

Non-intrusive Load Monitoring Solutions for Low- and
Very Low-Rate Granularity

Bochao Zhao

Centre for Intelligent Dynamic Communications
Department of Electronic and Electrical Engineering
University of Strathclyde, Glasgow

April 21, 2020

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for the examination which has led to the award of a degree.

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Abstract

Large-scale smart energy metering deployment worldwide and the integration of smart meters within the smart grid are enabling two-way communication between the consumer and energy network, thus ensuring an improved response to demand. Energy disaggregation or non-intrusive load monitoring (NILM), namely disaggregation of the total metered electricity consumption down to individual appliances using purely algorithmic tools, is gaining popularity as an added-value that makes the most of meter data.

In this thesis, the first contribution tackles low-rate NILM problem by proposing an approach based on graph signal processing (GSP) that does not require any training. Note that Low-rate NILM refers to NILM of active power measurements only, at rates from 1 second to 1 minute. Adaptive thresholding, signal clustering and pattern matching are implemented via GSP concepts and applied to the NILM problem. Then for further demonstration of GSP potential, GSP concepts are applied at both, physical signal level via graph-based filtering and data level, via effective semi-supervised GSP-based feature matching. The proposed GSP-based NILM-improving methods are generic and can be used to improve the results of various event-based NILM approaches.

NILM solutions for very low data rates (15-60 min) cannot leverage on low to high rates NILM approaches. Therefore, the third contribution of this thesis comprises three very low-rate load disaggregation solutions, based on supervised (i) K-nearest neighbours relying on features such as statistical measures of the energy signal, time usage profile of appliances and reactive power consumption (if available); unsupervised (ii) optimisation performing minimisation of error between aggregate and the sum of estimated individual loads, where energy consumed by always-on load is heuristically

Chapter 0. Abstract

estimated prior to further disaggregation and appliance models are built only by manufacturer information; and (iii) GSP as a variant of aforementioned GSP-based solution proposed for low-rate load disaggregation, with an additional graph of time-of-day information.

Contents

Abstract	ii
List of Akronyms	viii
List of Figures	xi
List of Tables	xiv
Acknowledgements	xviii
1 Introduction	2
1.1 Non-intrusive Load Monitoring Overview	2
1.2 Research Motivation	4
1.3 Research Contributions	6
1.4 Publication List	8
1.5 Thesis Overview	10
2 Background and Literature Review	12
Notations	12
2.1 Non-intrusive Load Monitoring	12
2.1.1 NILM Problem Formulation	13
2.1.2 Low-rate NILM Approach Overview	14
2.1.3 HMM vs. Deep Learning vs. GSP	20
2.2 Very Low-rate NILM Approach Overview	22
2.3 What constitutes noise in NILM and Denoising Solutions	27

Contents

2.4	Graph Signal Processing Preliminaries	30
2.5	Performance Evaluation Metrics for NILM	34
2.5.1	<i>F-measure</i>	34
2.5.2	Disaggregation Accuracy	35
2.5.3	Error in Estimating Total Power Consumption	36
2.5.4	Disaggregation Error Measure	36
2.5.5	Match Rate	36
2.5.6	Metrics for Very Low-rate NILM	37
3	Low-rate Power Disaggregation based on Graph Signal Processing	39
	Notations	39
3.1	Introduction	40
3.2	Training-less Graph Signal Processing based Clustering	41
3.3	Proposed Disaggregation Algorithm	42
3.3.1	Algorithm Overview	42
3.3.2	Edge Detection and Clustering	43
3.3.3	Feature Matching	47
3.3.4	Disaggregation Output	49
3.4	Performance Limits	50
3.4.1	Case 1	51
3.4.2	Case 2	51
3.4.3	Case 3	52
3.4.4	Case 4	53
3.4.5	Case 5	55
3.4.6	Summary	56
3.5	Case Study	57
3.6	Results and Discussion	62
3.6.1	Algorithm Performance	63
3.6.2	Comparison with State-of-the-Art	67
3.6.3	Performance Bound Analysis	69
3.7	Summary	70

4	Event-based Non-intrusive Load Monitoring Improvement using Graph Signal Processing	72
	Notations	72
4.1	Introduction	73
4.2	Proposed Signal Processing Algorithm for Enhancing NILM	74
4.2.1	Graph Filtering via Total Variation Regularization	75
4.2.2	Graph-based Bilateral Filtering	78
4.2.3	Edge Sharpening	79
4.2.4	Separating Similar Loads via NILM-result Refinement	80
4.3	Experimental Validation	83
4.3.1	Disaggregation Results using the Proposed NILM-result Refinement Approach	85
4.3.2	F_m and Acc^m Appliance-level Results	87
4.3.3	Further Insights with Additional Metrics	96
4.4	Summary	99
5	Very Low-rate Electricity Profile Disaggregation	101
5.1	Introduction	101
5.2	Hourly Energy Presentation Calculation	104
5.2.1	Down-sampling assuming Fixed Sampling Intervals	104
5.2.2	Down-sampling allowing Variable Sampling Intervals (Method 1)	104
5.2.3	Down-sampling allowing Variable Sampling Intervals (Method 2)	105
5.2.4	Methods Discussion	106
5.3	Proposed K-NN based Approach	107
5.3.1	K-NN Principles	108
5.3.2	Feature Extraction	108
5.3.3	Feature Matching	109
5.3.4	Experimental Validation and Discussion	112
5.4	Proposed Optimisation-based Approach (OPT)	115
5.4.1	Always-on Load Estimation and Removal	116
5.4.2	Appliance Modelling	118

Contents

5.4.3	Disaggregation via Optimisation	120
5.5	Graph Signal Processing - based Proposed Approach	120
5.6	Experimental Setup	123
5.6.1	Experimental Data	123
5.6.2	Benchmark Setup	124
5.6.3	Appliances Disaggregated	124
5.7	Experimental Results	124
5.7.1	OPT Parameters	124
5.7.2	GSP Parameters	126
5.7.3	CNN Parameters	126
5.7.4	Daily Consumption Accuracy and Match Rate Performance . . .	126
5.7.5	Analysis of Estimated vs. Actual Energy Consumption to explain Metrics	129
5.7.6	Experimental execution time	133
5.8	Summary	133
6	Conclusions and Future Work	135
6.1	Conclusions	135
6.2	Future Work	138
7	Appendix	141
7.1	Convolution Neural Network	141
7.2	Weekly and Monthly Acc^m Results for REFIT House 4 and 8 on Hourly Granularity	143
	Bibliography	145

List of Akronyms

ATP *AccurateTruePositive*

AUC Area Under Curve

B Bathroom GFI

BF Bilateral Filtering

BL Baseload

BR Baseload Removal

CNN Convolutional Neural Network

CO Combinatorial Optimization

CW Cloth Washer

DDSC Discriminative Disaggregation Sparse Coding

DEM Disaggregation Error Measure

DT Decision Tree

DTW Dynamic Time Warping

DW Dishwasher

EH Electrical Heater

EM Expectation-Maximization

Chapter 0. List of Akronyms

EWB Electronics Workbench

F Fridge

FFZ Fridge-freezer

FHMM Factorial Hidden Markov Model

F-HDP Factorial Hierarchical Dirichlet Process

F_m *F – measure*

FP *FalsePositive*

FN *FalseNegative*

FZ Freezer

GR Garage

GSP Graph Signal Processing

HMM Hidden Markov Model

HP Heat Pump

HSMM Hidden Semi-Markov Model

HWU Hot Water Unit

IAM Individual Appliance Monitoring

ITP *InaccurateTruePositive*

K Kettle

K-NN K-Nearest Neighbour

KO Kitchen Outlet

L Light

Chapter 0. List of Akronyms

M Microwave

MR Match Rate

MSE Mean Square Error

NILM/NIALM Non-Intrusive Load Monitoring

NM Noise Measure

O Oven

OPT proposed Optimisation-based NILM approach

PDF Probability Density Function

PR Precision

RE Recall

RSD RelativeStandardDeviation

S Stove

SBNMF Semi-Binary Non-negative Matrix Factorization

SGSP Supervised Graph Signal Processing

SNE Security/Network Equipment

SVM Support Vector Machine

T Toaster

TECA Total Energy Correctly Assigned

TER Total power consumption estimation Error Rate

TP TruePositive

UGSP Unsupervised Graph Signal Processing

Chapter 0. List of Akronyms

UT Utility room

WD Washer Dryer

Wh Watt-hour

WM Washing Machine

List of Figures

2.1	Example of typical features extracted from the aggregate power readings for House 1 of the REDD dataset on 30/04/2011.	15
2.2	Example of a state transition model (finite state machine) and a hidden Markov model.	21
2.3	Typical “disaggregation” noise in electrical loads measurements, observed in House 17 of the REFIT dataset.	28
3.1	Flow chart of the proposed unsupervised GSP-based NILM algorithm	44
3.2	F_m versus $\Delta\omega$ for different ω_1 and fixed σ_1 's	52
3.3	F_m versus σ 's where $\Delta\omega = 200$	53
3.4	F_m versus $\Delta\omega = \omega_2 - \omega_1$ and $\frac{\sigma_2}{\sigma_1}$	54
3.5	F_m versus $\Delta\omega$ for different ω_1	55
3.6	The limits of the proposed GSP-based NILM algorithm	55
3.7	Graph generation for Stage 1 of clustering.	57
3.8	Clustering results.	58
3.9	A typical positive-negative cluster pair.	60
3.10	Graph example for feature matching.	60
3.11	Disaggregation result for three typical days.	61
3.12	Pie charts of NILM results for House 2 from REDD dataset for three typical days.	62
3.13	Example of aggregate readings for REFIT House 9 on 25th Feb. 2015	66
3.14	Pie charts of NILM results for House 1 from REDD datasets.	68
3.15	REFIT House 8 kettle and TV disaggregation.	69

List of Figures

4.1	Flowchart of the proposed algorithm.	76
4.2	Example of graph filtering via total variation regularization, performed on window Ω of power measurements.	78
4.3	Histogram of $\Delta_i = \Delta P_i - \sum_{m \in \mathcal{M}} \Delta P_i^m $ for House 17 of REFIT dataset.	79
4.4	An example of vector $l_{i,j}^m$ for House 1 from the REDD dataset.	81
4.5	Graph generation	82
4.6	Performance of the proposed method with benchmarks for REDD House 1	87
4.7	Performance of the proposed method with benchmarks for REDD House 2	88
4.8	Performance of the proposed method with benchmarks UGSP, SGSP for fridge in each house from REDD dataset.	90
4.9	Performance of the proposed method with benchmarks UGSP, SGSP for the four most common appliances (i.e., B, DW, F, WD) in each house from REDD dataset.	91
4.10	Performance of the proposed method (P) with benchmarks UGSP, UGSP with BR, SGSP, SGSP with BR, DT, DT with BR and FHMM for REFIT House 2.	92
4.11	Typical appliance operation in the aggregate power consumption data before (shown in blue) and after processing via the proposed method (shown in red).	93
4.12	Performance of the proposed method (P) with benchmarks UGSP, UGSP with BR, SGSP, SGSP with BR, DT, DT with BR, CO and FHMM for REFIT House 17.	95
4.13	Disaggregation results, presented as percentage load contribution per appliance relative to the aggregate load, for House 1 from REDD datasets.	97
5.1	Hourly profile generation example for Method 1.	105
5.2	Hourly profile generation example for Method 2.	106
5.3	Hourly profile of House 4 from REFIT dataset calculated for a typical week.	107
5.4	Flow chart of the proposed algorithm.	111
5.5	\mathbf{E}_Q versus \mathbf{E}_P for the same time.	114

List of Figures

5.6	Always-on appliance disaggregation for REFIT House 4.	130
5.7	Always-on appliance disaggregation for REFIT House 8.	130
5.8	Disaggregation performance for other appliances in REFIT House 4. . .	132
5.9	Disaggregation performance for other appliances in REFIT House 8. . .	132
7.1	Proposed CNN architecture for very low-rate disaggregation. Both windows of 7-hour long aggregate data and the 4-th 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 flatten operation that reshapes a matrix into a vector array with x values. 'Batch Norm' is a batch normalisation layer. 'LeakyReLU' refers to a leaky version of a rectified linear unit activation function, with a fixed slope coefficient=0.3. 'Dropout x' means that the dropout layer randomly sets a fraction rate=x of input units to zero at each update during the training phase.	142

List of Tables

2.1	A summary of appliances disaggregated in literature at low to very low sampling rates. Each appliance is linked to the corresponding datasets, including REDD, REFIT, GREEND, AMPds, UK-DALE, Dataport, REMODECE and others not publicly available (from manufacturer, supplier, self-monitoring, etc.), denoted using “+”.	26
3.1	Stage 1 clustering results for House 2 from REFIT dataset.	47
3.2	Performance of the proposed approach and the number of clusters for different initial ρ	54
3.3	Performance of the proposed approach for House 1 from the REDD dataset.	64
3.4	Performance of the proposed approach for House 2 from the REDD dataset.	64
3.5	Performance of the proposed approach for House 6 from the REDD dataset.	65
3.6	Performance of the proposed approach for House 8 from the REFIT dataset.	65
3.7	Performance of the proposed approach for House 9 from the REFIT dataset.	66
3.8	Performance of three NILM approaches for Houses 2 and 6 from the REDD dataset.	67
3.9	Performance comparison with and for House 2 from the REDD dataset.	68
4.1	Parameter setting used in the experiments.	84

List of Tables

4.2	Performance of the proposed NILM-result refinement for multiple appliances. N refers to disaggregation with the proposed methods before NILM-result refinement in Fig. 4.1. P refers to disaggregation with all steps of Fig. 4.1, including NILM-result refinement (discussed in Subsection 4.2.4).	86
4.3	Normalised total power consumption estimation error rate (TER) per appliance for REFIT House 2.	96
4.4	Normalised total power consumption estimation error (TER) of UGSP and P-UGSP for REDD House 1.	96
4.5	$F - measure$ performance for four selected houses.	98
5.1	Analytical comparison of multiple hourly profile calculation methods on House 4 from REFIT dataset	106
5.2	Features considered for appliances from the AMPds dataset.	110
5.3	Performance of the proposed method for 15-minute data.	113
5.4	Performance of the proposed method for hourly data.	113
5.5	Performance of the proposed method for the AMPds dataset for hourly data.	114
5.6	Experimental data selection for training and testing	123
5.7	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.	125
5.8	Appliance models for REFIT House 4.	125
5.9	Appliance models for REFIT House 8.	125
5.10	Daily Acc^m results for REFIT House 4.	127
5.11	Daily Acc^m results for REFIT House 8.	127
5.12	Disaggregation daily MR^m results (%) for REFIT House 4.	128
5.13	Disaggregation daily MR^m results (%) for REFIT House 8.	129
5.14	Execution time for all methods validated on both houses	133

List of Tables

7.1	Weekly Acc^m results of the proposed methods with benchmarks for RE-FIT House 4.	144
7.2	Monthly Acc^m results of the proposed methods with benchmarks for REFIT House 4.	144
7.3	Weekly Acc^m results of the proposed methods with benchmarks for RE-FIT House 8.	144
7.4	Monthly Acc^m results of the proposed methods with benchmarks for REFIT House 8.	144

Acknowledgements

I would like to first thank Dr Lina Stankovic and Dr Vladimir Stankovic for their endless guidance and supervision for years on both academics and behaviour, which makes me a mature PhD. They opened a gate for me, where inside is an amazing and novel world made up of energy management, data analysis and machine learning. I would also like to thank all my colleagues in Strathclyde, reviewers and audiences for their thoughtful advice and comments.

