that OPT grouped disaggregation of refrigeration appliances with baseload (BL) as always-on loads; GSP clustered refrigeration appliances during disag-

gregation due to similarity in energy signature. The purpose of these figures, together with Figs. 6 and 7 which show energy estimated w.r.t ground truth for all other appliances, is to better understand overestimation and underestimation of consumption during disaggregation because this is not fully captured by the performance metrics.

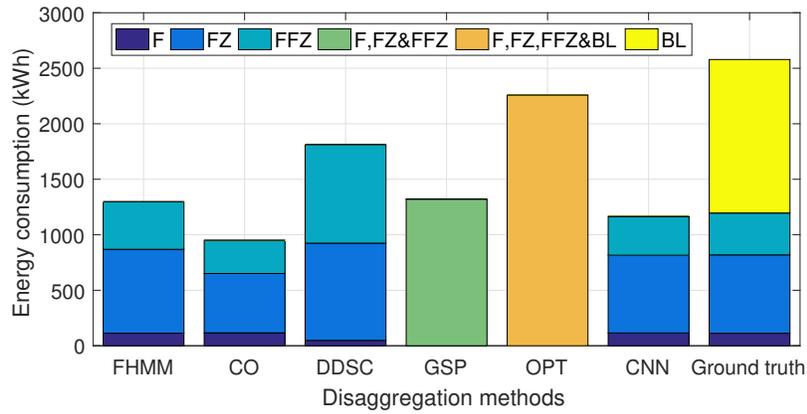


Figure 4: Always-on appliance disaggregation for REFIT House 4.

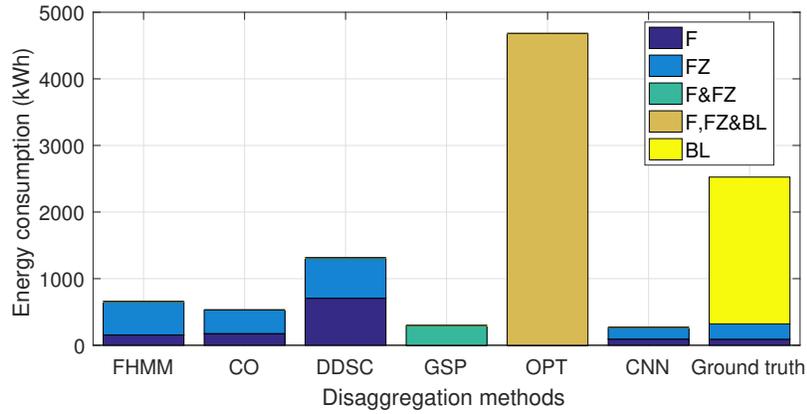


Figure 5: Always-on appliance disaggregation for REFIT House 8.

DDSC performance is poor because it significantly overestimates freezer consumption as shown in Figs. 4 and 5. FHMM and CO generally slightly overesti-

estimate F, FZ and FFZ consumption for both houses (except CO which underestimates F, FZ and FFZ for House 4). Both GSP and CNN consistently correctly estimate F, FZ and FFZ for both houses. While OPT reasonably estimates F, FZ, FFZ and BL, OPT overestimates always-on loads for House 8 due to 'noise' from unlabelled appliances contributing significantly to BL at odd hours.

Next, we discuss relative performance of all algorithms in terms of energy consumption of other appliances, namely washing machine, washer dryer, PC, TV, microwave, toaster and kettle, as shown in Figs. 6 and 7. In both houses, all benchmarking algorithms significantly overestimate or underestimate energy consumption with respect to the ground truth for the majority of appliances $m \in \mathcal{N}_1$. This is in agreement with the Acc^m and MR^m results, which have generally poorer performance compared to proposed algorithms. Although these approaches have sufficient sub-metering data for training, the lack of unlabelled appliance models or a noise model results in overestimation. This weakness is not obvious when the data is clean, i.e., where unlabelled loads are non-existent as in [26, 49]. In particular, overestimation is the most significant in CO performance of all approaches used, as its basis is ideal disaggregation as Eq.(1) with generally small measurement noise and all appliance being labelled. Inline with Acc^m and MR^m , the bar charts indicate DDSC is the most susceptible to the influence of unlabelled loads among all benchmarks for both houses. DDSC estimates the energy consumption of M and K which is close to the ground truth for House 4, while for 'noisier' House 8 containing more unlabelled loads, it suffers from overestimation of K.

