An Unsupervised Load Disaggregation Approach based on Graph Signal Processing Featuring Power Sequences

Xuhao Li
lixuhao@tju.edu.cn
Tianjin University, China

Bochao Zhao
bochaozhao@tju.edu.cn
Tianjin University, China

Wenpeng Luan
wenpengluan@tju.edu.cn
Tianjin University, China

Bo Liu
liubo@tju.edu.cn
Tianjin University, China

ABSTRACT

Non-intrusive load monitoring (NILM) offers appliance-level electricity usage details via analysing aggregate power readings. Although graph signal processing (GSP) concepts have been applied to load disaggregation from low-rate power measurements in an unsupervised manner, the robustness of GSP-based NILM solutions can be enhanced by improving feature selection. In this paper, a method is proposed for extracting state transition sequence (STS) features from power readings, instead of power changes and steady-state power sequences featured in the existing works. By building a graph for the extracted STSs, clustering can be performed, where dynamic time warping is used to calculate correlation between STSs. Finally, the grouped STSs is matched for performing load disaggregation. Experiments are carried out on publicly-accessible AMPds and REFIT datasets, showing the proposed method generally outperforms two state-of-the-art benchmarks in various evaluation metrics.

CCS CONCEPTS

• **Information systems** → Information systems applications; Data mining; *Clustering*.

KEYWORDS

Non–intrusive load monitoring, graph signal processing, feature extraction

ACM Reference Format:
Xuhao Li, Bochao Zhao, Wenpeng Luan, and Bo Liu. 2022. An Unsupervised Load Disaggregation Approach based on Graph Signal Processing Featuring Power Sequences. In *6th International Workshop on Non-Intrusive Load Monitoring (NILM '22), November 9–10, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3563357.3566156

1 INTRODUCTION

Non-intrusive load monitoring (NILM) is a technique to disaggregate power consumed by each appliance and identify their operational states by analysing the aggregate power readings via pure software tools. The NILM concept was raised by G. W. Hart in 1980s [1], as a low-cost and user-friendly alternative to load monitoring sensors. By offering fine-grained electricity consuming feedback, including the categories, operational power ranges and usage habits of major appliances, NILM enriches smart metering benefits and supports demand-side management. According to the sampling rates of power readings to be disaggregated, NILM tasks can be labelled as high-rate (above 1Hz, usually in kHz and MHz) and low-rate (from 1/60 Hz to 1Hz)[2]. Although more knowledge can be acquired from high-rate measurements, the data collection is unavailable by commercial meters deployed worldwide for billing purposes. Therefore, feasible solutions to the NILM problem on low-rate measurements capture more attention recently. Low-rate NILM approaches can be supervised or unsupervised, depending on whether sub-metering data is required for training.

Current low-rate NILM solutions are based on hidden Markov model (HMM) and its variants [3], random forest (RF) [4], deep learning [5] and graph signal processing (GSP) [6, 7], etc. In general, HMM-based NILM approaches are competitive in identifying periodic loads [3]. However, over-estimation may occur when disaggregating those with weak periodicities. In [4], both the Fourier series transformed from the aggregate and weather information are featured in a RF classifier. However, additional sensors increase costs. In [5], various neural networks are applied to NILM, based on convolutional neural network (CNN), long short-term memory (LSTM) and stacked denoising autoencoder (DAE), respectively Benefiting from the wide range of trainable parameters, CNN and DAE outperform LSTM. However, their performances are sensitive to the usage frequency of each appliance. Recently, Zhang et al. proposed a sequence-to-point (seq2point) CNN framework in [8], generally outperforming sequence-to-sequence networks in load disaggregation task. To avoid sub-metering for the target houses, transfer learning across houses and datasets is studied in [9]. However, its performance relies on the similarity between the appliance usage patterns learnt from source dataset during training and those of the data collected from target houses. Moreover, sub-metering for the target houses is also required to fine-tune the networks.