I would like to thank Prof. Zixiang Xiong and Dr Sameul Cheng with research teams for their help during my visits in Texas A&M, College Station, US (Oct.-Dec. 2015) and Tongji University, Shanghai, China (July 2017 and July 2018).

The author was supported by the University of Strathclyde Faculty of Engineering Scholarship (2014-2017), and University of Strathclyde Department of Electronic and Electrical Engineering Scholarship (2014-2017). The work presented in the first two research chapters was supported by the U.K. Engineering and Physical Sciences Research Council (EPSRC) Projects under the Transforming Energy Demand in Buildings through Digital Innovation (BuildTEDDI) Funding Programme under Grant REFIT EP/K002368. The work in the last chapter was also supported in part by the European Union Horizon 2020 research and innovation programme, Eco-Bot project under grant agreement No. 767625.

Finally from the bottom of my heart, I would like to thank my parents and Jiawen He, for years of support and encouragement.

Chapter 0. Acknowledgements

Chapter 1

Introduction

1.1 Non-intrusive Load Monitoring Overview

The total amount of smart energy meters installed worldwide will reach 780 million by 2020 [1], including 200 million in the EU [2]. Such large-scale deployment of smart metering devices has ignited a renewed interest in data analytical research to maximize benefits from the collected energy data, via real-time energy feedback, novel energy-saving services and more flexible pricing mechanisms that are useful to householders and other stakeholders. It leads to many algorithmic improvements and some commercial products aimed to enrich energy feedback [3]. Real-time energy feedback is currently available with wide-scale smart meter deployments via In-Home Displays, where the user interface in the form of a stand-alone display screen is used for presenting up-to-date energy-consumption, in kWh and pounds [4], or on the cloud. It can be made more informative and actionable by an exciting application that goes beyond remote billing, non-intrusive load monitoring. Non-intrusive load monitoring was proposed by G. W. Hart in the 1980's [5], usually abbreviated to NILM or NIALM, also known as load disaggregation.

NILM refers to estimating individual appliance energy consumption from the aggregate electricity measurements for billing purpose purely using software tools, that is, without sub-metering or additional hardware. NILM has the potential to provide low-cost, efficient and fine-grained energy feedback that can potentially reduce domestic

Introduction

electricity consumption by 0.7%-4.5% compared to pure aggregate consumption feedback [6]. It is claimed in [3] that NILM can deepen energy feedback leading to more efficient use of appliances (up to 20% of reduction in energy consumption is expected via appliance-feedback and specific appliance upgrade programs). Furthermore, the applications of NILM go beyond supporting energy-efficient behaviour [7, 8], as NILM has already been shown to support national surveys on the energy intensity of domestic activities, by mapping activities to appliance use [9]. Besides, NILM results can be used for quantifying and visualising appliance usage and further scalable appliance modelling [10]. Similarly, NILM supports the accurate estimation of the residential consumption phase of food life-cycle assessment, through data-driven appliance energy consumption modelling using disaggregated data [11]. NILM also benefits house maintenance and appliance retrofit [12], appliance-level anomaly detection [13], home automation and residents' activity recognition [14]. Such benefits rely on utilising the disaggregation results to assess domestic appliances and then offering feedback to the householder. For example, once an ageing appliance is detected during NILM, a suggestion of replacing it by a new energy-saving model might be given to the user if the return of investment is suitable [12]. In addition, appliance-level disaggregation results provide information on appliance usage patterns and can be mapped to domestic activities [14], such as cooking, entertainment, etc. Potential benefits of NILM also include energy suppliers can better forecast demand, system operators can monitor the effect of smart grid fluctuations on the residential microgrid and appliance manufacturers can optimise product design to meet customer usage habits [3].

Driven by the host of emerging applications and huge benefits, NILM has become a very active research area [15–17]. Numerous advanced signal processing and machine learning methods have been investigated to solve the NILM problem. Such NILM solutions include: 1) machine learning: stochastic finite state machines (Hidden Markov Model and its variants) [8, 18–25], support vector machines (SVM) [26], decision tree (DT) [26, 27], dynamic time warping (DTW) [27], k-nearest neighbours (K-NN) [26, 28–31], sparse coding [32, 33], motif mining [34], artificial neural networks, shallow [35, 36] and deep neural networks [7, 37–39]; 2) signal processing: graph signal processing

(GSP) [40, 41] and 3) mathematical methods: optimisation via learning of appliance models and occupancy information [42–46]. Besides, advanced hybrid NILM approaches are studied for improving NILM performance, e.g., k-means clustering based training followed by disaggregation using SVM [47], GSP with result refinement using simulated annealing [16], deep neural network utilised to learn deeper and multiple layers of sparse signal representation [48]. The open questions of the current NILM methods include: i) no set of features has been defined, nor a general NILM approach been designed that perform equally well for all types of appliances; ii) to date, there has been no formal evaluation of an algorithm’s trade-off between complexity and disaggregation accuracy; iii) the NILM approaches based on statistical models usually limit the number of appliances to be disaggregated, as their complexity is usually exponential.

1.2 Research Motivation

In this section, three NILM-related problems that have not been solved properly are introduced and the motivation for the contributions of this thesis, are discussed.

- **Algorithm design weaknesses:** Despite significant NILM research in recent years, low-rate NILM defined as NILM of electricity readings captured with sampling rate from 1 second to 1 minute via widespread smart meters [49], is still an open problem. Current low-rate NILM methods are reviewed in Section 2.1. The Weaknesses of such approaches include: requirement of a large amount of data for training, the requirement of extra information other than pure measurements collected for metering and billing purposes, validation on data with few noises but not real-world noisy scenarios and disaggregation only for top consumed appliances, etc. Hence, a low-complexity, low-rate NILM method is proposed that operate only on the measurements used for billing purposes.
- **Susceptibility to measurement noise:** Besides, current state-of-the-art NILM solutions are susceptible to measurement noise and outliers when dealing with real-world data and do not demonstrate sufficient accuracy [15,17]. One reason for this is the complex nature of the NILM problem with effective solutions requiring

both core physical-level signal processing - to process acquired signals reducing jitter, noise, spurious events [16, 50] - and machine learning-based clustering and classification [15]. GSP-based approaches have recently been proposed for tackling the NILM problem, via supervised [16] and unsupervised approaches proposed in Chapter 3. However, this prior work applied GSP at the data processing stage only, i.e., as a robust classification or clustering tool, without exploiting GSP's properties as effective physical signal filters [51], which can combat NILM sensitivity to measurement noise and the influence of unknown appliances. It is well recognized [15, 16, 50], that without appropriate processing of the physical measured signal, NILM will often not be accurate or successful, regardless of the effectiveness of the employed classification method.

- **Challenges of very low-rate NILM:** In respect of very low-rate energy disaggregation (usually with granularity above 10 minutes), prior work in this area is limited. Majority of NILM approaches are developed and validated on power measurements with sampling rates of 1 minute or higher, and cannot be used with very low-rate smart meter data. However, mainly due to storage, data management, and privacy constraints [52], the resolution of load measurements available from roll-out smart meters is much lower, e.g., 15 minutes in Italy [53], 1 hour in Spain [54], 15 minutes or 1 hour in the US [55], 30 minutes in the UK [49], 1 hour in British Columbia and Ontario, Canada [52]. Compared with power measurements of higher granularity, the energy consumption signal at very low granularity features limited state transitions, fewer low-consuming appliances' feature patterns and a much higher probability of multiple appliances running simultaneously. Thus, lack of well-known features and increased appliance noise make very low-rate NILM a challenging problem [28]. 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 lim-

ited to aggregated known-appliance load profiles instead of actual smart meter readings containing many unknown appliances.

1.3 Research Contributions

The contributions of this thesis are clarified in this section, where approaches are proposed for addressing the three concerns described in Section 1.2.

First, a *blind* approach is proposed in Chapter 3 for low-rate electricity measurements with the granularity from 1 second to 1 minute. The proposed approach disaggregates any aggregate active power dataset requiring neither training set nor prior knowledge, including knowledge of appliances contributing to the aggregate or their number. It relies on GSP [51], an emerging field based on representing a dataset using a discrete signal indexed by nodes of a graph. Opposite to the supervised approach of [40,56] where GSP is employed for data classification only, GSP is used three times in the proposed disaggregation approach: first for robust event detection, then to perform clustering, and finally for feature matching.

In contrast to traditional machine-learning approaches, such as HMM, that require plenty of observations to construct a graph, the proposed GSP approach takes an intuitive approach in constructing a graph without relying on the signal’s statistics [57]. Thus, it is expected that the proposed approach will work well in the absence of a training dataset, unlike traditional HMM-based and other machine learning methods [8, 21, 22, 58, 59]. The proposed approach is event-based and relies only on time-series data without any training, where good accuracy is demonstrated using two open-access datasets: REDD [60] (US houses) and ‘noisier’ REFIT [61] (UK houses). Comparing with the REDD dataset, the REFIT dataset suffers from a higher-level of unlabelled appliance noise [19].

To address the second concern, in Chapter 4 GSP is proposed as a tool that brings together low-level *signal* processing and application-driven *data* processing to improve the performance of various event-based NILM approaches, suitable for diverse electrical load datasets. Two *universal* algorithms are proposed to enhance NILM:

- **Graph-based NILM pre-processing:** Capitalizing on recent advances in GSP filtering (see [51, 62, 63]), a novel signal processing approach is proposed to mitigate sensor noise and sharpen signal edges to improve detection of on/off appliance events, which in turn facilitates more effective feature extraction and classification in NILM. Two types of GSP filters are designed - one based on total variation regularization [62] and the other based on bilateral filtering [64]. Since graph bilateral filtering results in a smoother output at the cost of occasionally filtering out true events, An algorithm is developed to select, automatically, the best filtering method.
- **NILM-result refinement:** Relying on robust GSP-based *data* classification [16, 51, 63], a novel NILM result *refining* method is proposed, applicable to any NILM algorithm; this method is based on semi-supervised GSP-based feature matching to improve disaggregation results by removing confusion between appliances with similar power levels that are often misclassified by the initial NILM classification engine.

The methods are proposed as generic external tools for improving the disaggregation performance of a range of NILM approaches, including supervised, semi-supervised and unsupervised NILM. The effectiveness of the proposed methods are demonstrated across three state-of-the-art NILM approaches, based on DT [27], supervised GSP [16] and unsupervised GSP NILM proposed in Chapter 3. Besides the methods from [16, 27], the performance is also benchmarked against two additional NILM methods from the publicly available NILMTK toolbox based on Factorial Hidden Markov Model (FHMM) and Combinatorial Optimization (CO) [5, 18]. Similarly, results are validated using two datasets of true power measurements: REDD [60] and REFIT [61].

With respect to very low-rate load disaggregation, solutions are proposed in both supervised and unsupervised approaches, which differ in requirement of training. First, K-NN is utilised as a supervised solution of electricity usage profile disaggregation of energy measurements at 15 and 60 min granularity to identify a range of appliances. Relative standard deviation is proposed as a metric to determine which features are most useful for disaggregating particular appliances. Unlike [28, 31, 65], the disaggrega-

tion results are validated using three open-access datasets of true power measurements: REDD [60], REFIT [61] and AMPds [66] (a Canadian house).

Then a training-less optimisation-based (OPT) NILM approach is proposed that aims to estimate appliance-level consumption. The algorithm starts with estimating and removing the baseload, then finding the combination of appliance models generated from manufacture information, that minimises a cost function. OPT utilise only Appliance manufacturer information in addition to smart meter measurement. A GSP-based NILM approach, adapted from the higher resolution GSP-based approach proposed in Chapter 3 is also utilised to disaggregate hourly electricity profile data, where an additional graph for time-of-day features is added. NILM approaches used for benchmarking include a convolutional neural network (CNN) (described in 7.1), FHMM and CO implemented in NILMTK [18] and Discriminative Disaggregation Sparse Coding (DDSC) proposed in [32]. All disaggregation approaches are validated on the REFIT dataset [61], known to be 'noisier' than other publically available datasets due to many unknown appliances.

1.4 Publication List

In this section, all published or submitted works related to this thesis by far are listed.

1. B. Zhao, L. Stankovic, and V. Stankovic, "Blind non-intrusive appliance load monitoring using graph-based signal processing," in *Proc. 3rd IEEE Global Conf. Signal Info. Process. (GlobalSIP 2015)*, Orlando, FL, USA, Dec. 2015, pp. 68–72.

I did design, programming and simulation for the proposed GSP-based algorithm for low-rate NILM in Matlab and I wrote the paper draft.

2. 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, Apr. 2016.

I did design, programming and simulation for the proposed GSP-based algorithm for low-rate NILM in Matlab and I wrote the paper draft.

Introduction

3. 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*, Mar. 2018.

I did design, programming and simulation for the proposed K-NN algorithm for very low-rate NILM in Matlab and I wrote the paper draft.

4. 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. 53944–53959, Sep. 2018.

I did design, programming for the proposed pre- and post-processing algorithms in Matlab and I did the simulation of applying proposed algorithms on unsupervised GSP-based NILM method. I also wrote the paper draft.

5. K. He, B. Zhao, L. Stankovic, and V. Stankovic, "A generic optimisation-based approach for improving non-intrusive load monitoring," *IEEE trans. Smart Grid (Early Access)*, Mar. 2019.

I did simulation and analysis of results for a benchmarking post-processing algorithm based on GSP in Matlab.

6. B. Zhao, M. Ye, L. Stankovic, and V. Stankovic, "Non-intrusive load disaggregation solutions for very low-rate smart meter data," *Applied Energy*, Apr. 2020 (In press).

I did design, programming and simulation for two proposed algorithm for hourly NILM based on optimisation and graph signal processing and one benchmark using sparse coding. I also wrote the paper draft except the content related to Convolutional Neural Network proposed by a co-author.

Then the correlation between publication and chapters is clarified. Chapter 3 contributions on a training-less low-rate NILM approach are published in papers 1 and 2. Chapter 4 contributions on pre-processing for low-rate NILM and NILM result refining methods are published in papers 4 and applied as a benchmark in paper 5. Chapter 5

contributions on three very low-rate NILM approaches are published in papers 3 and 6.

1.5 Thesis Overview

The rest of the thesis chapters have the following layout.

Chapter 2 – provides a detailed background review of several topics: low-rate (1-60seconds) and very low-rate (15-60minutes) NILM solutions; noise types in meter readings which affect NILM performance; methods proposed for improving NILM performance; preliminaries of GSP and evaluation metrics for NILM, used in the entire thesis.