While GSP performs well for both houses for high consuming and long duration appliances such as WM and WD, at very low-rate, GSP suffers from underestimation of PC and M and overestimation for TV, as shown in Fig. 6. This is observed with Acc^m almost 0.5 and $MR^m < 30$ for PC and M. GSP is also unreliable for PC, TV and K for House 8, with significant underestimation.

Proposed OPT and CNN both generally perform best for all appliances. From Figs. 6 and 7, OPT performs better for House 4 while CNN performs better for House 8. For both houses, supervised CNN slightly outperforms

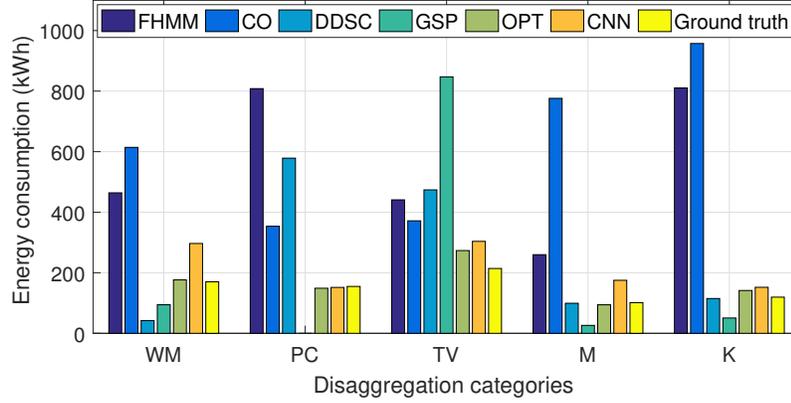


Figure 6: Disaggregation performance for other appliances in REFIT House 4.

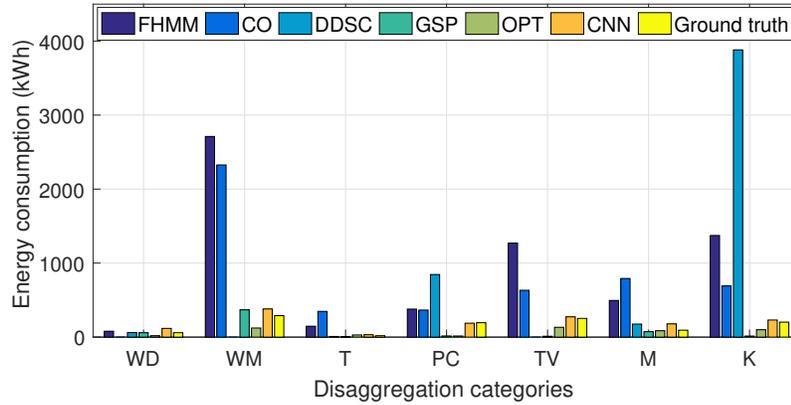


Figure 7: Disaggregation performance for other appliances in REFIT House 8.

unsupervised OPT for most appliances apart from M. Although in CNN the training set for M is sufficient due to availability of many M activations, the periodicity is overly learnt, resulting in overestimation for M. Indeed, \hat{E}^m obtained by CNN for each appliance contains more or less periodic components, thus underestimation is not observed in the CNN results.

The energy contributed by unlabelled always-on appliances is expected to be disaggregated in the always-on load estimation step of the proposed OPT algorithm, which can sometimes be overestimated as for the case of House 8. The

remaining appliances which are unlabelled might be falsely detected as labelled appliances with similar wattage/energy-consumption-per-run. However, the use of model parameter \bar{T}^m reduces the likelihood of such false assignments.