Not like such model-based algorithms, GSP is an emerging signal processing tool, representing the stochastic properties of signals via graphs. In [10], a supervised GSP-based NILM method is proposed, where the graph total variation is minimized, referring to generally piece-wise smooth of the underlying graph signal. For performance refinement, simulated annealing is applied as post-processing to minimize the difference between the aggregate and the sum of disaggregated power in [11]. Note that sudden changes on active power signals are detected and featured in both [10] and [11]. To perform GSP-based NILM in an unsupervised manner, GSP concepts are utilized multiple times in adaptive threshold selection, signal clustering, and pattern matching in [6]. As in [10] and [11], the method proposed in [6] features power changes, limiting its robustness. Firstly, it is proposed only for disaggregating the loads with magnitude-wise close and matchable power changes when being switched ON and OFF. Secondly, transients lasting longer than the sampling period may be segmented into multiple consecutive power changes, resulting in mismatching. Although pre-processing proposed in [12] can sharpen state transition edges in power signals and gain performance improvement, transient characteristics are destroyed. Instead of widely-featured power change events, steady-state sequences are segmented from the aggregate and featured

NILM '22, November 9–10, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9890-9/22/11…$15.00
https://doi.org/10.1145/3563357.3566156

by an unsupervised GSP-based NILM method in [7]. Since the segmented sequences may differ in length, Dynamic Time Warping (DTW) is utilized to calculate similarity in graph edge weighting. However, its performance is sensitive to simultaneous load operation and measurement noises.

Driven by the aforementioned shortcomings of current works, an unsupervised GSP-based NILM approach is proposed in this paper, featuring power sequences from the aggregate. Different from the steady-state sequences segmented in [7], state transition sequence (STS) is defined in this paper, consisting of 'rising' and 'falling' power sequences. A 'rising' sequence refers to the starting power transient of an operational state, while its ending power transient can be represented by a 'falling' sequence. As in [7], DTW is utilized in graph edge weighting to calculate distances among extracted STSs for similarity quantification. The proposed method is benchmarked with two state-of-the-art unsupervised GSP-based NILM solutions proposed in [6] and [7]. All methods are validated on two publicly-available datasets, AMPds dataset from Canada and REFIT dataset from the UK. Our main contributions include:

- a state transition sequence extraction method is proposed for segmenting informative STS features;
- an unsupervised GSP-based NILM approach is proposed, where the similarity calculation for STSs is based on DTW;
- the proposed method is validated on open-access AMPds and REFIT datasets, benchmarked with state-of-the-art GSP-based NILM methods using different features.

The rest of this paper is organised as follows. The proposed method is explained in Section 2. In Section 3, we clarify the experimental setup and discuss the results. Section 4 contains the conclusion and future work.

2 METHODOLOGY

The proposed unsupervised GSP-based NILM method consists of three stages: STS extraction, GSP-based STS clustering and STS matching.

2.1 State Transition Sequence Extraction

Instead of the signal segmentation for capturing steady-state sequences in [7], a method for extracting STSs is proposed. STSs, storing complete power transients of beginnings and ends of operational states, are informative features in further clustering and matching. The proposed STS extraction algorithm is presented in Algorithm 1.

In Line 1, given an N-length aggregate power signal $\mathbf{p}$, t_0 and t_1 are heuristically set sequence length constraints, with a threshold T for mitigating small power changes not due to load state transition.After initialisation in Line 2, each power variation sample with $|\Delta p_i| \geq T$ and its previously neighbours are assigned to a sequence $\mathbf{e}$, based on the adaptive rules as in Lines 5–13. Otherwise, if a power variation sample with $|\Delta p_i| < T$ is near the stored sudden power change event in $\mathbf{e}$, i.e., their period is shorter than t_1, it will be merged into $\mathbf{e}$, as shown in Lines 14-17. In Line 18, standardisation is carried out on $\mathbf{e}$, by subtracting its minimum entry from each entry therein. Based on whether the ending entry in $\mathbf{e}$ is greater than its mean $\bar{\mathbf{e}}$, $\mathbf{e}$ can be labelled as a 'rising' or 'falling' STS. Thus, each 'rising' STS is grouped into a set $\mathcal{E}^{\mathrm{R}}$ while each 'falling' one is stored in $\mathcal{E}^{\mathrm{F}}$, with $\mathbf{e}$ reset to empty. After repeating such procedure for all power variation samples Δp_i, all 'rising' and 'falling' STSs are stored in $\mathcal{E}^{\mathrm{R}}$ and $\mathcal{E}^{\mathrm{F}}$, respectively.