Chapter 3 – describes the unsupervised GSP approach proposed for low-rate (1-60seconds) NILM problem. Five cases of disaggregation on two appliance loads are studies via control variates as a deeper algorithm capability investigation. The algorithm performances, compared with the state-of-the-art approach based on unsupervised HMM, are demonstrated for two public datasets.

Chapter 4 – this chapter explains in detail how the graph filtering is applied to NILM and combined with bilateral filter. How semi-supervised GSP is utilised to refine NILM results by separating similar loads is also clarified. Proposed methods are tested in conjunction with unsupervised and supervised GSP and DT, against a baseload removal pre-processing method. FHMM and CO with median filters are exploited as benchmarks. The results, via multiple metrics, are analysed and discussed with the evaluation of measurement noise level in each house.

Chapter 5 – investigates three solutions of very low-rate NILM (15-60minutes). First, by proposing a supervised K-NN with features including appliance time usage profile and using relative standard deviation for quality evaluation of each feature and wise feature selection of each appliance. Then validation of such K-NN approach on three datasets is shown, for both 15-min and hourly electricity profile entries. Next, an unsupervised OPT is proposed for disaggregation on hourly consumption readings where only manufacturer information is required to build appliance models. Besides, an unsupervised GSP approach adapted from the GSP-based approach proposed for

Introduction

low-rate NILM as in Chapter 3 is applied to this problem by adding an extra graph for time information. Finally, both of the proposed unsupervised approaches are comprehensively benchmarked with state-of-the-art DDSC, CO, FHMM and CNN approaches.

Chapter 6 – summarises the findings and discusses potential future work.

Chapter 2

Background and Literature Review

Notations

- A** $N \times N$ adjacency matrix
- D** $N \times N$ degree matrix
- L** $N \times N$ degree matrix
- P^m** Sub-metering active power signal of Appliance m
- P** Aggregate active power signal
- n** noise signal in smart meter readings
- s** $N \times 1$ graph signal
- G** An undirected graph containing N vertices
- M** the set of known appliances in the house
- V** A set of N vertices
- ρ Scaling factor for Gaussian kernel weighting

2.1 Non-intrusive Load Monitoring

NILM refers to analytical methods that take as an input electrical parameters (voltage, current, active/reactive power etc.) measured at the household's mains meter and output energy consumption, broken down to appliance level. Compared with intrusive

load monitoring, namely monitoring the power consumption of individual appliances via smart plugs, NILM is cheaper and user-friendly. As reported in the systematic review of [6], multiple studies support the effectiveness of disaggregated energy feedback and personalized energy-saving recommendations lead to a reduction in electricity use, as 0.7%-4.5% on average. Additionally, disaggregated power consumption results can support national surveys on energy intensity of domestic activities [9], scalable appliance modelling [10], accurate estimation of the residential consumption phase of food life-cycle assessments [11] and house maintenance and retrofit [12]. The disaggregation results also have the potential to produce useful appliance usage/operation information. For example, the accurate estimation of appliance usage frequency and time-of-day information through NILM supports offering advice for home automation and residents' activity recognition [14].

Usually, an energy provider would be running the NILM algorithms from its backend (on the cloud) that enables bidirectional communications to the customer for acquiring meter data and sending NILM results. However, the low-complexity unsupervised NILM approaches requiring neither training nor extra information could be embedded on chips in the smart metering equipment. Therefore, NILM can be partly done within user-end and the unlabelled disaggregation results are sent to the cloud for further labelling. Generally, the low complexity and short execution time of energy disaggregation solutions would result in a low implementation investment for NILM solution providers and their customers. Therefore, the benefits of a well-proposed NILM solution outweigh the investments costs in the medium to long term. The rest of this section reviews the background of energy disaggregation, by first formulating the disaggregation problem and then presenting an overview of existing NILM approaches.

2.1.1 NILM Problem Formulation

Let P_{t_i} be the total household's active power measurement at time instance t_i , for $i = 1, \dots, n$, which can be presented as a sum of the power contribution of individual

appliances and different types of noise:

$$P_{t_i} = \sum_{m \in \mathcal{M}} P_{t_i}^m + n_{t_i}, \quad (2.1)$$

where $P_{t_i}^m$ refers to the power consumed by each individual appliance m in the set of known appliances in the house denoted by \mathcal{M} . The noise term n_{t_i} includes random measurement noise, baseload, plus all unlabelled appliances in the house. By using time index i to represent time instance t_i , Eq. (2.1) can be simplified as:

$$P_i = \sum_{m \in \mathcal{M}} P_i^m + n_i, \quad (2.2)$$

The NILM task is to find the power contribution of each individual appliance m , P_i^m for each i . Therefore, the problem can be formulated into an optimisation problem:

$$\min_{P_i^m} \left| P_i - \sum_{m \in \mathcal{M}} P_i^m \right|. \quad (2.3)$$

Next, an overview of approaches taken to tackle the problem of finding P_i^m in the presence of noise is provided.

2.1.2 Low-rate NILM Approach Overview

In this subsection, an overview of existing low-rate NILM approaches is presented that work on active power measurements only, at rates from 1sec to 1min, which resembles the type of data, available using smart energy meters deployed on large scale worldwide [49]. Such approaches can be classified by their characteristics (e.g., event-based and state-based, supervised and unsupervised, etc.).

NILM approaches can be event-based or state-based. For event-based NILM, windows of *events* are first identified, where an event is defined as a sequence of power measurements. Each event starts with a rising edge due to an appliance being switched on, or a multi-state appliance transiting to a higher power state, and ends with a falling edge when an appliance is switched off, or a multi-state appliance returning to a lower-power state [67]. After such events are identified (usually via edge detection),

features are extracted from each such event window, where some typical feature types are illustrated in Fig. 2.1. The detected rising/falling power edge magnitude features are shown in Fig. 2.1b. In Fig. 2.1c, the area features are extracted from the smart meter readings. The time duration features can be observed in Fig. 2.1d.

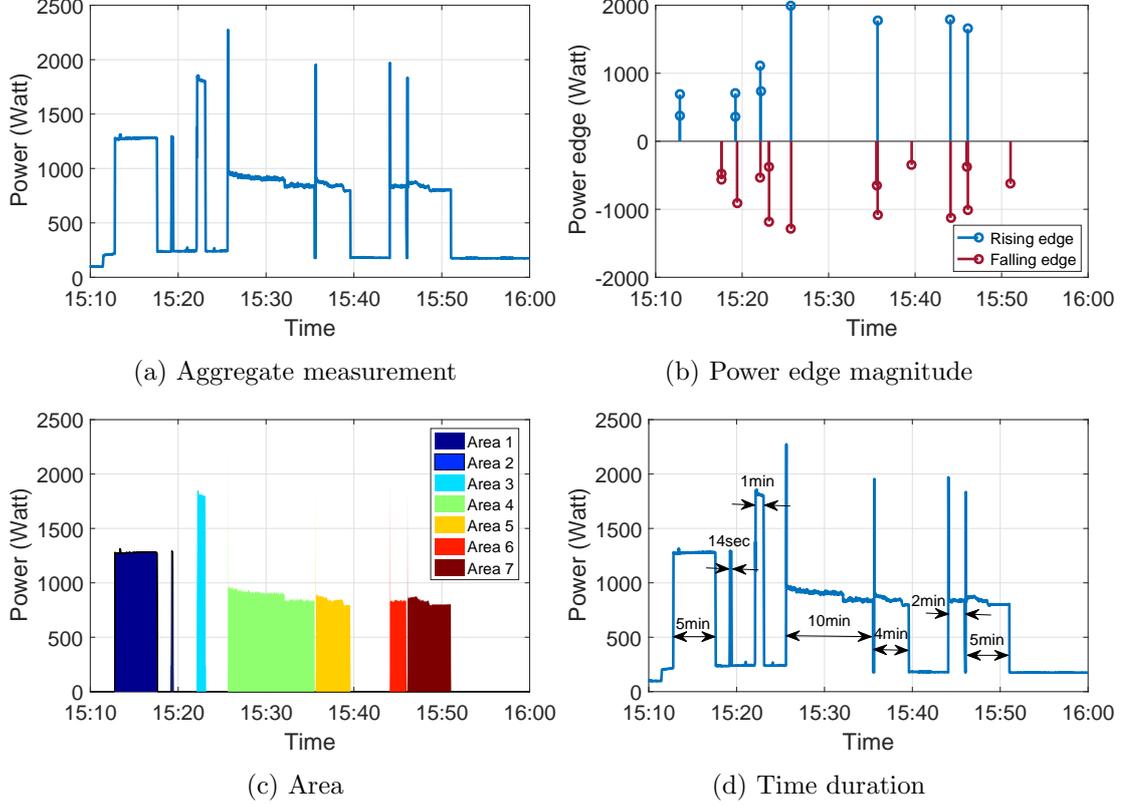


Figure 2.1: Example of typical features extracted from the aggregate power readings for House 1 of the REDD dataset on 30/04/2011.

Note that the features extracted from current/voltage phases are usually proposed for measurements sampled at high rates, in the order of kHz and MHz, like odd harmonic current vectors utilised in [68] and [69], which are not available for low- and very low-rate NILM.

Finally, classification is performed on the extracted features using a model built during the training process through SVM, DT, neural network, etc. [7, 27, 47]. Instead of classification, clustering with labelling via matching grouped features with a database can also be utilised to identify events [70]. Examples of event-based NILM include:

a hybrid K-means clustering of the training data followed by SVM-based energy disaggregation [47]; DT-based algorithm [27], DTW [27], GSP-based NILM [16]; and a method of [71] based on maximum likelihood classifier with low sensitivity to noise.

State-based, probabilistic approaches [20–22, 72, 73], on the other hand, usually based on HMMs and their variants, rely on building an appliance load model using a finite state machine by learning parameters for prior distributions of appliance states. Four different HMM architectures are proposed in [20] with further improvements based on differential HMM in [21], and using expectation-maximization (EM) to generate accurate appliance state transition models in an unsupervised manner in [22]. These HMMs are good at estimating always-on loads or the appliances containing periodic components. However, since all HMMs are model-based, training data or prior knowledge is indispensable [22]. Although some HMM variants can be unsupervised, as in [21] and [22], plug-level labels are collected via intrusive load monitoring or reported by users. A NILM method using particle filtering to estimate FHMM inference is presented in [8]. Although the method is real-time capable and does not require training, it requires prior knowledge for defined appliance models and it can not separate appliances of similar power ranges. Additionally, sub-metering is not available in real-world load disaggregation scenarios. In [23], additional statistical features are considered to improve FHMM accuracy, showing improvement on two real-world datasets in 4 various evaluation metrics. In [24], a dynamic FHMM-based method is proposed as a new cloud-based on-line NILM service, which outperforms long-lasting appliances such as lighting and fridge. Similarly, sparse coding based approaches [32, 74] require training data to design appropriate basis functions and dictionaries used to transform the signals based on their sparse nature. However, the sparse coding proposed in [32] is susceptible to noise due to unlabelled loads and validated on the sum of sub-metering readings but not the real aggregate.

Compared to event-based NILM approaches, state-based NILM approaches commonly require good *a-priori* information for initialization of appliance state models or a large training dataset for good performance [22, 25, 75]. Event-based NILM approaches, on the other hand, are often easier to implement and deploy due to data

reduction via event feature extraction [16, 27]. However, as reported in [16], due to their reliance on edge detection, event-based NILM methods are susceptible to measurement noise and unknown appliances, and often misclassify appliances with similar operational power range.

Based on the dependency of the approach on a labelled training set (i.e., a diary of which appliance changed state and when), all NILM approaches can be divided into supervised, semi-supervised, and unsupervised. Supervised NILM approaches require a training phase where both aggregate measurements and the appliance-level load readings are used [76], including HMM [19], K-NN [77], sparse coding [32, 74], sparse representation [78], GSP [16, 40] and deep learning [79]. In [19], sub-metering measurements are used to build super-state HMMs, and the Viterbi algorithm is then exploited for disaggregation. The super-state HMM in [19] can disaggregate 18 appliances and it shows the capability to noise. However, like most NILM solution relying on prior appliance modelling, it cannot identify the new appliance which is not learnt. Besides the well-known steady-state power and state transient features, humidity and temperature data are also needed in [77]. NILM problem is regarded as a multi-label classification with a solution via sparse representation in [78]. Although the approach in [78] outperforms benchmarks based on graph learning and deep learning, it cannot estimate different stages of each appliance. The neural network proposed in [79] requires less training data and features power changes instead of ON/OFF operational states. However, no more than 5 appliances are identified, with only classification accuracy. In [80], the NILM methods utilising appliance feature matching are reviewed. A more systemic review of NILM approaches including state-of-the-art learning-based load disaggregation algorithms is presented in [15]. Datasets and evaluation metrics are discussed in [81]. The review of [82] focuses on the cloud computing in NILM.

However, providing an accurate labelled training set is often impractical due to the requirement for sub-metering or accurate time-diaries. Hence, *unsupervised* approaches have become popular. All methods closely related to the proposed unsupervised GSP NILM solution are grouped as follows. Group 1 comprises all traditional unsupervised approaches, which require unlabelled training data to build appliance

models or populate appliance database. In [17], an overview of unsupervised NILM methods with performance comparison is provided. Group 2 consists of NILM methods that use ‘known houses’ for building appliance models, which are then used for disaggregation in ‘unknown’ (‘unseen’) houses. Methods that do not require training before NILM disaggregation form Group 3.

Group 1 methods are usually based on hierarchical clustering or HMMs where appliance models are generated, manually or automatically, during the training phase. The early work on unsupervised NILM is presented in [20] where four low-rate NILM methods are proposed using (conditional) FHMM and Hidden semi-Markov models. This method cannot disaggregate baseload and fridge, and is prone to converge to a local minimum. More recently, new approaches that address some of the weaknesses of [20] are described in [21, 22, 58, 83] based on FHMM, differential FHMMs, additive FHMM and Hierarchical Dirichlet Process Hidden Semi-Markov Model factorial structure, respectively. A magnitude-base unsupervised NILM approach presented in [84] uses standard HMM with smoothing to obtain better features. Two FHMM-based approaches in [25, 75] exploit context information and interactions chains, respectively, to improve performances of standard FHMM.

Although the above probabilistic state-based approaches are unsupervised, they use expert knowledge to set *a-priori* values for each appliance state and require a training set (usually where appliance operations do not overlap) to build/refine the state models, such as [59, 85]. The performance of these methods depends on how well generated models approximate appliance true usage. Thus the longer the training period, the better the results. Moreover, it has not been demonstrated that these methods can be generalized across houses; that is, if applied to a house that was not included in training (‘unseen house’), it is uncertain that the methods will perform well. Unsupervised, time-series approaches, such as [27], do not build appliance models but require training periods to build a database of time-series signatures, necessary for pattern matching. Similarly, the unsupervised approach of [34], based on probabilistic sequential mining and temporal motif mining, requires extra information such as the number of appliances.