5.6. Experimental execution time

The experiments were performed on an Intel i7-4710HQ CPU 2.5GHz machine, running Windows10. OPT, GSP and DDSC are implemented in Matlab2016a. CNN, as well as FHMM and CO, embedded in NILMTK toolkit, are implemented in Python.

Table 10: Execution time

		FHMM	CO	DDSC	GSP	OPT	CNN
House 4	Training(12 weeks)	4s	1s	4hours	-	-	48s
	Testing(78 weeks)	36min	<1s	3hours	7min	2min	1s
House 8	Training(12 weeks)	12s	<1s	5hours	-	-	47s
	Testing(67 weeks)	6hours	<1s	4hours	7min	2min	1s

The execution time is shown in Table 10 for all methods, where the runtime for FHMM and CO is inline with those reported in [49]. The CO approach, with the lowest complexity, is always the fastest. DDSC takes the longest time, mainly due to the complexity of its iterative optimization for dictionary and activation matrices. Note that in [49], DDSC is implemented in a more efficient Python implementation, hence a difference in execution times. Note that the proposed unsupervised GSP-based approach takes a few minutes to execute, which is in contrast to the long execution time reported in [49]. Moreover, both OPT and CNN trade-off efficacy and efficiency, and execute within short periods of time.

6. Conclusion

This paper addresses the gap in load disaggregation at very low-rates, particularly without resorting to extensive metadata (i.e., only using smart meter

data at hourly granularity as input to our proposed algorithms, and wattage information of appliances obtained from an appliance survey). This is especially critical now as it is emerging that nationwide smart meter roll-outs worldwide will provide 15min-1hour measurements and not at higher resolution as previously thought. Consequently, we demonstrate, on realistic datasets, that very low-rate load disaggregation is possible for a range of appliances. The contributions of this paper are a detailed review of the current-state-of-the-art in very low-rate load disaggregation or non-intrusive load monitoring, two proposed unsupervised algorithms and one proposed supervised NILM approach, testing on 2 years of real-world noisy REFIT dataset and benchmarked on three commonly used load disaggregation methods, use of different metrics to evaluate disaggregation performance. Results for all proposed and benchmark methods are provided using energy consumption accuracy and match rate metrics, as well as showing estimated vs actual energy consumption per appliance disaggregated, in order to better understand overestimation or underestimation of energy consumption for particular appliances. Execution time is also included as a metric to assess practicality of the methods. Compared to benchmark algorithms (FHMM, CO and DDSC), GSP, OPT and CNN provide better disaggregation performance for all appliances. Although time-of-day (extracted from aggregate data) is utilised as a feature for both CNN and GSP, CNN outperforms GSP through deep learning of features. OPT benefits from modelling and removal of permanent consumption in advance, and has similar performance to CNN. Note, however, that unsupervised OPT only requires manufacturer’s wattage information whereas supervised CNN requires sub-metering data for training, which is not always available, Therefore the proposed approaches have been shown to disaggregate effectively a wide range of loads using true aggregate smart meter measurements containing many unlabelled appliances. White goods such as refrigeration appliances, washing machine and washer dryer and a few other household appliances can be reliably disaggregated at very low-rate smart meter data, with no additional metadata besides appliance wattage information. Additionally, the low complexity and short execution time of the

proposed energy disaggregation solutions would result in a low implementation investment for energy disaggregation solution providers and their customers. Therefore, while return on investment is a separate study in its own right, we are confident that the benefits outweigh the investments costs in the medium to long term.

Future work will investigate more scalable or transferable approaches, comprising unsupervised and supervised methods, that can work with minimal training data and can work on any unseen dataset. For the proposed OPT, it would be beneficial to undertake a sensitivity analysis to assess the effect of noise-level, i.e., the percentage of energy consumed by unlabelled appliances, on disaggregation performance.

7. Acknowledgement

This project is co-funded by the European Commission under the “H2020-EU.3.3.1. - Reducing energy consumption and carbon footprint by smart and sustainable use” program topic, according to the Grant agreement No. 767625. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp used for this research.