2.2 GSP-based STS clustering

Then, clustering is independently carried out on the generated STS sets $\mathcal{E}^{\mathrm{R}}$ and $\mathcal{E}^{\mathrm{F}}$, based on separately built graphs. Therefore, taking $\mathcal{E}^{\mathrm{R}}$ as an example, the flow chart of its clustering is shown as in Fig. 1. Given a set $\mathcal{E}^{\mathrm{R}}$ containing N sequences, a graph can

Algorithm 1: STS extraction

1 **Input:** $\mathbf{p}$; N; T; t_0; t_1;
  Output: $\mathcal{E}^{\mathrm{R}}$; $\mathcal{E}^{\mathrm{F}}$;
2 **initialise** $\mathbf{e} \leftarrow [\,]$; $i \leftarrow 1$; $i_0 \leftarrow 0$; $i_1 \leftarrow -t_1$; $\mathcal{E}^{\mathrm{R}} \leftarrow \emptyset$; $\mathcal{E}^{\mathrm{F}} \leftarrow \emptyset$;
3 **while** $i \leq N-1$ **do**
4 $\Delta p_i \leftarrow p_{i+1} - p_i$;
5 **if** $|\Delta p_i| \geq T$ **then**
6 **if** $\mathbf{e} = [\,]$ **then**
7 **if** $i - i_0 \geq t_0$ **then**
8 **assign** $\mathbf{p}_{i-t_0+1:i-1}$ to $\mathbf{e}$;
9 **else if** $i > i_0 + 2$ **then**
10 **assign** $\mathbf{p}_{i_0+1:i-1}$ to $\mathbf{e}$;
11 **else if** $i = i_0 + 2$ **then**
12 **assign** p_{i_0+1} to $\mathbf{e}$;
13 **assign** p_i to $\mathbf{e}$; $i_0 \leftarrow i$; $i_1 \leftarrow i$;
14 **else if** $i - i_1 < t_1$ **then**
15 **assign** p_i to $\mathbf{e}$;
16 **else if** $i - i_1 = t_1$ **then**
17 **assign** p_i to $\mathbf{e}$;
18 $\mathbf{e} \leftarrow \mathbf{e} - \mathbf{e}_{\min}$;
19 **if** $\mathbf{e} \leq \bar{\mathbf{e}}$ **then**
20 $\mathcal{E}^{\mathrm{R}} \leftarrow \mathcal{E}^{\mathrm{R}} \cup \{\mathbf{e}\}$;
21 **else**
22 $\mathcal{E}^{\mathrm{F}} \leftarrow \mathcal{E}^{\mathrm{F}} \cup \{\mathbf{e}\}$;
23 $\mathbf{e} \leftarrow [\,]$;
24 $i++$;
25 **return** $\mathcal{E}^{\mathrm{R}}$; $\mathcal{E}^{\mathrm{F}}$

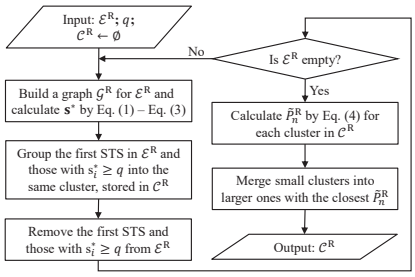


Figure 1: Flow chart of GSP-based STS clustering.

be defined as $\mathcal{G}^{\mathrm{R}} = (\mathcal{V}, \mathbf{A})$, where $\mathcal{V}$ is a set of graph nodes and $\mathbf{A}$ is the adjacency matrix. In $\mathcal{G}^{\mathrm{R}}$, each node v_i corresponds to a sample s_i in graph signal $\mathbf{s}$, as a mapping of sequence $\mathbf{e}^i$ in $\mathcal{E}^{\mathrm{R}}$, for $i = 1, ..., N$. Besides, each entry $A_{i,j}$ refers to the similarity or correlation between nodes v_i and v_j, usually weighted by a Gaussian kernel weighting function:

$$A_{i,j} = \exp\left\{-\frac{DTWdist(\mathbf{e}^i, \mathbf{e}^j)^2}{\rho^2}\right\}, \tag{1}$$

where ρ is a scaling factor, and $DTWdist(.)$ is the DTW distance between sequences as in [7]. In GSP-based clustering, $s_1 \leftarrow 1$ corresponds to the clustering target which can be randomly picked from the input signal [6]. Thus, we pick the first STS as the clustering target, while the remaining is to be grouped, with corresponding samples in $\mathbf{s}$ initialised as zeros.