NILM methods in Group 2 assume the existence of ‘training’ or ‘known’ houses where submetering data is available and used to build appliance models or populate an appliance database, which is then used in unknown or unseen houses. In [47], for example, k-means and SVM are combined to disaggregate washing machine in an unknown house using models generated in two ‘known’ houses, showing competitive performance on large loads such as washing machine or dishwasher against the case that training on the same house. Deep learning is used in [37], where three neural network architectures are adapted to NILM with supervised training in known houses. The method performs well on a house, unseen during training. However, as any other deep learning approach, the proposed approaches require a large training set, have high computational complexity and do not perform well for multi-state appliances. Similar performance can be observed in [86], where a sequence-to-point deep neural network architecture proposed for low-rate NILM is trained on a dataset and tested on another dataset. In [87], spatiotemporal pattern network is applied to NILM via training on houses with sub-metering and testing on similar houses without sub-metering. However, extra information such as temperature and time diary is required for assessing the similarity between houses.

In [88], it is assumed that if two houses have a similar aggregate consumption during different seasons, it is likely that they will also have a similar consumption at the appliance level. Based on this assumption, instead of performing NILM to disaggregate appliance usage, K-NN is suggested in [88] for ‘similar’ houses, where sub-metering is available, to predict the disaggregated consumption without any NILM.

Though Group 2 approaches do not need to be trained on the data for the house being disaggregated, their main drawback is that they do not work well for uncommon appliances, are sensitive to outliers, require a large set of houses for training where sub-metering is possible, and cannot generalize well across different geographical locations.

Group 3 approaches are the closest to the proposed methods in this thesis as they do not require training before disaggregation takes place. An unsupervised event detector for NILM presented in [89] applies Kernel Fisher Discriminant Analysis without any training; however, it requires high sampling rates and uses current harmonics. In [90],

an unsupervised low-rate NILM approach, based on clustering and matching pursuit is proposed; however, the approach uses both active and reactive power, performs poorly for appliance loads below 400W and concludes that the results might improve only if additional features are included. An unsupervised FHMM-based approach introduced in [72] learns the appliance models on-the-fly, thus its performance gradually improves requiring some time to reach high accuracy. Apart from the aforementioned approaches proposed for residential energy disaggregation, in [91] the energy consumed by a hospital is disaggregated via non-negative matrix factorization using extra temperature and time features.

In Chapter 3, the proposed approach deviates from Group 1 contributions above, in that the proposed approach does not require any training or any expert/customer input to perform disaggregation, which makes the proposed system practical and potentially massively deployable (see [92] for a discussion about necessary features to ensure NILM practicality). In contrast to Group 2 methods, the proposed method does not require the existence of houses with sub-metering. The proposed method is event-based forming patterns of appliance signatures on-the-fly using clustering and relies on pattern matching to label the identified patterns. Since the proposed approach is signal processing based (as opposed to machine learning based), unlike the method presented in [72], the proposed approach does not need to learn appliance features over time, since it is not based on the appliance model generation.

2.1.3 HMM vs. Deep Learning vs. GSP

HMM relies on building an appliance load model using a finite state machine by learning parameters for prior distributions of appliance states. For each appliance to be disaggregated, a state transition model is generated. An example of a state transition model for an ON/OFF appliance with a standby state, such as TV, is illustrated in Fig. 2.2a. Each node in the model corresponds to a steady state, and edges between nodes are weighted by a state transition matrix learnt from labelled training data [22]. An HMM example is shown in Fig. 2.2b, where \mathbf{x} is a time-series signal of hidden transition states and \mathbf{y} refers to the observation signal, i.e., aggregated measurement in

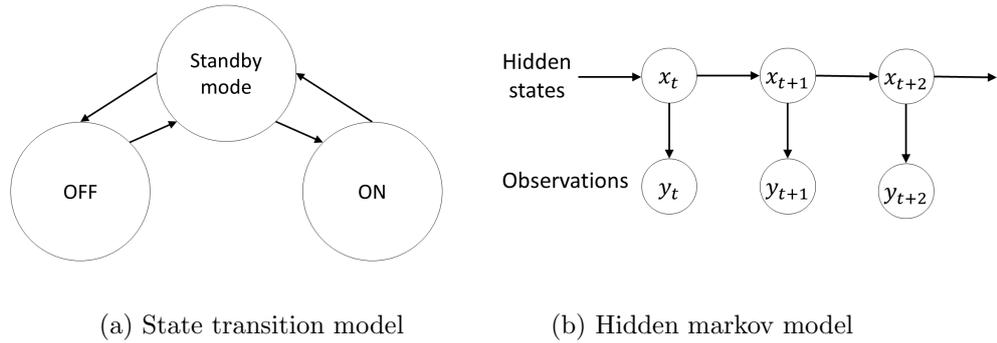


Figure 2.2: Example of a state transition model (finite state machine) and a hidden Markov model.

NILM. Then Viterbi algorithm is usually used to perform disaggregation on HMM [22]. HMM variants usually differ from the graphical architecture, via being conditioned to extra features or setting different dependency rules [20]. In [93], constrained particle swarm optimization is used to improve FHMM performance.

Deep learning, usually presented as artificial neural networks (ANN), is widely studied and applied to NILM problem in recent years. An ANN is represented as a directed graph of nodes with edges between nodes. The nodes are artificial neurons and the edges are used for passing information between neurons. Note that the information transmission can be executed between the same neuron but in distinct time stamps. Multiple layers are built for arranging neurons, including an input layer, an output layer and hidden layers between them. Each neuron in one layer is connected with every neuron in the consecutive layers. Such connections are weighted and the weights keep updating during the learning process, namely, backward pass. The forward pass transmits information to the output, thus the error between the output and the target can be calculated by an objective function. The weights' updating is in the backward direction. When being applied to solve NILM problem, ANN can perform a sequence-to-sequence estimation as in [37, 39] and a sequence-to-point estimation as in [7]. The difference is a sequence-to-sequence estimation refers to estimating a sequence of appliance-level power readings; however, only one appliance reading is estimated for a sequence-to-point case [7]. Note that the neural network proposed in [7] estimates the appliance

reading at mid-point of the sequence, thus, it is not real-time capable as future readings are needed. The performance of deep learning at difference low rates (from 6sec to 1min) is investigated in [94], showing a clear reduction in time but remaining accuracy.

HMM is a classic machine learning method, which is claimed to be good for disaggregation of always-on appliances and the appliances containing periodic components [27]. Deep learning, via artificial neural networks, usually outperforms other NILM methods for most appliances if training data is sufficient and high-quality [7]. Either training data, including plug-level sub-metering data and aggregate measurements with individual appliance usage labels without overlapping, or expert knowledge is required in HMM for defining appliance models [59]. Similarly, deep learning requires a large volume of training data and is susceptible to the wrongly claimed labels [7]. Thus, applying HMM and deep learning in the real-world NILM cases is hard, where only low-rate aggregate measurement for billing purpose is available. That is, there is no plug-level training data, extra survey or environment data. Compared with HMM and deep learning, GSP is proposed as a training-less NILM solution, with no aforementioned drawbacks. GSP is an emerging signal processing method, where the supervised GSP is claimed to succeed in NILM [40]. Note that HMM and deep learning only disaggregate appliances which models are learnt previously, thus, such methods cannot identify the new appliance which is not learnt. However, the proposed GSP relying on unsupervised clustering is expected to disaggregate appliance-level clusters of transient events including unreported new appliances.

2.2 Very Low-rate NILM Approach Overview

This section provides a detailed review of very low-rate (15-60minutes) NILM solutions. An hourly energy disaggregator is proposed in [95], 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 the study of power factors [52], which is validated on the same dataset used in [95]. Unlike [95], disaggregation results for three clusters grouped by power factor are provided in [52], achieving *F-measure* 59%-81% for all clusters. Another disaggregation method carried out on energy profile readings collected from Canadian houses is presented in [46], based on modelling piece-wise functions of hourly real energy versus external temperature for base-load, heating and cooling devices. A model-based regression method is proposed in [45] 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 the household survey. The approach relies on the assumption that space heating consumption is weather-dependent and related to household size, number of residents, etc. More recently, similar regression model is applied in [96] to predict hourly energy profile for heating and cooling demands, where the utilisation of weather feature is enriched. The drawbacks of these methods are the requirement of environment data (weather, temperature, etc.) and only performing disaggregation for cooling and heating demands which are correlated to weather.

An unsupervised approach based on semi-binary non-negative matrix factorization (SBNMF) is proposed in [97] for the 30-min NILM problem, where SBNMF and its variants are used for dictionary learning. Dictionaries are labelled by a random forest classifier utilising a pre-learnt descriptive database on the cloud. In addition, consumer feedback for improving dictionary learning is studied. Consequently, modified 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 DT, SVM and K-NN, are proposed in [26] and validated on energy profile data from a subset called IRISE within the REMODECE dataset [98] at sampling rates of 10 min and 1 hour. Three appliance cat-

egories, defined based on clusters obtained using principal component analysis, contain domestic appliance types showed in Table 2.1. 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 [28], [30] and [31]. In [28], 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 [30]. Results are presented for 15-min and hourly electricity profiles from REDD [60], REFIT [61] and AMPds [66] 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 [31], 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 [40], is applied to aggregated power measurements of known appliances down-sampled to 15 minutes in [99], 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 known appliances and a small amount of added noise to simulate energy consumed by unknown appliances for only 18 days. An unsupervised hybrid approach is proposed

in [100] 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, the number of inhabitants, their age and employment status, type of heating used, nominal power per appliance and probability of activity. FHMM and CO from NILMTK, DDSC from [32] and GSP are used for benchmarking. Note that the aggregate power signal used in [100] is not whole-house smart metering measurements like in this paper, but defined as the sum of sub-measurements, namely, the influence of different types of noise due to unknown loads are ignored in disaggregation. The conclusion is that while the performance of the supervised GSP [40], implemented using the Graph Signal Processing Toolbox of Matlab [101], 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 [32], 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 [32, 33] for details). Results are provided only for aggregated sub-metered readings of 10 known appliances, again not representative of real aggregate smart metering power consumption readings that include noise due to unknown appliances. A variant of DDSC obtains up to 55% for consumption accuracy. Long short-term memory is used to forecast the hourly power for the next 24 hours with hourly weather feature in [102]. Like the deep learning applied to low-rate NILM, it requires large volume of data for training.

Table 2.1 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 as granularity decreases, e.g., Hairdryer, Microwave, Kettle, Toaster, etc. The REDD dataset is also noted to be 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

Table 2.1: A summary of appliances disaggregated in literature at low to very low sampling rates. Each appliance is linked to the corresponding datasets, including REDD [60], REFIT [61], GREEND [103], AMPds [66], UK-DALE [104], Dataport [105], REMODECE [98] 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 [60]			[29]
Clothes dryer [60, 98, 105]+	[26, 32, 95]	[26, 31, 99]	[19, 27, 29, 33]
Clothes washer [61, 66, 98] [60, 103–105]+	[26, 30, 32, 95]	[26, 28, 31, 97]	[7, 27, 29, 37–39] [16, 47]
Dishwasher [60, 61, 66, 98] [103, 104]	[26, 30, 32, 95]	[26, 28, 31, 99]	[7, 19, 27, 33, 37, 38] [16]
Electronics [60, 66, 105]+	[30, 32]	[31]	[16, 29, 33, 39, 47]
Fridge/freezer/ fridge-freezer [60, 61, 104, 105]+	[30, 32, 95]	[28, 30, 31] [97, 99, 100]	[7, 27, 29, 33, 37, 38] [16, 39, 47, 99]
Geyser +		[99]	[99]
Hairdryer [61]			[27]
Heat pump [60, 66]	[30]		[19]
Hot water unit/ furnaces [60, 61, 66, 98]+	[26, 30, 95]	[26, 28]	[16, 27, 33]
HVAC [60, 61, 66]+	[30, 32, 45, 95]	[28, 30, 97, 100]	[16, 19, 29, 33]
Kettle [61, 104]+		[28]	[7, 16, 27, 37, 38]
Kitchen outlets [60, 61, 98]+	[30, 32, 95]	[26, 30]	[16, 27, 29, 33]
Lighting [60]+	[32, 95]	[100]	[29]
Microwave [60, 61, 98, 104]+	[26]	[26, 28]	[7, 27, 29, 33, 37, 38] [16, 39, 47, 99]
Oven [60, 61, 98]+	[26]	[26, 28]	[19, 27, 29]
PC/ICT +	[32]	[28, 100]	
Stove [60]+	[95]		[16, 33, 47]
Toaster [60, 61]			[27, 47, 99]
TV [61, 103]+	[32]	[28, 97]	[16, 27, 47]

(few unknown appliances operating simultaneously) dataset which is not the case in actual smart meter measurements, made up of 40+ appliances present in a 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, the 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 the manufacturer’s wattage or energy-consumption-per-run information. The GSP-based approach proposed in [41] 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 [99] and [100], the proposed unsupervised GSP algorithm has a Matlab execution time of fewer than 8 minutes to disaggregate an hourly profile of 78 weeks. Note that the baseload removal pre-processing step proposed in [99] fails to separate baseload from noisy hourly energy profile signal and makes no improvement to disaggregation.

Additionally, unlike the majority of the aforementioned literature, the proposed approaches do not resort to environmental data. The proposed approaches are evaluated on public datasets that closely resemble real-life ‘noisy’ smart meter measurements that include many unknown appliances. Besides *detecting* appliance use (classification only), the proposed approaches 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.

2.3 What constitutes noise in NILM and Denoising Solutions

In this section, a review of noise in NILM problem and existing approaches for improving NILM performance are discussed. As discussed in Section 2.1, current state-of-the-art solutions are susceptible to measurement noise and outliers when dealing with

real-world data and do not demonstrate sufficient accuracy [15, 17]. One reason for this is the complex nature of the NILM problem with effective solutions requiring both core physical-level signal processing - to process acquired signals reducing jitter, noise, spurious events [16, 50] - and machine learning-based clustering and classification [15, 19, 27, 29].