8. References

- [1] A. Krishnan and A. Markkanen, “Smart metering devices and services update,” ABI Research, Oyster Bay, NY, USA, Tech. Rep. AN-1558, 2015.
- [2] E. Commission, “Smart grids and meters,” <https://ec.europa.eu/energy/en/topics/markets-and-consumers/smart-grids-and-meters>, July 2014, updated Aug. 22, 2019.
- [3] G. Zhang, G. G. Wang, H. Farhangi, and A. Palizban, “Data mining of smart meters for load category based disaggregation of residential power consumption,” *Sustainable Energy, Grids and Networks*, vol. 10, pp. 92–103, 2017.

- [4] A. Bahmanyar, S. Jamali, A. Estebarsari, E. Pons, E. Bompard, E. Patti, and A. Acquaviva, “Emerging smart meters in electrical distribution systems: Opportunities and challenges,” in *2016 24th Iranian Conference on Electrical Engineering (ICEE)*. IEEE, 2016, pp. 1082–1087.
- [5] A. hoc group of the Expert Group 1 – Standards and Interoperability, “My energy data,” European Smart Grids Task Force, Tech. Rep., Nov. 2016, https://ec.europa.eu/energy/sites/ener/files/documents/report_final_eg1_my_energy_data_15_november_2016.pdf.
- [6] V. Y. Pillitteri and T. L. Brewer, “Guidelines for smart grid cybersecurity,” NIST Interagency, Tech. Rep. (NISTIR)-7628 Rev 1, 2014.
- [7] “Smart metering equipment technical specifications version 2,” Department of Energy and Climate Change, UK, Tech. Rep., Jan. 2013, https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/68898/smart_meters_equipment_technical_spec_version_2.pdf.
- [8] J. Kelly and W. Knottenbelt, “Does disaggregated electricity feedback reduce domestic electricity consumption? a systematic review of the literature,” *arXiv preprint arXiv:1605.00962*, 2016.
- [9] C. Zhang, M. Zhong, Z. Wang, N. Goddard, and C. Sutton, “Sequence-to-point learning with neural networks for non-intrusive load monitoring,” in *Thirty-second AAAI conference on artificial intelligence*, 2018.
- [10] 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.
- [11] L. Stankovic, V. Stankovic, J. Liao, and C. Wilson, “Measuring the energy intensity of domestic activities from smart meter data,” *Applied Energy*, vol. 183, pp. 1565–1580, 2016.

- [12] D. Murray, J. Liao, L. Stankovic, and V. Stankovic, "Understanding usage patterns of electric kettle and energy saving potential," *Applied energy*, vol. 171, pp. 231–242, 2016.
- [13] D. Murray, L. Stankovic, V. Stankovic, and N. Espinoza-Orias, "Appliance electrical consumption modelling at scale using smart meter data," *Journal of Cleaner Production*, vol. 187, pp. 237–249, 2018.
- [14] D. Christensen, L. Earle, and B. Sparn, "Nilm applications for the energy-efficient home," National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2012.
- [15] H. Rashid, P. Singh, V. Stankovic, and L. Stankovic, "Can non-intrusive load monitoring be used for identifying an appliances anomalous behaviour?" *Applied Energy*, vol. 238, pp. 796–805, 2019.
- [16] S. M. Tabatabaei, S. Dick, and W. Xu, "Toward non-intrusive load monitoring via multi-label classification," *IEEE Transactions on Smart Grid*, vol. 8, no. 1, pp. 26–40, 2017.
- [17] K. He, L. Stankovic, J. Liao, and V. Stankovic, "Non-intrusive load disaggregation using graph signal processing," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1739–1747, 2018.
- [18] 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.
- [19] N. Batra, J. Kelly, O. Parson, H. Dutta, W. Knottenbelt, A. Rogers, A. Singh, and M. Srivastava, "Nilmk: 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.
- [20] 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.