Let $\mathbf{D}$ be a diagonal matrix with $D_{i,i} = \sum_j A_{i,j}$, the graph Laplacian matrix $\mathbf{L}$ can be defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$. Then the *global smoothness* of the graph signal can be defined as:

$$S_2(\mathbf{s}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} A_{i,j}(s_j - s_i)^2 = \mathbf{s}^T\mathbf{L}\mathbf{s}, \tag{2}$$

reflecting the intrinsic structure of the underlying graph. It is believed that if $\mathbf{s}^T\mathbf{L}\mathbf{s}$ is small, the graph signal $\mathbf{s}$ will be generally piece-wise smooth. The optimization problem $\arg\min_{\mathbf{s}} \|\mathbf{s}^T\mathbf{L}\mathbf{s}\|_2^2$ has a closed-form solution:

$$\mathbf{s}^* = \mathbf{L}^{\#}_{2:N,2:N}(-s_1)\mathbf{L}^T_{1,2:N}, \tag{3}$$

where $(.)^{\#}$ denotes the pseudo-inverse matrix.

Since each s_j^* corresponds to an STS to be clustered, a fixed threshold q is used to select candidates. Thus, we group the first STS (with $s_1 = 1$) and those with $s_j^* \geq q$ into the same cluster via storing them in a sub-set in C^{R}. After removing such clustered STSs from $\mathcal{E}^{\mathrm{R}}$, the above-mentioned process is repeated for the updated $\mathcal{E}^{\mathrm{R}}$ until no STS is left. Like in [6], small clusters are merged into larger ones for simplifying further cluster and matching. Assuming that the n-th cluster $c^{\mathrm{R},n}$ in C^{R} contains J STSs, thus for selecting mergeable clusters, we define the *magnitude* of $c^{\mathrm{R},n}$ as:

$$\tilde{P}_n^{\mathrm{R}} = \frac{1}{J}\sum_{j=1}^{J} |\mathbf{c}_{\mathrm{end}}^{\mathrm{R},n,j} - \mathbf{c}_1^{\mathrm{R},n,j}|, \tag{4}$$

where $\mathbf{c}_1^{\mathrm{R},n,j}$ and $\mathbf{c}_{\mathrm{end}}^{\mathrm{R},n,j}$ denote the first and last elements in the j-th sequence of $c^{\mathrm{R},n}$. Then, each small cluster in C^{R} with $J \leq J_0$ is merged to the larger one with the closest $\tilde{P}_n^{\mathrm{R}}$. Eventually, a set C^{R} is obtained as the clustering result of 'rising' STSs. Similarly, C^{F} can be obtained for 'falling' STSs.

2.3 Power STS matching

For matching the clusters of 'rising' STSs with those of 'falling' STSs via matching STSs, a cluster matching algorithm is proposed in Algorithm 2. In Algorithm 2, α and β are heuristically set trade-off factors of *magnitude* variation $\Delta\tilde{P}_m$ and duration $\Delta\tilde{T}_m$. After initialisation, *magnitudes* of all clusters in C^{R} and C^{F} are calculated as in Line 3. Then, STS matching starts for the $c^{\mathrm{R},n}$ with the largest $\tilde{P}_n^{\mathrm{R}}$, via selecting a candidate from C^{F} with the least d, as in Lines 4-16. In Lines 17-18, the paired 'rising' and 'falling' STSs are assigned in $\mathcal{S}_k^{\mathrm{R}}$ and $\mathcal{S}_k^{\mathrm{F}}$, separately, while the unpaired ones are stored in additional clusters $\mathcal{R}^{\mathrm{R}}$ and $\mathcal{R}^{\mathrm{F}}$ in Line 19. After performing similar STS matching steps on $\mathcal{R}^{\mathrm{R}}$ and $\mathcal{R}^{\mathrm{F}}$ until no remaining STS can be matched, two sets $\mathcal{S}^{\mathrm{R}}$ and $\mathcal{S}^{\mathrm{F}}$ are obtained.