In practice, sensor noise, transient spikes and signal fluctuations around an appliance’s mean operating power inevitably appear in real-world electrical load meter measurements, as illustrated in Fig. 2.3. In Fig. 2.3a, transient spikes are present due to

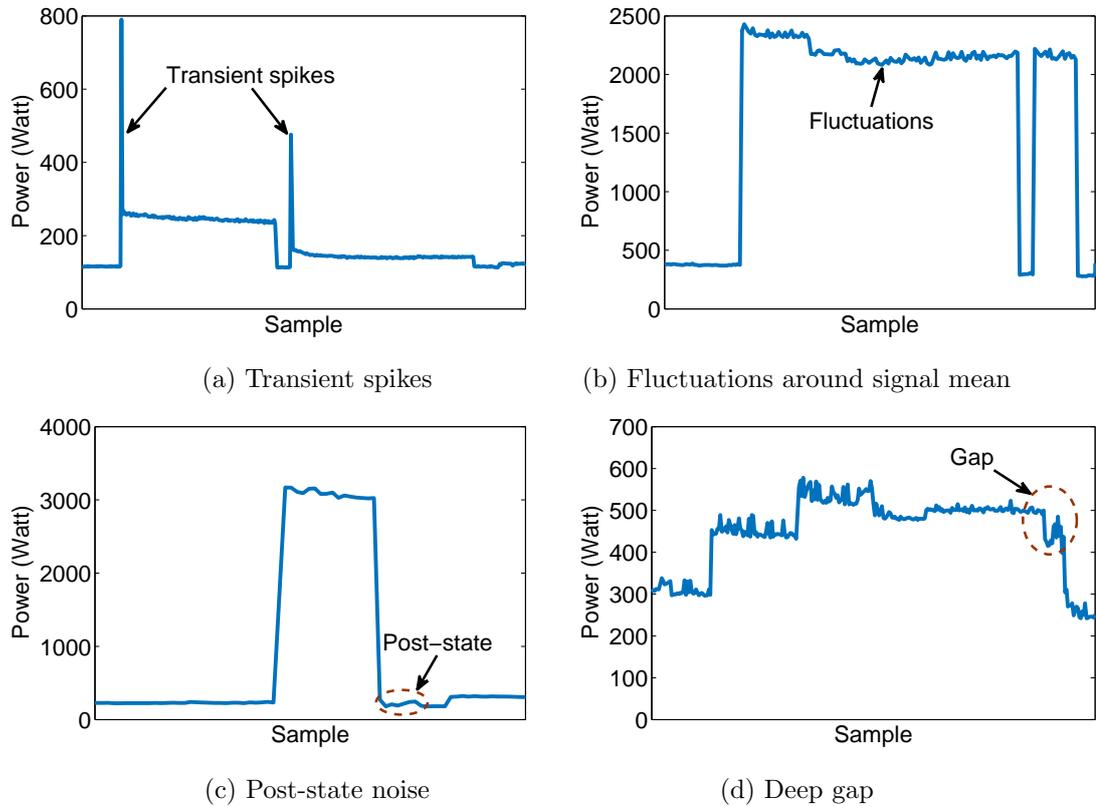


Figure 2.3: Typical “disaggregation” noise in electrical loads measurements, observed in House 17 of the REFIT dataset.

sensor measurement noise and fast changes in the signal that cannot be captured at low rates. The ‘fluctuations’ of power level around its mean value during an appliance run can be observed in Fig. 2.3b, which often results in the NILM algorithm misclassifying (part of) this appliance run as another appliance that operates at low power levels. In

Fig. 2.3c, a kettle is switched on/off, which causes ‘pre/post state’ signal fluctuations, common for appliances with a heating element. Fig. 2.3d shows a ‘deep gap’ caused by the simultaneous switching on of a PC and fridge-freezer.

All the above examples are considered to be “disaggregation noise” as they negatively affect NILM algorithm performance, which would ideally deal with rectangular-shaped signals with fairly unique amplitude values. For state-based NILM approaches, measured signal processing is based on signal smoothing with the main objective of removing outliers so that state conditions can be identified. Event-based NILM approaches are more sensitive to abrupt state transitions [16], which make edges undetectable. Hence, the key to successful disaggregation lies in efficient smart meter signal processing to ensure that the edges are sharp enough to be effectively detected by edge detection and transients, that cannot be effectively captured with low sampling rates (of the order of 1-60 seconds), are removed. Otherwise, regardless of the effectiveness of the following classification step, the NILM output would not be sufficiently reliable.

Next, the signal processing methods that have been proposed for NILM are reviewed. Total variation regularization is used prior to the additive FHMM-based NILM algorithm to remove outliers and minimise the influence of rarely used appliances in [21]. Median filtering is commonly used to remove spikes and noise, and smooth the signal [25, 33, 58, 72, 84, 106] where the window size of the median filter is heuristically chosen based on the signal granularity. For example, median filtering is exploited twice in [106], for total noise removal and partly for signal smoothing. Median filtering together with smoothing is applied in [84] to an unsupervised HMM-based NILM approach. Various smoothing filters for NILM are investigated in [50], including median filter, mean filter, kernel-weighted average filter and the possible combination of multiple filters. Although kernel filter performs the best, a median-mean filter is finally chosen due to the high complexity and cost of kernel filters. A neural network architecture performing dimensionality reduction is introduced in [37] as a denoising step. Down-sampling can also be regarded as a signal processing/noise removal method, since the majority of signal fluctuations are filtered out and most appliance state transition edges will be sharpened after down-sampling [16, 22, 56]. In [56], the baseload signal is

heuristically estimated and removed before GSP-based disaggregation.

Signal processing and “denoising” can lead to a cleaner signal that would improve the subsequent classification. However, signal processing alone cannot solve the issue of similarity of appliance loads, that is, very close operational mean power values of two or more appliances. An inference approximation method is proposed in [21] to refine NILM results, where Additive Factorial approximate maximum a posteriori is proposed taking advantage of the additive structure of FHMM and the observation of aggregate power. This can also be addressed via a probabilistic search method, reported in [16]. That is, simulated annealing is added after the primary GSP-based NILM to refine two-state (e.g., ON/OFF) appliance identification by optimising the difference between the power measurements and the corresponding primary power estimate according to the possible combination of multiple appliances which are switched on simultaneously, reported in [16]. In [16], post-processing approaches are proposed for NILM based on convex optimisation tools, where combinatorial boolean quadratic problems are solved in multiple ways. In Chapter 3, each rising edge, which refers to a switching-on or upward state transition event, is matched with one nearest pair from the cluster with magnitude-wise closest falling edges. Methods of [16] and [40] have drawbacks in that: 1) these algorithms are sensitive to spikes and fluctuations; 2) low-load appliances and appliances with low operational states in multi-state appliances are often misclassified; 3) these algorithms usually perform poorly when disaggregating long-lasting, low-load appliances due to the difficulty of accurate detection and feature matching. Chapter 4 aims at addressing these drawbacks and providing a universal signal processing solution suitable for all event-based NILM methods.

2.4 Graph Signal Processing Preliminaries

This section reviews the background of GSP and gives an overview of the concepts of GSP relevant to the entire thesis. GSP is an emerging approach that provides robust means for signal denoising [62], clustering [107], and classification [16], where complex relationships between samples of high dimensional data are represented using graphs. The emerging signal processing concepts on graphs are systemically introduced in [51]

and [63]. A data classifier via regularization on graphs is proposed in [108], then applied for depth map denoising in [109]. Recently, the classification via graph regularization has been proposed for tackling the NILM problem, via supervised approaches in [40] and improved design in [16].

In this work, GSP is applied as a robust clustering tool, explained in Chapter 3 and used for benchmarking in the following chapters. In Chapter 4, GSP is exploited as an effective physical signal filter, which can combat NILM sensitivity to measurement noise and the influence of unknown appliances.

GSP is based on *graph signals* obtained by indexing a dataset by nodes of a graph. The basic idea is to represent a dataset using a graph defined by a set of nodes and a weighted adjacency matrix. Each node in the graph corresponds to an element in the dataset while the adjacency matrix defines all edges in the graph and their weights, where assigned weights reflect the degree of similarity or correlation, between the nodes.

Before moving on to GSP concepts, the notation rules for the whole thesis should be clarified. Bold upper-case and lower-case letters are used to denote matrices and vectors, respectively. The entry in the i -th row and j -th column of matrix \mathbf{A} is denoted by $A_{i,j}$. x_i denotes the i -th element in vector \mathbf{x} . $\mathbf{A}_{a_1:b_1, a_2:b_2}$, for $a_1 < b_1$ and $a_2 < b_2$, represents the sub-matrix in \mathbf{A} , with rows from a_1 to b_1 and columns from a_2 to b_2 . Similarly, $\mathbf{x}_{a:b}$, for $a < b$, denotes the vector $[x_a, \dots, x_b]$. \mathbf{I} is the identity matrix, i.e., $\mathbf{I} = \text{diag}(1, \dots, 1)$. Sets are denoted by upper-case calligraphic letters.

Given an acquired set of N -length discrete measurements \mathbf{x} , an undirected graph $\mathcal{G} = (\mathcal{V}, \mathbf{A})$ is defined, where $\mathcal{V} = \{v_0, v_1, \dots, v_{N-1}\}$ is a set of vertices and each vertex v_i corresponds to one acquired measurement. The adjacency matrix $\mathbf{A} \in \mathbb{C}^N$ is $N \times N$, defining all edges between nodes in the graph and their weights. That is, $A_{i,j}$ corresponds to the weighted edge from v_i to v_j where the weight depends on the relationship between vertices v_i and v_j . The values of $A_{i,j}$ are often naturally defined by the physical meaning of the collected data. If that is not the case, a Gaussian kernel weighting function is often used to define the values of $A_{i,j}$, as in [16, 51]:

$$A_{i,j} = \exp \left\{ -\frac{|\text{dist}(x_i, x_j)|^2}{\rho^2} \right\}, \quad (2.4)$$

where ρ is a scaling factor and $dist(x_i, x_j)$ can be Euclidean distance.

For example, in [16], the graph is designed to perform supervised classification of signal samples, where each signal sample x_i is indexed by a graph vertex v_i . The adjacency matrix then carries information about the correlation between signal samples and is defined as:

$$A_{i,j} = \exp\left(-\frac{\|x_i - x_j\|_2^2}{\rho^2}\right). \quad (2.5)$$

On the other hand, in [62,63], the graph is designed to perform denoising of a time-series signal \mathbf{x} , where $A_{i,j}$ is set to zero if v_i and v_j are not time consecutive samples, that is:

$$A_{i,j} = \begin{cases} \exp\left(-\frac{\|x_i - x_j\|_2^2}{\rho^2}\right), & \text{for } |i - j| \leq 1 \\ 0, & \text{for } |i - j| > 1. \end{cases} \quad (2.6)$$

A vector $\mathbf{s} \in \mathbb{R}^N$ is then defined as the *graph signal* that maps $\mathcal{V} \rightarrow \mathbb{R}$ [51], where each element s_i represents the function value at node v_i . The graphs and signals on graphs defined above can be conveniently used to represent very different data structures, such as time series, images, sensors, tracked objects, social networks, hyperlinked documents etc. [51,63].

Since the majority of natural signals are piecewise smooth, signal global smoothness is often used as a prior for regularization in different inverse problems [110,111], and is defined as [112]:

$$S_p(\mathbf{s}) = \frac{1}{p} \sum_{i \in \mathcal{V}} \left[\sum_{j \in \mathcal{N}_i} A_{i,j} (s_j - s_i)^2 \right]^{\frac{p}{2}}, \quad (2.7)$$

where $\mathcal{N}_i \subseteq \mathcal{V}$ is the set of vertices that are connected to v_i . For $p = 2$, it becomes the graph Laplacian quadratic form [62]. The global smoothness of a graph reflects the piecewise smoothness of the signal with respect to the underlying graph structure, i.e., if a graph signal is piecewise smooth, the global smoothness of its underlying graph is generally small.

Let \mathbf{D} be an $N \times N$ diagonal matrix with entries on the main diagonal given by

$D_{i,i} = \sum_j A_{i,j}$. Let \mathbf{L} denote the graph Laplacian operator [63] defined as:

$$\mathbf{L} = \mathbf{D} - \mathbf{A}. \quad (2.8)$$

Then $S_2(\mathbf{s})$ in Eq. (2.7) can be expressed in terms of the Laplacian matrix as [51]:

$$S_2(\mathbf{s}) = \frac{1}{2} \sum_{i,j} A_{i,j} (s_j - s_i)^2 = \mathbf{s}^T \mathbf{L} \mathbf{s}. \quad (2.9)$$

If \mathbf{s} is piecewise smooth with respect to underlying graph structure, then $\mathbf{s}^T \mathbf{L} \mathbf{s}$ is generally small. The global graph-signal smoothness can effectively be used as a prior for regularization, since the Laplacian regularizer $\mathbf{s}^T \mathbf{L} \mathbf{s}$ is a good measure of variation in the signal modulated by weights in \mathbf{A} . Then, to find the smoothest signal, the global smoothness minimisation problem can be formulated as:

$$\arg \min_{\mathbf{s}} \left\| \mathbf{s}^T \mathbf{L} \mathbf{s} \right\|_2^2. \quad (2.10)$$

Then the expansion of $\mathbf{s}^T \mathbf{L} \mathbf{s}$ can be formulated as:

$$\begin{aligned} \mathbf{s}^T \mathbf{L} \mathbf{s} &= \mathbf{s}_{1:n} \mathbf{L}_{1:n,1:n} \mathbf{s}_{1:n} + \mathbf{s}_{1:n} \mathbf{L}_{1:n,n+1:N} \mathbf{s}_{n+1:N} \\ &+ \mathbf{s}_{n+1:N}^T \mathbf{L}_{n+1:N,1:n} \mathbf{s}_{1:n} + \mathbf{s}_{n+1:N}^T \mathbf{L}_{n+1:N,n+1:N} \mathbf{s}_{n+1:N}. \end{aligned} \quad (2.11)$$

Note that $\mathbf{s}_{1:n}$ refers to a graph signal corresponding to the vector of n known samples used for training in the supervised classification approaches [40, 113]. Thus $\mathbf{s}_{n+1:N}$ corresponds to the rest samples to be classified.

Since \mathbf{D} is a diagonal matrix, \mathbf{L} is also diagonally symmetric. Thus, since the first term in Eq. (2.11) does not affect minimisation, minimisation Eq. (2.10) is simplified as:

$$\arg \min_{\mathbf{s}} \left\| \mathbf{s}^T \mathbf{L} \mathbf{s} \right\|_2^2 = \arg \min_{\mathbf{s}_{n+1:N}} \left\{ 2 \mathbf{s}_{n+1:N}^T \mathbf{L}_{n+1:N,1:n} \mathbf{s}_{1:n} + \mathbf{s}_{n+1:N}^T \mathbf{L}_{n+1:N,n+1:N} \mathbf{s}_{n+1:N} \right\}. \quad (2.12)$$

As an unconstrained quadratic programming problem, this minimisation has a closed

form solution [114, 115]:

$$\mathbf{s}^* = \mathbf{L}_{n+1:N, n+1:N}^\# (-\mathbf{s}_{1:n}) \mathbf{L}_{1:n, n+1:N}^T, \quad (2.13)$$

where $(.)^\#$ denotes the pseudo-inverse matrix. \mathbf{s}^* is the smoothness optimisation solution, i.e., a solution for $\mathbf{s}_{n+1:N}$ that minimises the total graph variation. Eventually classification decisions are made based on \mathbf{s}^* [16, 40].

2.5 Performance Evaluation Metrics for NILM

This section summarises all evaluation metrics for NILM used in the following chapters.

2.5.1 *F-measure*

The evaluation metrics used are adapted from [20], *Precision (PR)*, *Recall (RE)* and *F-measure (F_m)*. As in [20], *true positives (TP)* are separated into two cases, *accurate true positive (ATP)* and *inaccurate true positive (ITP)*. *ATP* denotes the correct claim the detected appliance was used and the corresponding events are correctly named; *ITP*, on the other hand, denotes the correct claim the detected appliance was used but the corresponding events are falsely named; *false positive (FP)* represents an incorrect claim that the appliance was not used; and *false negative (FN)* indicates that the appliance operational events were not detected. Then:

$$PR = ATP / (ATP + FP) \quad (2.14)$$

$$RE = ATP / (ATP + ITP + FN) \quad (2.15)$$

$$F_m = 2 \cdot (PR \cdot RE) / (PR + RE), \quad (2.16)$$

PR represents the event detection accuracy where high *PR* reflects low *FP*, and *RE* represents the events detection strength and clustering accuracy where lower *FN* and *ITP* result in a higher *RE*. F_m balances *PR* and *RE*.