- [21] K. Basu, V. Debusschere, S. Bacha, U. Maulik, and S. Bondyopadhyay, “Nonintrusive load monitoring: A temporal multilabel classification approach,” *IEEE Transactions on industrial informatics*, vol. 11, no. 1, pp. 262–270, 2015.
- [22] J. Liao, G. Elafoudi, L. Stankovic, and V. Stankovic, “Non-intrusive appliance load monitoring using low-resolution smart meter data,” in *2014 IEEE International Conference on Smart Grid Communications (Smart-GridComm)*. IEEE, 2014, pp. 535–540.
- [23] J. Lines, A. Bagnall, P. Caiger-Smith, and S. Anderson, “Classification of household devices by electricity usage profiles,” in *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 2011, pp. 403–412.
- [24] B. Zhao, L. Stankovic, and V. Stankovic, “Electricity usage profile disaggregation of hourly smart meter data,” in *4th International Workshop on Non-Intrusive Load Monitoring*, 2018.
- [25] N. Batra, A. Singh, and K. Whitehouse, “Gemello: Creating a detailed energy breakdown from just the monthly electricity bill,” in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2016, pp. 431–440.
- [26] J. Z. Kolter, S. Batra, and A. Y. Ng, “Energy disaggregation via discriminative sparse coding,” in *Advances in Neural Information Processing Systems*, 2010, pp. 1153–1161.
- [27] M. Figueiredo, B. Ribeiro, and A. de Almeida, “Electrical signal source separation via nonnegative tensor factorization using on site measurements in a smart home,” *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 2, pp. 364–373, 2014.
- [28] 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.

- [29] A. Ebrahim and O. Mohammed, “Pre-processing of energy demand disaggregation based data mining techniques for household load demand forecasting,” *Inventions*, vol. 3, no. 3, pp. 1–17, 2018.
- [30] D. Murray, L. Stankovic, V. Stankovic, S. Lulic, and S. Sladojevic, “Transferability of neural network approaches for low-rate energy disaggregation,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 8330–8334.
- [31] K. Kumar and M. G. Chandra, “Event and feature based electrical load disaggregation using graph signal processing,” in *2017 IEEE 13th International Colloquium on Signal Processing & its Applications (CSPA)*. IEEE, 2017, pp. 168–172.
- [32] D. Piga, A. Cominola, M. Giuliani, A. Castelletti, and A. E. Rizzoli, “Sparse optimization for automated energy end use disaggregation,” *IEEE Transactions on Control Systems Technology*, vol. 24, no. 3, pp. 1044–1051, 2016.
- [33] G. Tang, Z. Ling, F. Li, D. Tang, and J. Tang, “Occupancy-aided energy disaggregation,” *Computer Networks*, vol. 117, pp. 42–51, 2017.
- [34] G. W. Hart, “Nonintrusive appliance load monitoring,” *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.
- [35] A. Kipping and E. Trømborg, “Modeling and disaggregating hourly electricity consumption in norwegian dwellings based on smart meter data,” *Energy and Buildings*, vol. 118, pp. 350–369, 2016.
- [36] B. J. Birt, G. R. Newsham, I. Beausoleil-Morrison, M. M. Armstrong, N. Saldanha, and I. H. Rowlands, “Disaggregating categories of electrical energy end-use from whole-house hourly data,” *Energy and Buildings*, vol. 50, pp. 93–102, 2012.

- [37] 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.
- [38] S. Singh and A. Majumdar, “Deep sparse coding for non-intrusive load monitoring,” *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4669–4678, 2018.
- [39] M. A. Mengistu, A. A. Girmay, C. Camarda, A. Acquaviva, and E. Patti, “A cloud-based on-line disaggregation algorithm for home appliance loads,” *IEEE Transactions on Smart Grid*, 2018.
- [40] B. Zhao, K. He, L. Stankovic, and V. Stankovic, “Improving event-based non-intrusive load monitoring using graph signal processing,” *IEEE Access*, vol. 6, pp. 53 944–53 959, 2018.
- [41] 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, no. 160122, 2017.
- [42] G. Zhang, G. Wang, H. Farhangi, and A. Palizban, “Residential electric load disaggregation for low-frequency utility applications,” in *2015 IEEE Power & Energy Society General Meeting*. IEEE, 2015, pp. 1–5.
- [43] A. Miyasawa, Y. Fujimoto, and Y. Hayashi, “Energy disaggregation based on smart metering data via semi-binary nonnegative matrix factorization,” *Energy and Buildings*, vol. 183, pp. 547–558, 2019.
- [44] A. De Almeida, P. Fonseca, B. Schlomann, N. Feilberg, and C. Ferreira, “Residential monitoring to decrease energy use and carbon emissions in europe,” in *International Energy Efficiency in Domestic Appliances & Lighting Conference*. Citeseer, 2006.
- [45] 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, no. Citeseer, 2011, pp. 59–62.