Algorithm 2: Power STS matching

1 **Input:** α; β; C^{R} containing N clusters $c^{\mathrm{R},n}$; C^{F} containing M clusters $c^{\mathrm{F},m}$;
Output: $\mathcal{S}^{\mathrm{R}}$;$\mathcal{S}^{\mathrm{F}}$;
2 **initialise** $\mathcal{S}^{\mathrm{R}} \leftarrow \{\}$, $\mathcal{S}^{\mathrm{F}} \leftarrow \{\}$, $\mathcal{I} \leftarrow \{\}$, $\mathcal{R}^{\mathrm{R}} \leftarrow \{\}$, $\mathcal{R}^{\mathrm{F}} \leftarrow \{\}$; $k \leftarrow 1$;
3 **calculate** $\tilde{P}_n^{\mathrm{R}}\ \forall\ c^{\mathrm{R},n}$ **in** C^{R} and $\tilde{P}_m^{\mathrm{F}}\ \forall\ c^{\mathrm{F},m}$ **in** C^{F} by Eq. (4);
4 **for** $c^{\mathrm{R},n}$ **in** C^{R} with the largest $\tilde{P}_n^{\mathrm{R}}$ **do**
5 **foreach** $c^{\mathrm{F},m}$ **in** C^{F} **do**
6 $q \leftarrow 1$; $q_0 \leftarrow 0$; $d \leftarrow \infty$; $\Delta\tilde{\mathbf{t}}^n \leftarrow [\,]$; $\Delta\tilde{P}_m \leftarrow |\tilde{P}_n^{\mathrm{R}} - \tilde{P}_m^{\mathrm{F}}|$;
7 **foreach** $\mathbf{c}^{\mathrm{R},n,j}$ **in** $c^{\mathrm{R},n}$ **do**
8 $t^{\mathrm{R}} \leftarrow$ the time instance of the largest power increase **in** $\mathbf{c}^{\mathrm{R},n,j}$; $d^{\mathrm{t}} \leftarrow \infty$; $q \leftarrow q + q_0$; $q_0 \leftarrow 0$;
9 **foreach** $\mathbf{c}^{\mathrm{F},m,l}$ **in** $c^{\mathrm{F},m}$ **do**
10 $t_l^{\mathrm{F}} \leftarrow$ the time instance of the largest power decrease **in** $\mathbf{c}^{\mathrm{F},m,l}$;
11 **if** $t_l^{\mathrm{F}} > t^{\mathrm{R}}$ and $t_l^{\mathrm{F}} - t^{\mathrm{R}} < d^{\mathrm{t}}$ **then**
12 $\Delta\tilde{t}_q^n \leftarrow t_l^{\mathrm{F}} - t^{\mathrm{R}}$; $d^{\mathrm{t}} \leftarrow t_l^{\mathrm{F}} - t^{\mathrm{R}}$; $q_0 \leftarrow 1$;
13 **if** $\Delta\tilde{\mathbf{t}}^n \neq \emptyset$ **then**
14 $\Delta\tilde{T}_m \leftarrow$ the median value of $\Delta\tilde{\mathbf{t}}^n$;
15 **if** $\alpha * \Delta\tilde{P}_m + \beta * \Delta\tilde{T}_m < d$ **then**
16 $d \leftarrow \alpha * \Delta\tilde{P}_m + \beta * \Delta\tilde{T}_m$; $\mathcal{I} \leftarrow c^{\mathrm{F},m}$;
17 **find** all pairs of $\mathbf{c}^{\mathrm{R},n,j}$ **in** $c^{\mathrm{R},n}$ and $\mathbf{c}^{\mathrm{F},m,l}$ **in** $\mathcal{I}$, where the paired $\mathbf{c}^{\mathrm{F},m,l}$ and $\mathbf{c}^{\mathrm{R},n,j}$ are time-wise closest to each other and $\mathbf{c}^{\mathrm{F},m,l}$ lies after its pair $\mathbf{c}^{\mathrm{R},n,j}$;
18 **store** all paired 'rising' STSs into $\mathcal{S}_k^{\mathrm{R}}$ and 'falling' STSs into $\mathcal{S}_k^{\mathrm{F}}$, and **remove** them from $c^{\mathrm{R},n}$ and $c^{\mathrm{F},m}$;
19 $\mathcal{R}^{\mathrm{R}} \leftarrow \mathcal{R}^{\mathrm{R}} \cup c^{\mathrm{R},n}$; $\mathcal{R}^{\mathrm{F}} \leftarrow \mathcal{R}^{\mathrm{F}} \cup c^{\mathrm{F},m}$; $k{+}{+}$;
20 **repeat** STS matching for $\mathcal{R}^{\mathrm{R}}$ and $\mathcal{R}^{\mathrm{F}}$ as in Lines 17-19 **until** the remaining are unmatchable;
21 **return** $\mathcal{S}^{\mathrm{R}}$; $\mathcal{S}^{\mathrm{F}}$

3 EXPERIMENTAL SETUP & PERFORMANCE EVALUATION

For validation, experiments are carried out on open-access Canadian AMPds dataset[13] and British REFIT dataset[14], where data is collected from real houses and widely used in NILM works. For AMPds dataset sampled at 1min, four-month data from 2012-04-01 to 2012-07-31 is selected in the experiments.While, a longer period of data sampled at 6-8sec is picked for REFIT House 2, from 2013-09-18 to 2014-04-06. Two state-of-the-art unsupervised GSP-based NILM solutions are used as benchmarks and denoted by UGSP [6] and UGSP-DTW [7], featuring power changes and power steady-state sequences, correspondingly. Evaluation metrics include *True Positive* (*TP*), *False Positive* (*FP*), *False Negative* (*FN*), *Precision* (*PR*), *Recall* (*RE*) and $F_{measure}$ (F_m).