F_m can be calculated from the estimation and the ground truth of ON/OFF states

instead of the disaggregated appliance-level power/energy result. Thus, F_m is used only for evaluating classification accuracy, that is, when the state of an appliance changed, but not how much energy was consumed. Disaggregation accuracy indicates the error between the estimation of appliance-level power consumption and its ground truth.

2.5.2 Disaggregation Accuracy

The disaggregation accuracy metric for Appliance m is defined as:

$$Acc^m = 1 - \frac{\sum_{i=1}^N |\hat{P}_i^m - P_i^m|}{2 \sum_{i=1}^N P_i^m}, \quad (2.17)$$

where N is the number of samples, P_i^m and \hat{P}_i^m refer to the measured power of Appliance m at time instant i and its estimated value after disaggregation, respectively. This evaluation metric, 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. Namely, Acc^m demonstrates for each appliance, the error between actual power consumption and its estimate, is used in [37, 60].

For comparison with recent approaches [25, 58], the variance of Eq. (2.17) $Acc.$ is also used to demonstrate overall disaggregation accuracy for each house, defined as,

$$Acc. = 1 - \frac{\sum_{i=1}^N \sum_{m \in \mathcal{M}} |\hat{P}_i^m - P_i^m|}{2 \sum_{i=1}^N \bar{P}_i}, \quad (2.18)$$

where \bar{P}_i denotes the observed total power consumption at time instance i . Therefore, both Acc^m and $Acc.$ are applicable for evaluation of the NILM methods where power consumption are estimated instead of estimation of ON/OFF states. Acc^m is used for evaluation per appliance and $Acc.$ is for the whole house. Besides, both Acc^m and $Acc.$ demonstrate normalised error instead of actual error, enabling the disaggregation result comparison among various appliances.

2.5.3 Error in Estimating Total Power Consumption

A metric is also introduced, complimentary to disaggregation accuracy, to further explain the performance of NILM approaches. The error rate of total power consumption (TER) measure, using the same notation as above, is defined as:

$$TER = \left| \sum_{i=1}^N \hat{P}_i^m - \sum_{i=1}^N P_i^m \right| / \sum_{i=1}^N P_i^m, \quad (2.19)$$

Like $Acc.$, TER is a house-level metric. However, in contrast to $Acc.$, TER ignores the appliance-level disaggregation error and does not need appliance-level power consumption estimation result. That is, it only requires the estimation of total power consumption.

2.5.4 Disaggregation Error Measure

Besides, the Disaggregation Error Measure (DEM) is exploited to assess overall performance with respect to the noise measure (NM) of [19] to understand the performance of the proposed algorithms in correlation with the ‘noisiness’ of the dataset:

$$DEM = \frac{\sum_{i=1}^N \left| P_i - \hat{P}_{base} - \sum_{m \in \mathcal{M}} \hat{P}_i^m \right|}{\sum_{i=1}^N P_i}, \quad (2.20)$$

where \hat{P}_{base} refers to the estimated baseload. In contrast to the sample-by-sample error measure per appliance such as Acc^m , DEM measures the total disaggregation error per sample. Therefore, DEM is suitable for the scenarios that disaggregation results are not presented in appliance-level.

2.5.5 Match Rate

In addition to the above measures, an evaluation metric applicable to load disaggregation is utilised as disaggregation match rate for appliance m defined as:

$$MR^m = \frac{\sum_{i=1}^N \min\{\hat{P}_i^m, P_i^m\}}{\sum_{i=1}^N \max\{\hat{P}_i^m, P_i^m\}}. \quad (2.21)$$

MR is a metric where the evaluation is based on the overlapping rate of true and estimated energy, and stated to have best overall performance in [81] and references therein. Similar to Acc^m , MR^m requires appliance-level disaggregation estimate per sample. It was also used in [97] for very low-rate disaggregation.

2.5.6 Metrics for Very Low-rate NILM

F_m provides classification accuracy, i.e., the accuracy in determining when an appliance was running and whether each sample is correctly identified, whereas Acc^m represents estimation accuracy, namely the accuracy in estimating the energy each appliance used for each sample, and $Acc.$ is an overall house-level presentation of Acc^m . While MR represents estimation accuracy for the selected whole period. Acc^m and MR are believed as better metrics for very low-rate NILM than F_m and similar classification metrics. As it is claimed in [116] classification accuracy metrics used for power disaggregation performance evaluation, including F_m and event-based confusion matrix well-cited in NILM works, will drop as data sampling rate decreases from few seconds to 15 minutes. Apparently, classification accuracy metrics are not suitable for disaggregation on the hourly profile. A more detailed discussion of different metrics used in NILM can be found in [81] and other overview papers.

Classification accuracy metrics are not used for validation of very low-rate NILM in Chapter 5. Instead, since the NILM use-case is on the problem of how reliably the relative contribution of individual loads contributing to the aggregate consumption can be estimated, the consumption accuracy Acc^m is utilised as in Eq. (2.22), where \hat{E}_d^m is defined as Eq. (2.23).

$$Acc^m = 1 - \frac{\sum_{d=1}^N |\hat{E}_d^m - E_d^m|}{2 \sum_{d=1}^N E_d^m}, \quad (2.22)$$

$$\hat{E}_d^m = \sum_{h=1}^{24} \hat{E}_{(d-1)*24+h}^m, \quad (2.23)$$

Acc^m in this case 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.

MR is also utilised in Chapter 5 as in [97]. The daily disaggregation match rate is defined as MR^m in Eq. (2.24), which shows the overall matching accuracy between actual and estimated daily energy consumption for Appliance m .

$$MR^m = \frac{\sum_{d=1}^N \min\{\hat{E}_d^m, E_d^m\}}{\sum_{d=1}^N \max\{\hat{E}_d^m, E_d^m\}}. \quad (2.24)$$

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 sudden power change features, numerous simultaneous operation of multiple loads, the stronger negative influence of different types 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 [81]. From [37], Acc^m is a good metric for demonstrating the error between estimation and actual measurements when such error is generally small. For very low-rate disaggregation where over-estimation 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 $Acc^m < 0$; the other one Disaggregator 2 fails to disaggregate Appliance m and returns nothing, but resulting in $Acc^m = 0.5$ by Eq. (2.22). Such Acc^m results intuitively mean Disaggregator 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 cannot give any feedback. 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.

Chapter 3

Low-rate Power Disaggregation based on Graph Signal Processing¹

Notations

K	Acceptable precision of a cluster
T_0	Initial threshold
T_N	Adaptive threshold for negative events
T_P	Adaptive threshold for positive events
$\Delta\mathbf{P}$	Differential power signal
α	Weight factor for solution of magnitude graph
β	Weight factor for solution of time graph
\mathbf{A}	$N \times N$ adjacency matrix
\mathbf{C}^N	Negative cluster of a cluster pair
\mathbf{C}^P	Positive cluster of a cluster pair
\mathbf{D}	$N \times N$ degree matrix
\mathbf{L}	$N \times N$ degree matrix

¹This chapter is mainly based on the work that appeared in IEEE Access [117]. The Python codes for the NILM approach presented in [117] can be accessed via https://github.com/loneharoon/GSP_energy_disaggregator.

\mathbf{P}	Aggregate active power signal
Φ^P	A set of magnitude differences between positive target and each negative candidate
Φ^T	A set of time intervals between positive target and each negative candidate
Π	A set of appliance-state change events
\mathbf{s}	$N \times 1$ graph signal
\mathcal{C}	A set of all clusters
\mathcal{G}	An undirected graph containing N vertices
\mathcal{V}	A set of N vertices
μ_i	Mean value of all elements in a cluster
μ_m	Mean value of Gaussian distribution of Appliance m
ρ	Scaling factor for Gaussian kernel weighting
σ_i	Standard deviation of all elements in a cluster
σ_m	Standard deviation of Gaussian distribution of Appliance m
q	Threshold for acceptable graph signal estimation
r	Radius of a quadrant

3.1 Introduction

For tackling the NILM problem, in this chapter, a new, *blind*, approach is proposed for low-rate electricity measurements that do not require any training. The proposed approach disaggregates any aggregate active power dataset without any prior knowledge, including knowledge of appliances contributing to the aggregate or their number. It relies on graph signal processing (GSP) [51], an emerging field based on representing a dataset using a discrete signal indexed by nodes of a graph. GSP offers an alternative to conventional signal processing approaches by embedding the structure of signals onto a graph, leading to a powerful scalable and flexible approach suitable for a range of applications (see [51,63,113,114] and references therein). A GSP-based NILM approach is proposed in [40]. However, the approach of [40] is supervised and employs GSP for data classification only.

In contrast to traditional machine-learning approaches, such as Hidden Markov

Model (HMM), that require plenty of observations to construct a graph, the proposed graph signal processing approach takes an intuitive approach in constructing a graph without relying on the signal’s statistics [57]. Thus, it is expected that the proposed approach will work well in the absence of a training dataset, unlike traditional HMM-based and other machine learning methods [8, 21, 22, 58, 59].

Specifically, for NILM disaggregation of the active power signal, GSP is used three times: first for robust event detection, then to perform clustering, and finally for feature matching. The approach is event-based and relies only on time-series data without any training, where good accuracy is demonstrated using two open-access datasets: REDD [60] and REFIT [61].

This chapter is organized as follows. In section 3.2, the preliminaries of training-less GSP clustering is introduced. Section 3.3 describes the proposed disaggregation methodology, followed by an analysis of the proposed algorithm and its limits in Section 3.4. Section 3.5 presents a case study. Section 3.6 describes the experimental setup followed by the results and discussion. The research findings are concluded in Section 3.7.

3.2 Training-less Graph Signal Processing based Clustering

Note that the proposed training-less GSP-based clustering is developed from the supervised classification approaches [40, 113] requiring training, which are reviewed in Section 2.4. Given an acquired set of measurements \mathbf{x} , the underlying graph is built as $\mathcal{G} = (\mathcal{V}, \mathbf{A})$, where the set of graph nodes \mathcal{V} are mapped by graph signal \mathbf{s} . Similar to [16, 40], adjacency matrix \mathbf{A} is empirically weighted by Gaussian kernel weighting function as Eq. (2.5). Then its global smoothness can be expanded as:

$$\mathbf{s}^T \mathbf{L} \mathbf{s} = s_1 \mathbf{L}_{1,1} s_1 + s_1 \mathbf{L}_{1,2:N} \mathbf{s}_{2:N} + \mathbf{s}_{2:N}^T \mathbf{L}_{2:N,1} s_1 + \mathbf{s}_{2:N}^T \mathbf{L}_{2:N,2:N} \mathbf{s}_{2:N}. \quad (3.1)$$

Note that Eq. (3.1) is the same as Eq. (2.11), except n is set to 1. That is, a vector of known samples used for training in the supervised classification approaches [40, 113] is replaced with a randomly picked sample, s_1 . Thus the closed form minimisation solution becomes:

$$\mathbf{s}^* = \mathbf{L}_{2:N,2:N}^\#(-s_1)\mathbf{L}_{1,2:N}^T. \quad (3.2)$$

In the proposed unsupervised GSP-based NILM approach, Eq. (3.2) is always exploited to calculate global smoothness minimisation solutions for all graphs.

3.3 Proposed Disaggregation Algorithm

In this section, the proposed disaggregation algorithm is described, by first presenting an overview of the proposed approach, and then describing each of its building blocks.

3.3.1 Algorithm Overview

An event-based algorithm is proposed for finding the solution to the disaggregation problem above. Event-based methods first identify windows of *events*, i.e., statistically significant changes in active power that could indicate that one or more appliances have changed their operational state (for example, switched on/off). Event detection is usually done via edge detection with fixed or adaptive thresholds [27]. After events have been identified, relevant features are extracted from each event window. Finally, the extracted features are classified into pre-defined appliance classes using a model defined during training.

The proposed algorithm follows the above steps, performing data filtering to adapt edge detection thresholds, and clustering to identify events and extract features - active power edges. Then, it replaces the final conventional classification step with a feature matching step.

The proposed method does not require any prior knowledge about the house or appliances therein, such as the number of used appliances and/or their type. In contrast to probabilistic methods that learn the appliance model, offline or online, the proposed

signal processing approach does not rely on probabilistic modelling and is not sensitive to adding and removing appliances from the house. Moreover, it performs equally well for frequently used and uncommon appliances. It has been shown to work on aggregate data in the presence of noise and unknown loads and starts to disaggregate without needing a pre-disaggregation training period.

As is common practice [15, 80], the method works on sliding time windows, whose duration can be adjusted. Window sizes from a month, a week down to a day and an hour were tested, but shorter windows are also possible. After each window has been processed, disaggregated appliances are named using a current database, whose contents are updated with new signatures if an appliance is not found. If the database is empty, appliances would be added as they are disaggregated and arbitrary labelled (e.g., Appliance X), until labels are provided, via an app or a web-interface.

Note that, as a training-less unsupervised approach that does not rely on appliance modelling, different power states of multi-state appliances as well as different modes of operation of appliances with different operation cycles, such as air conditioning, are treated as separate appliances. In the final labelling stage, these appliances are labelled as the same appliance using the database.

As with all low-rate NILM algorithms, power states that are magnitude-wise very similar cannot be separated. In Section 3.4, the bound for which two power states can be separated is quantified, depending on load fluctuation and magnitude difference between the power states.

The computational complexity of the proposed algorithm is within both Class ‘P’ in time and class ‘PSPACE’ in space [118], so it is real-time and real-world applicable, and suitable for online applications. The flowchart of the overall method is shown in Fig. 3.1. The following two subsections detail the two key building blocks in Fig. 3.1.