- [46] S. Makonin, B. Ellert, I. V. Bajić, and F. Popowich, “Electricity, water, and natural gas consumption of a residential house in canada from 2012 to 2014,” *Scientific data*, vol. 3, no. 160037, 2016.
- [47] V. Stankovic, J. Liao, and L. Stankovic, “A graph-based signal processing approach for low-rate energy disaggregation,” in *2014 IEEE symposium on computational intelligence for engineering solutions (CIES)*. IEEE, 2014, pp. 81–87.
- [48] K. Kumar and M. G. Chandra, “An intuitive explanation of graph signal processing-based electrical load disaggregation,” in *2017 IEEE 13th International Colloquium on Signal Processing & its Applications (CSPA)*. IEEE, 2017, pp. 100–105.
- [49] J. Holweger, M. Dorokhova, L. Bloch, C. Ballif, and N. Wyrsh, “Unsupervised algorithm for disaggregating low-sampling-rate electricity consumption of households,” *Sustainable Energy, Grids and Networks*, vol. 19, no. 100244, 2019.
- [50] N. Perraudin, J. Paratte, D. Shuman, L. Martin, V. Kalofolias, P. Vandergheynst, and D. K. Hammond, “Gspbox: A toolbox for signal processing on graphs,” *arXiv preprint arXiv:1408.5781*, 2014.
- [51] A. Monacchi, D. Egarter, W. Elmenreich, S. D’Alessandro, and A. M. Tonello, “Greend: An energy consumption dataset of households in italy and austria,” in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 2014, pp. 511–516.
- [52] 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.
- [53] O. Parson, G. Fisher, A. Hersey, N. Batra, J. Kelly, A. Singh, W. Knottenbelt, and A. Rogers, “Dataport and nilmtk: A building data set designed

- for non-intrusive load monitoring,” in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2015, pp. 210–214.
- [54] M. Dong, P. C. Meira, W. Xu, and C. Chung, “Non-intrusive signature extraction for major residential loads,” *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1421–1430, 2013.
- [55] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 2.1,” <http://cvxr.com/cvx>, Mar. 2014.
- [56] P. Tseng, “An infeasible path-following method for monotone complementarity problems,” *SIAM Journal on Optimization*, vol. 7, no. 2, pp. 386–402, 1997.
- [57] C. Dinesh, S. Makonin, and I. V. Bajic, “Incorporating time-of-day usage patterns into non-intrusive load monitoring,” in *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2017, pp. 1110–1114.
- [58] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.
- [59] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International Conference on Machine Learning (ICML)*, 2015, pp. 448–456.
- [60] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *International Conference on Machine Learning (ICML)*, vol. 30, no. 1, 2013, p. 3.
- [61] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *arXiv:1207.0580*, 2012.

- [62] P. Lemberger, “On generalization and regularization in deep learning,” *arXiv:1704.01312*, 2017.
- [63] L. Pereira and N. Nunes, “Performance evaluation in non-intrusive load monitoring: Datasets, metrics, and toolsa review,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 6, pp. 1–17, 2018.
- [64] P. Huber, M. Gerber, A. Rumsch, and A. Paice, “Prediction of domestic appliances usage based on electrical consumption,” *Energy Informatics*, vol. 1, no. 1, pp. 1–16, 2018.