We first demonstrate the NILM results for AMPds dataset, as in Table 1. Ignoring the appliances which are rarely used or consume low power in the selected period, target appliances for disaggregation include Basement Plugs and Lights (BME), Clothes Dryer (CDE), Clothes Washer (CWE), Dishwasher (DWE), Kitchen Fridge (FGE), Heat Pump (HPE), Television (TVE) and Utility Room Plug (UTE). The overall F_m achieved by the proposed method is 0.82, superior to UGSP by 0.12 and UGSP-DTW by 0.05, respectively. For explanation, DWE is taken as an example. On one hand, the switching-ON transients of DWE usually last for 2-3min, which are longer than the sampling period as 1 min, leading to incompletely detected power changes and inferior state transition event matching result in UGSP. On the other hand, its steady-states are short and difficult to be captured in UGSP-DTW. However, the better F_m performance of the proposed method is benefited from extracting and matching STSs, which contain unique power transient characteristics of DWE. Note that the power consumed by FGE is around 110Watt, which is close to those for low-power operational states of other appliances, leading to high *FP* while low *PR* and F_m for UGSP. However, calculating STS similarity based on DTW distance and further STS matching in the proposed method contribute to F_m improvement by 0.12, confirming that information learnt from extracted STSs in the proposed method supports identifying FGE loads against others.

Then we move on to the NILM results for House 2 from REFIT dataset, as in Table 2. The main appliances in REFIT House 2 to be disaggregated include Fridge-freezer (FF), Washing Machine (WM), Dishwasher (DW), TV, Microwave (M), Toaster (T), and Kettle (K). From Table 2, both UGSP-DTW and the proposed method generally outperform UGSP, with around 0.1 improvement on F_m by featuring power sequences and DTW-based graph edge weighting. Note that

The proposed method									
App.	BME	CDE	CWE	DWE	FGE	HPE	TVE	UTE	Overall
TP	162	82	13	192	3702	75	6	70	4302
FP	56	5	11	18	420	12	9	23	554
FN	150	41	150	112	710	13	105	62	1343
PR	0.74	0.94	0.54	0.91	0.90	0.86	0.40	0.75	0.89
RE	0.52	0.67	0.08	0.63	0.84	0.85	0.05	0.53	0.76
F_m	0.61	0.78	0.14	0.75	0.87	0.86	0.10	0.62	0.82

UGSP [6]									
App.	BME	CDE	CWE	DWE	FGE	HPE	TVE	UTE	Overall
TP	110	98	25	152	2884	62	2	32	3365
FP	70	12	52	13	1298	6	21	19	1491
FN	183	56	195	103	642	21	83	53	1336
PR	0.61	0.89	0.32	0.92	0.69	0.91	0.09	0.63	0.69
RE	0.38	0.64	0.11	0.60	0.82	0.75	0.02	0.38	0.72
F_m	0.47	0.74	0.17	0.72	0.75	0.82	0.04	0.47	0.70

UGSP-DTW [7]									
App.	BME	CDE	CWE	DWE	FGE	HPE	TVE	UTE	Overall
TP	89	61	20	80	3351	71	5	35	3712
FP	49	5	16	5	953	14	31	12	1085
FN	122	39	130	92	623	13	86	46	1151
PR	0.64	0.92	0.56	0.94	0.78	0.84	0.14	0.74	0.77
RE	0.42	0.61	0.13	0.47	0.84	0.85	0.05	0.43	0.76
F_m	0.51	0.73	0.22	0.62	0.81	0.84	0.08	0.55	0.77

Table 1: Performance evaluation on AMPds dataset.