3.3.2 Edge Detection and Clustering

The objective of this step is event detection and feature extraction. Let $\Delta P_i = P_{i+1} - P_i$, for $i = 1, \dots, N - 1$, denote the power variation signal between adjacent aggregate power readings, where aggregate power signal is N -length \mathbf{P} . Note that each $\Delta P_i >$

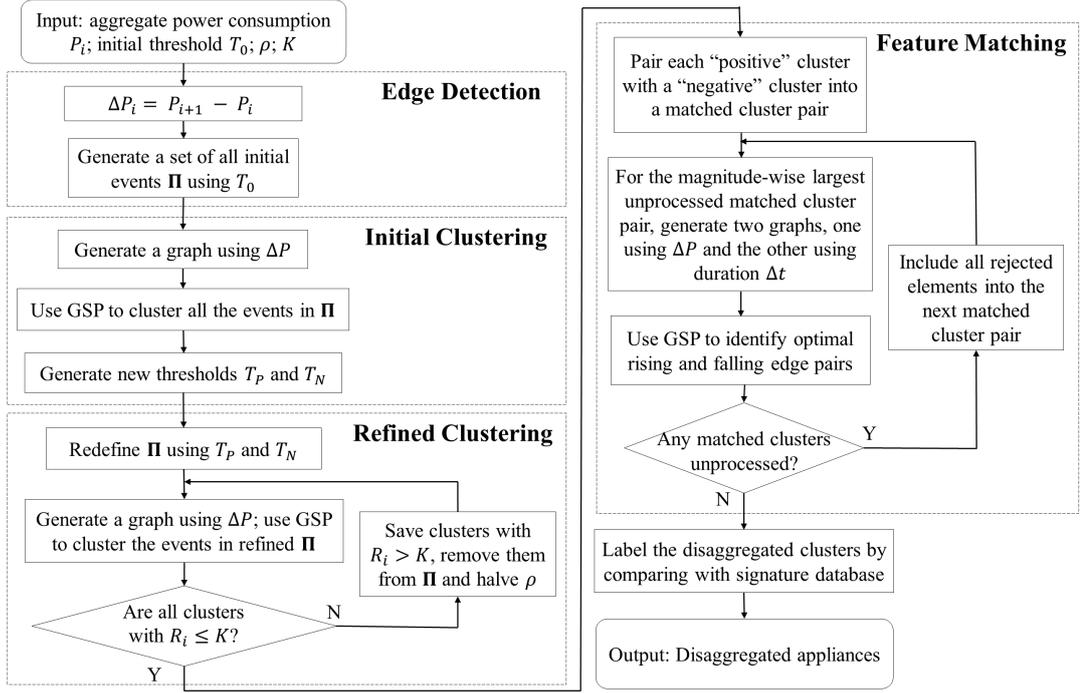


Figure 3.1: Flow chart of the proposed algorithm.

0 is defined as an increasing edge and $\Delta P_i < 0$ are decreasing edges. An event is defined as a statistically significant change in ΔP_i that indicates appliance switching on/off or state transitions in multi-state appliances, such as washing machine and air conditioner. Conventionally, events are detected using fixed or adaptive thresholding, where the threshold needs to be large enough to filter out power variations of the same appliance but small enough to capture low loads. In the following, a GSP-based method is proposed to find the optimal adaptive threshold for event detection based on the available power readings.

An initial, small, threshold T_0 (in the order of Watts) is defined to avoid detecting stand-by settings; thus, all

$$\Delta P_i \in (-\infty, -T_0) \cup (T_0, \infty) \quad (3.3)$$

will form a set of candidates $\mathbf{\Pi}$ for event detection. Note that $|\Delta P_i| > T_0$ could indicate an event but it could also occur purely due to power load fluctuations (i.e., power load

noise). Then the latter cases are filtered out from $\mathbf{\Pi}$, via iterative GSP-based adaptive thresholding and clustering.

Clustering is performed in two stages. Both stages consist of consecutive passes through the data samples in $\mathbf{\Pi}$, wherein each pass a cluster is formed from all data samples most correlated to the first data sample by applying regularization on the underlying graph. In Stage 1 the initial threshold T_0 is used to filter out low-magnitude edges, while in Stage 2 some of the clusters are refined by adapting the threshold to the data samples.

Specifically, Stage 1 clustering is started by building a graph using samples in $\mathbf{\Pi}$, i.e., all samples ΔP_i that are greater than T_0 or less than $-T_0$, associating them to nodes v_i of a graph. $A_{i,j}$ denotes the weight of an edge from node v_i to v_j that depends on the level of correlation between $x_i = \Delta P_i$ and $x_j = \Delta P_j$, calculated as in Eq. (2.5) using the Euclidean distance measure. s_1 is set to 1 if $\Delta P_1 > T_0$ and -1 otherwise, and initialise all $s_j = 0$, for $j > 1$. In the first pass, all samples are clustered statistically similar to s_1 .

To do that, Eq. (3.2) is calculated. If $s_j^* > q$, where q is a constant fixed through all iterations, then ΔP_j is assigned to the first cluster of events (together with s_1), and is removed from $\mathbf{\Pi}$. This way, the first cluster \mathbf{C}_1 of events are formed. In the second pass, it again starts with the first remaining element in $\mathbf{\Pi}$ and the procedure of generating the graph and calculating Eq. (3.2) using unclustered $\Delta P'_i$ s is repeated to form a cluster of events statistically similar to the new s_1 . Again, only $\Delta P'_i$ s for which $s_i^* > q$, will be added to the next cluster of events. The passes continue through the remaining data samples until all candidate events are clustered and $\mathbf{\Pi}$ becomes an empty set.

Note that after Stage 1, each cluster will comprise purely positive or purely negative edges. Let μ_i and σ_i denote, respectively, the mean value and the standard deviation of Cluster \mathbf{C}_i . Instead of setting a constant threshold or utilising the variance of a cluster, the *quality* of cluster \mathbf{C}_i is evaluated by *relative standard deviation (RSD)*, R_i :

$$R_i = \left| \frac{\sigma_i}{\mu_i} \right|. \quad (3.4)$$

Note that the lower the R_i , the higher the *quality* of cluster \mathbf{C}_i in the sense that the clustered points will be closely grouped together around the mean, indicating good statistical similarity between the cluster elements. If the mean of the cluster is low, the samples will be more prone to noise, thus cluster elements need to be grouped closer together. Note that although the variance is commonly used for evaluating the precision of a cluster, it is not suitable for evaluating the *quality* of a disaggregated cluster in the proposed algorithm. Each disaggregated cluster corresponds to a unique appliance operating at various power ranges, thus the cluster's *quality* should be correlated with its average magnitude. Therefore, for two clusters with the same variance, the cluster with the larger mean would have better *quality*. For example, assume that two clusters of transient events are disaggregated with average power magnitudes around 2500 Watt and 200 Watt, respectively. The fluctuations of both clusters are magnitude-wise similar around 100 Watt, that is, the variances of both clusters are similar. The cluster of 2500 Watt transient events is labelled as a high-consuming appliance such as microwave, and the other cluster might be labelled as a lower-consuming appliance such as fridge. Obviously, the cluster containing 200-Watt fridge transients has greater likelihood than the one containing 2500-Watt microwave transients. However, their variances are similar and unavailable to distinguish the *quality* of disaggregated clusters. Therefore, the proposed *RSD* is more suitable for cluster *quality* evaluation, making use of the mean value of each cluster.

The lowest mean value μ_i of the cluster among all clusters of positive elements and the highest μ_i of the cluster among those of negative elements will determine T_P and T_N thresholds used for positive and negative edges, respectively. That is, a set of candidate events $\mathbf{\Pi}$ is redefined as:

$$\mathbf{\Pi} = \Delta P \in (-\infty, T_N) \cup (T_P, \infty). \quad (3.5)$$

Here an example of Stage 1 clustering results is demonstrated in Table 3.1. Based on the aforementioned rules, in this case μ_4 and μ_5 are set as adaptive thresholds T_P and T_N , respectively. It can be observed from Table 3.1 that, R_4 and R_5 are extremely higher than R_i values under 10% for other clusters. It is due to loads with low working

Table 3.1: Stage 1 clustering results for House 2 from REFIT dataset.

Cluster No.	μ_i	σ_i	R_i
\vdots	\vdots	\vdots	\vdots
3	387.05	34.29	8.86%
4	144.72	99.05	68.44%
5	-117.37	93.21	79.42%
6	-392.65	33.29	8.48%
\vdots	\vdots	\vdots	\vdots

power, measurement noise and small fluctuations are grouped into the same clusters. Thus, adaptive thresholds are set to μ_4 and μ_5 in order to filter out falsely clustered noise and fluctuations.

In the cluster refining stage (Stage 2), the same GSP-based clustering iterations are carried out as above but only re-clustering elements in the low-quality clusters that have $R_i > K$, by halving ρ in Eq. (2.5), or effectively reducing the edge weights for the same correlation, in each pass, where K is a heuristically obtained constant defining the acceptable precision of a cluster. After each pass, all resulting clusters with RSD $R_i \leq K$, will be stored as the final clusters, passed to the following Feature Matching step, and removed from $\mathbf{\Pi}$. The clustering will end when there are no remaining elements in $\mathbf{\Pi}$.

Finally, the smallest-sized clusters (clusters with the least number of elements) are merged into larger-sized clusters to make the number of clusters comprising increasing power edges equal to the number of clusters comprising decreasing power edges.

3.3.3 Feature Matching

Since the final clusters contain “positive” clusters (comprising increasing power edges) and the same number of “negative” clusters (with decreasing power edges), each “positive” cluster is paired with a “negative” cluster that has the closest absolute mean value.

Next, for each positive-edge-negative-edge cluster pair, GSP is used to match each element (i.e., increasing power edge) in the positive cluster with an element (decreasing edge) in the paired negative cluster, by exploiting magnitude differences as well as time

intervals between the edges, like two matching features.

The feature matching starts with the magnitude-wise largest cluster, that is, a cluster that has the largest mean. Let \mathbf{C}^P and \mathbf{C}^N denote two paired clusters, that is, the vectors of increasing and decreasing power edges. The task is, for each $C_i^P \in \mathbf{C}^P$ to find an optimal match among all candidates $C_i^N \in \mathbf{C}^N$. To do this, a graph is formed by considering only edges in \mathbf{C}^N that occur after C_i^P , since the decreasing edge must occur after the increasing edge, and before C_{i+1}^P . This subset of \mathbf{C}^N will be regarded as a set of candidates, denoted by Φ . Let Φ^M represent the set of magnitude differences between C_i^P and each element in Φ . Let Φ^T represent the set time intervals between C_i^P and each element in Φ .

Two graphs are formed:

1. the first graph, $G_M = \{\mathbf{V}^M, \mathbf{A}^M\}$, with nodes indexed by the elements of Φ^M , and

$$A_{i,j}^M = \exp \left\{ -\frac{|\text{dist}(\phi_i^M, \phi_j^M)|^2}{\rho^2} \right\}. \quad (3.6)$$

The graph signal, \mathbf{s}^M , is defined as follows: s_1^M is set to be the average value of the elements in Φ^M , and for $j > 1$, $s_j^M = 0$;

2. the second graph, $G_T = \{\mathbf{V}^T, \mathbf{A}^T\}$, has nodes indexed by the elements of Φ^T , and

$$A_{i,j}^T = \exp \left\{ -\frac{|\text{dist}(\phi_i^T, \phi_j^T)|^2}{\rho^2} \right\}. \quad (3.7)$$

The graph signal, \mathbf{s}^T , is defined as follows: s_1^T is set to be the median value of the elements in Φ^T , instead of the mean value to reduce the influence of the outliers, and for $j > 1$, $s_j^T = 0$;

The graph signal is calculated that minimises the global smoothness of each of the two graphs using Eq. (2.13), obtaining the solutions \mathbf{s}^{M*} and \mathbf{s}^{T*} .

Next an optimisation problem is formulated for feature matching, $i = 1, \dots, n$, as:

$$\arg \max_i \left\{ \alpha s_i^{M*} + \beta s_i^{T*} \right\}, \quad (3.8)$$

where n is the number of candidates, that is, the length of \mathbf{s}^{M*} and \mathbf{s}^{T*} , and α and β

are heuristically chosen, with $(\alpha + \beta = 1)$ to tradeoff magnitude difference and time. The solution of Eq. (3.8) returns the best decreasing edge for each increasing edge C_i^P . Note that in real cases for a particular i if Φ between C_i^P and C_{i+1}^P is empty, C_i^P will be added to the next edge matching iteration. That is, C_i^P and C_{i+1}^P are both positive candidates to be matched, thus sizes of Φ^M , Φ^T and the corresponding graph are naturally enlarged.

After pairing all edges between the matched positive and negative cluster, the rejected (unpaired) edges will be included into the next cluster, i.e., the next magnitude-wise smaller cluster and the above iteration carried out again (see Fig. 3.1).

3.3.4 Disaggregation Output

The above feature matching method forms a pair of matched positive and negative clusters, where each such pair corresponds to one potential appliance state. The matched samples from the two paired clusters define the start and the end of the appliance running event. Each disaggregated event is finally labelled by comparing the disaggregated signature with an existing database of appliance signatures available for that particular household, which can at first be done via crowd-sourcing [119] or a short-time diary, i.e., the signature is extracted at the time-stamp the householder switches on and off the appliances in their house, and after that automatically, for example, as in [27]. If the appliance is not present in the database (there is no match between database signatures and the extracted event using the above feature matching approach), it will be added to the database. If the database is empty, the appliances would be added as they are disaggregated and arbitrarily labelled, until the consumer confirms their label.

As is common practice in NILM [15, 27, 80], each appliance load is estimated using the average appliance power consumption during the identified event time interval, although, some more sophisticated approaches are possible [120].

3.4 Performance Limits

It is expected that the proposed algorithm will work well if the average load of each appliance is distinct enough from other appliance loads in the house, and if P^m , for each m , does not fluctuate much. Next, the influence of the mean and variance of the appliance power load on the disaggregation result is estimated. That is, given statistics of the appliance load, whether the algorithm will be successful in disaggregating these appliances is predicted. Since the proposed approach disaggregates one appliance at a time, to estimate the limits of the proposed approach, without loss of generality, a mixture of two appliance loads is considered.

Each appliance power load is modelled as a two-state Markov source, wherein the on-state Source m (corresponding to Appliance m power load) has Gaussian distribution with mean ω_m and variances σ_m^2 . In the off-state, the source load is zero. It is noted and widely accepted that for the majority of appliances, the power load follows a Gaussian distribution [20, 22]. That is, for $m = 1, 2$, the i -th sample of Source m is defined as:

$$x_{mi} = \omega_m + n_{mi}, \quad (3.9)$$

where n_{mi} denotes zero-mean Gaussian “noise” with standard deviation σ_m that models the fluctuation of the source load from its mean. Then, a mixture of the two time-series signals is generated as:

$$y_i = \mu_i x_{1i} + \nu_i x_{2i}, \quad (3.10)$$

where μ and ν are binary-valued random vectors, which drive transitions between the ON and OFF states. Both μ_1 and ν_1 are initially set to 0, assuming that at the start both appliances are OFF. The remaining values are generated according to the previous state and the following transition probabilities: $P(\mu_{i+1} = 1 | \mu_i = 1) = 0.95$, $P(\mu_{i+1} = 1 | \mu_i = 0) = 0.03$, $P(\nu_{i+1} = 1 | \nu_i = 1) = 0.94$, $P(\nu_{i+1} = 1 | \nu_i = 0) = 0.05$, which are selected to mimic appliance usage. Since most domestic appliances’ operating power is below 3000 Watt (W), both ω_1 and ω_2 are set in this range.

The performance of the algorithm is analysed by looking at five cases. First, $\Delta\omega =$

$\omega_2 - \omega_1$ varies while keeping the variances fixed. Then, the means are fixed and both σ_1 and σ_2 vary. In Case 3, ω_1 and σ_1 are fixed and ω_2 and σ_2 vary. Then, all four parameters vary and the influence of the scaling factor in the Gaussian Kernel function scaling factor, ρ , is investigated. The F – *measure*, F_m , is intuitively utilised as performance measure (defined as Eq. (2.16)).