The proposed Method								
App.	FF	WM	DW	TV	M	T	K	Overall
TP	1071	17	130	52	136	44	270	1720
FP	372	12	37	92	20	12	30	575
FN	105	8	30	50	102	21	24	340
PR	0.74	0.63	0.78	0.36	0.87	0.79	0.90	0.75
RE	0.91	0.65	0.81	0.51	0.57	0.68	0.92	0.83
F_m	0.82	0.64	0.80	0.42	0.69	0.73	0.91	0.79

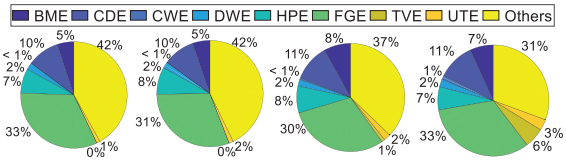
UGSP [6]								
App.	FF	WM	DW	TV	M	T	K	Overall
TP	962	16	96	19	92	29	233	1447
FP	551	5	97	89	84	11	41	848
FN	214	20	49	80	34	23	52	472
PR	0.64	0.76	0.50	0.18	0.52	0.73	0.85	0.60
RE	0.82	0.44	0.66	0.19	0.73	0.56	0.82	0.73
F_m	0.72	0.56	0.57	0.18	0.61	0.63	0.83	0.69

UGSP-DTW [7]								
App.	FF	WM	DW	TV	M	T	K	Overall
TP	1026	12	110	30	121	38	269	1606
FP	314	9	56	112	56	25	26	598
FN	123	7	33	64	36	22	35	320
PR	0.77	0.57	0.66	0.21	0.68	0.60	0.91	0.73
RE	0.89	0.63	0.77	0.32	0.77	0.63	0.89	0.83
F_m	0.82	0.60	0.71	0.25	0.72	0.62	0.90	0.78

Table 2: Performance evaluation for REFIT House 2.

the proposed method scores higher on TP than UGSP-DTW does, which is in line with the results on AMPds dataset. Since DW and WM operate at similar power ranges, around 2200 Watt and 2250 Watt, respectively, the task of disaggregating them is beyond the performance boundaries of UGSP [6]. However, DW and WM differ in operation duration, namely, the length of steady-state sequences featured in UGSP-DTW. Thus, UGSP-DTW outperforms UGSP for both DW and WM. Furthermore, DW takes a longer period to reach a stable power range since being switched ON than WM dose. Therefore, in the proposed method, 'rising' STSs of DW and WM can be grouped into various clusters and help refine disaggregation results for both appliances. It can be observed from both Table 1 and Table 2 that the disaggregation result for TV is the worst, due to frequent and significant power fluctuations prevent all methods from extracting qualified features for good clustering and matching performance. However, the proposed method is still slightly superior to the others. In all, the NILM results show that matching 'rising' and 'falling' STSs contributes to NILM performance improvement.

For demonstrating the disaggregated energy shares, pie charts are drawn for the NILM experiments on AMPds dataset as in Fig. 2. From Fig. 2, the proposed method performs better than benchmarks



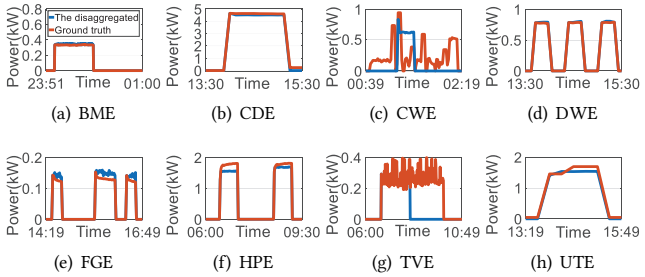
(a) UGSP (b) UGSP-DTW (c) The proposed method (d) Ground truth

Figure 2: Pie charts of disaggregated energy shares on AMPds dataset.

for most appliances, consistent with the results shown in Table 1. The underestimated power consumption shares of TVE for all methods also match their high FN results shown in Table 1.

The disaggregated load power signals are demonstrated in Fig. 3. From Fig. 3, the disaggregated power signals for most appliances are close to the ground truth except those for CWE and TVE, as in Table 1. Note that feature matching for CWE is a hard task, as it has multiple operational states in various power ranges. In terms of TVE, its disaggregation result is affected by power fluctuations while operating, which is afore-mentioned.

In conclusion, UGSP performs poor for the appliances operating at close power ranges or those with long-term power transients, as claimed in [6]. However, it requires the least storage and computation resources among the three methods. UGSP-DTW benefits from the steady-state sequence feature and outperforms UGSP, however,



(a) BME (b) CDE (c) CWE (d) DWE

(e) FGE (f) HPE (g) TVE (h) UTE

Figure 3: Typical disaggregation result of the proposed method on AMPds dataset.

its performance for short-lasting loads is limited. Although the proposed method requires more computational resources for STS clustering and matching, it succeeds in PR and F_m improvement for most appliances.

4 CONCLUSION AND FUTURE WORK

In this paper, an unsupervised GSP-based approach is proposed to perform load disaggregation on low-rate power measurements. For breaking the performance limitation of the existing unsupervised GSP-based NILM methods due to feature selection, STSs are extracted and featured in the proposed method for further clustering and matching. Experiments on open-access AMPds and REFIT datasets show the proposed method generally outperforms two state-of-the-art benchmarks, UGSP featuring power changes and UGSP-DTW featuring steady-state power sequences. For most appliances, the power consumption disaggregated by the proposed method is closer to the ground truth comparing to the benchmarks. Thus, it can be concluded that STS extraction and further matching in the proposed method help improve NILM performance. Future work includes validation across more datasets with various sampling rates and robustness enhancement for multi-state appliances via investigation on identifying all extracted STSs and steady-state power sequences belonging to the same operational cycle while matching.

ACKNOWLEDGMENTS

This work was supported by the Joint Funds of the National Natural Science Foundation of China (No. U2066207).

REFERENCES

[1] George William Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891, 1992.
[2] Xuhao Li, Bochao Zhao, Wenpeng Luan, and Bo Liu. A training-free non-intrusive load monitoring approach for high-frequency measurements based on graph

signal processing. In *2022 7th Asia Conference on Power and Electrical Engineering (ACPEE)*, pages 859–863, 2022.
[3] Stephen Makonin, Fred Popowich, Ivan V Bajić, Bob Gill, and Lyn Bartram. Exploiting hmm sparsity to perform online real-time nonintrusive load monitoring. *IEEE Transactions on smart grid*, 7(6):2575–2585, 2015.
[4] Ziwei Xiao, Wenjie Gang, Jiaqi Yuan, Ying Zhang, and Cheng Fan. Cooling load disaggregation using a nilm method based on random forest for smart buildings. *Sustainable Cities and Society*, 74:103202, 2021.
[5] Jack Kelly and William 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*, pages 55–64, 2015.
[6] Bochao Zhao, Lina Stankovic, and Vladimir Stankovic. On a training-less solution for non-intrusive appliance load monitoring using graph signal processing. *IEEE Access*, 4:1784–1799, 2016.
[7] Kanghang He, Vladimir Stankovic, and Lina Stankovic. Building a graph signal processing model using dynamic time warping for load disaggregation. *Sensors*, 20(22):6628, 2020.
[8] Chaoyun Zhang, Mingjun Zhong, Zongzuo Wang, Nigel Goddard, and Charles Sutton. Sequence-to-point learning with neural networks for non-intrusive load monitoring. In *Thirty-second AAAI conference on artificial intelligence*, 2018.
[9] Michele D'Incecco, Stefano Squartini, and Mingjun Zhong. Transfer learning for non-intrusive load monitoring. *IEEE Transactions on Smart Grid*, 11(2):1419–1429, 2019.
[10] Vladimir Stankovic, Jing Liao, and Lina Stankovic. A graph-based signal processing approach for low-rate energy disaggregation. In *2014 IEEE symposium on computational intelligence for engineering solutions (CIES)*, pages 81–87. IEEE, 2014.
[11] Kanghang He, Lina Stankovic, Jing Liao, and Vladimir Stankovic. Non-intrusive load disaggregation using graph signal processing. *IEEE Transactions on Smart Grid*, 9(3):1739–1747, 2018.
[12] Bochao Zhao, Kanghang He, Lina Stankovic, and Vladimir Stankovic. Improving event-based non-intrusive load monitoring using graph signal processing. *IEEE Access*, PP(99):1–1, 2018.
[13] Stephen Makonin, Bradley Ellert, Ivan V. Bajic, and Fred Popowich. Electricity, water, and natural gas consumption of a residential house in Canada from 2012 to 2014. *Scientific Data*, 3(160037):1–12, 2016.
[14] David Murray, Lina Stankovic, and Vladimir Stankovic. An electrical load measurements dataset of united kingdom households from a two-year longitudinal study. *Scientific Data*, 4:160122, 2017.

A PARAMETER SETTINGS

The parameter settings in the experiments for AMPds dataset and REFIT House 2 are shown in Table 3.

Comparing to the duration $\Delta\bar{T}_m$ for the appliances in AMPds dataset, those for REFIT House 2 are more distinguishable. Thus, a higher β is set for REFIT House 2 to facilitate feature representation.

Parameters	AMPds dataset	REFIT House 2
T	100	30
t_0	3	2
t_1	3	2
q	0.5	0.5
α	0.8	0.75
β	0.2	0.25
J_0	50	40
ρ	9.4	8.1

Table 3: Parameter settings for AMPds dataset and REFIT House 2.