3.4.1 Case 1

Firstly, ω_1 , σ_1 and σ_2 are fixed and ω_2 varies, to see how $\Delta\omega = \omega_2 - \omega_1$ influences the disaggregation results.

In Fig. 3.2, the results are presented for four different values of ω_1 , 200, 600, 1000 and 1400W. These values represent a range of typical domestic appliances, from low-power appliances such as fridges and freezers that often operate around 200W, mid-power appliances, such as stoves and the dry mode of air conditioners that usually operate at around 600W, to high-power appliances such as toasters, washing mode of dishwashers (around 1000W) and typical electric kettles and microwaves (1400W).

It can be noticed from each subplot in Fig. 3.2 that the performance curve of F_m values always has two clearly distinguishable states: a low F_m -value state, for small $\Delta\omega$, when the performance is poor, and a high F_m -value state for large $\Delta\omega$, labelled in Fig. 3.2d. Thus, as expected, for the fixed source variances, the proposed disaggregation approach always works well when $\Delta\omega$ is large enough. A sharp increasing edge provides a clear performance limit when the algorithm accuracy is close to 100% or $F_m = 1$. This result is expected, since the larger the difference in appliance loads making up the aggregate, the better the disaggregation.

3.4.2 Case 2

Next, ω_1 and ω_2 are set to 600W and 800W, respectively, thus, $\Delta\omega$ is fixed as 200W. Both σ_1 and σ_2 are kept changing to find how the variance term affects the performance. The noise variance is equivalent to the power load fluctuation of an appliance, that is, the power deviation from the mean value.

Fig. 3.3 presents the F_m performance versus σ_1 and σ_2 and the corresponding fitted

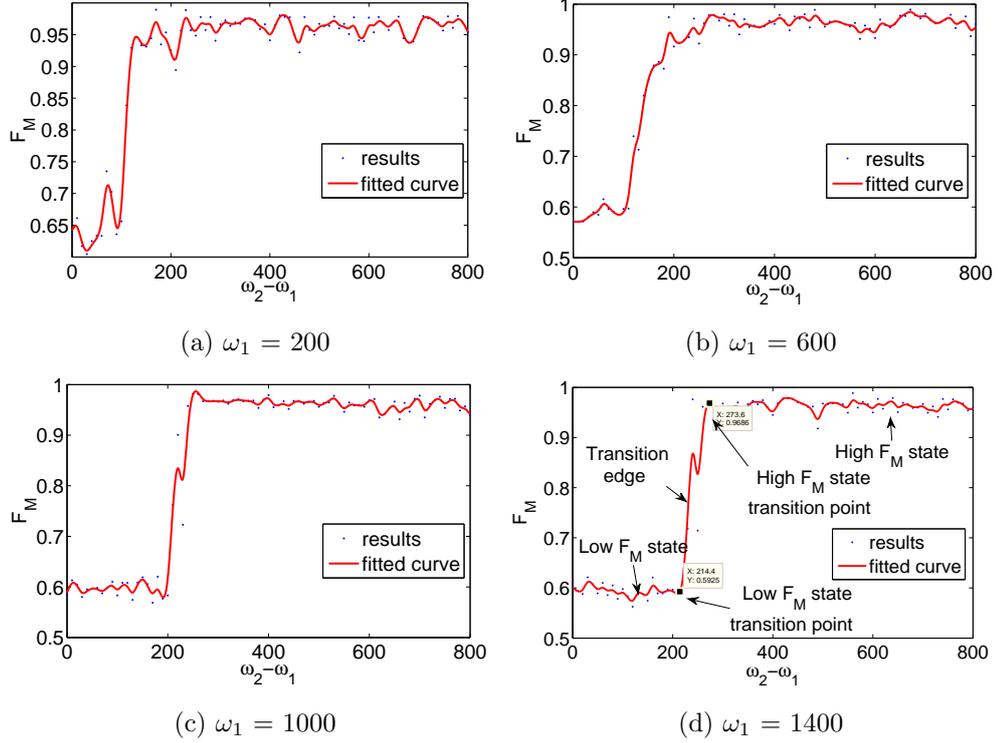


Figure 3.2: F_m versus $\Delta\omega$ for different ω_1 . Both σ_1 and σ_2 are fixed to 20.

surface which is a quartic polynomial consisting of 14 coefficients. Assuming that $F_m > 0.8$ provides acceptable performance, Fig. 3.3c shows the $[\sigma_1, \sigma_2]$ plane where F_m is greater than 0.8, which can be fitted as a quadrant defined by:

$$\sigma_1^2 + \sigma_2^2 \leq r^2, \quad (3.11)$$

where $\sigma_1 \geq 0, \sigma_2 \geq 0$ and $r \approx 52$ in this example. For different ω_1 and ω_2 , heuristically, it is found that the radius of the fitted quadrant will change, but Eq. (3.11) will still hold. Thus, there is a clear limit in the intensity of variation that the proposed algorithm can tolerate, and this depends on the mean values of power loads.

3.4.3 Case 3

In this step, ω_1 is fixed to 600W, σ_1 is fixed to 20 and ω_2 and σ_2 vary to investigate how the performance is affected with respect to $\Delta\omega$ and $\frac{\sigma_2}{\sigma_1}$. Since it can be observed

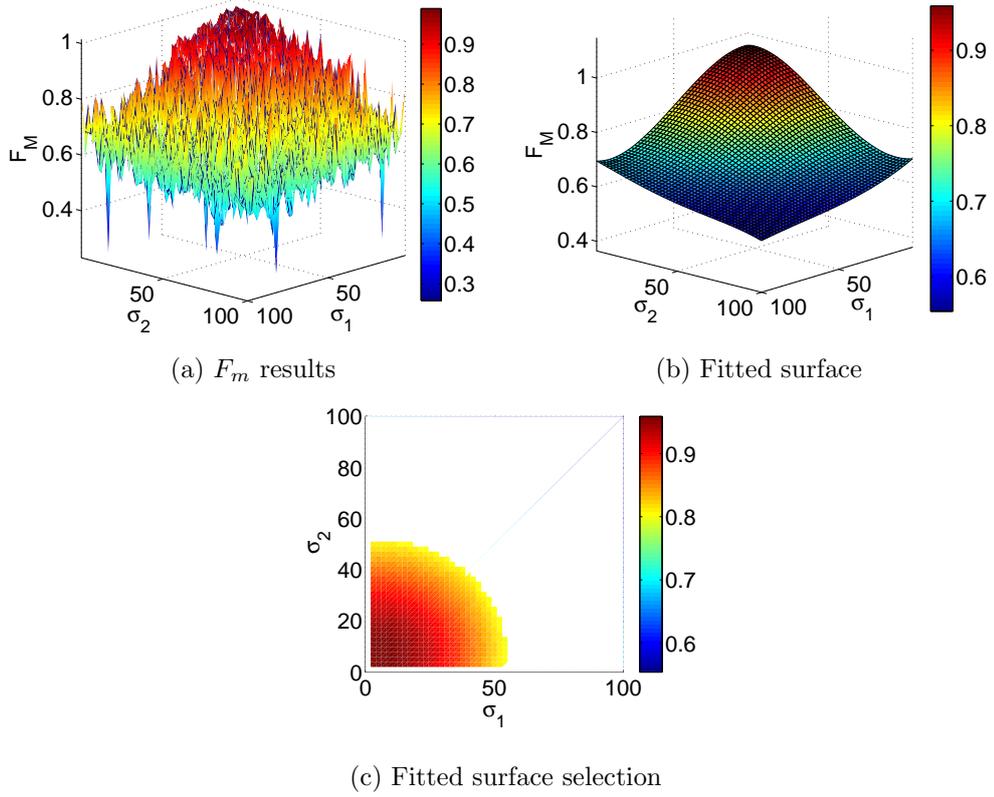


Figure 3.3: F_m versus σ 's where $\Delta\omega = 200$.

from Fig. 3.2 that there exists a value of $\Delta\omega$ when F_m sharply increases, the two planes are searched for obtained by varying the ratio $\frac{\sigma_2}{\sigma_1}$.

Both high F_m “plane” and low F_m “plane” are clearly shown in Fig. 3.4. When $\frac{\sigma_2}{\sigma_1}$ is less than 2, the separation edge between high and low F_m “plains” is steep. It is more moderate when $\frac{\sigma_2}{\sigma_1}$ becomes larger, and the edge almost disappears when $\frac{\sigma_2}{\sigma_1}$ is around 5. So, if the variance ratio is too high, the algorithm will not perform well, which is expected since one (“noisier”) source will significantly affect disaggregation of the other.

3.4.4 Case 4

Next, the performance of the proposed algorithm is investigated as a function of the scaling factor, ρ , in the Gaussian kernel weighting function. To capture the changes in

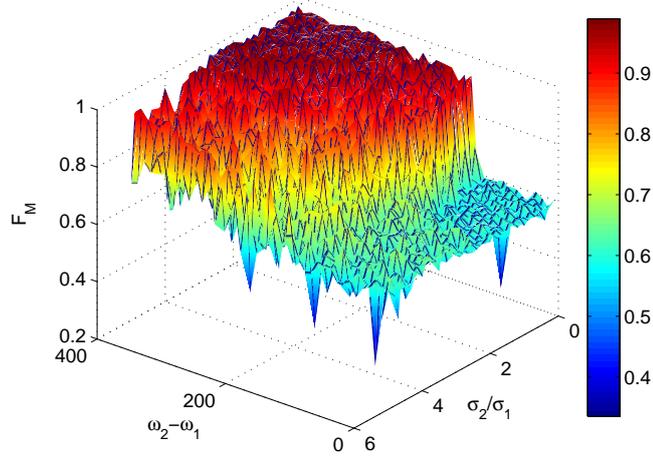


Figure 3.4: F_m versus $\Delta\omega = \omega_2 - \omega_1$ and $\frac{\sigma_2}{\sigma_1}$.

all four parameters, the sensitivity index d' is used and defined by:

$$d' = \frac{|\omega_2 - \omega_1|}{\sqrt{\frac{1}{2}(\sigma_1^2 + \sigma_2^2)}}. \quad (3.12)$$

It can be seen that the F_m performance versus d' in Fig. 3.5 for four different initial ρ in the Gaussian kernel weighting function.

Fig. 3.5 clearly shows that, as expected, the greater the d' , the better the performance. In the figure, the blue points represent the cases of $F_m \geq 0.8$. Note that the proposed approach always performs poorly when d' is less than 3.

It seems that ρ does not affect the results. The iterative nature of the proposed clustering approach reduces the effect of ρ . One can see from Table 3.2, that by increasing ρ , the number of clusters before and after pairing change, but the F_m values stay similar.

Table 3.2: Performance of the proposed approach and the number of clusters for different initial ρ .

ρ	5	25	45	65	85	105
F_m	0.98	0.98	0.98	0.97	0.97	0.98
No. clust. before pairing	17	8	6	7	7	5
No. clust. after pairing	6	6	4	6	6	4

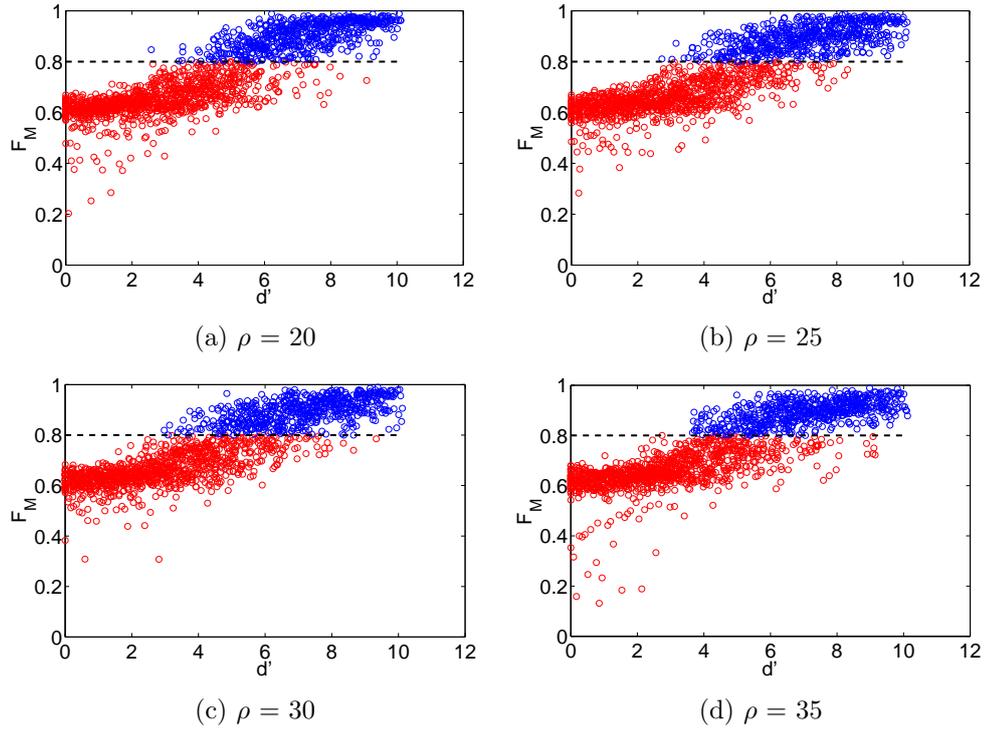


Figure 3.5: F_m versus $\Delta\omega$ for different ω_1 .

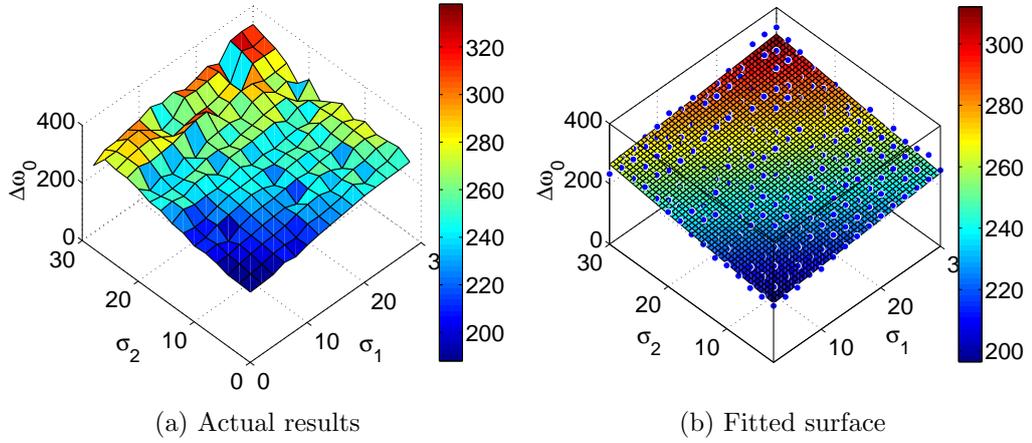


Figure 3.6: The limits of the proposed algorithm. All values on and above the surface lead to $F_m \geq 0.8$.

3.4.5 Case 5

Finally, the case study focuses on the distribution of high F_m transition state points, when F_m becomes acceptably high, namely, the limit of the performance of the proposed

approach. ω_1 is fixed to 1000W, σ_1 and σ_2 are kept changing in the range from 2 to 30, with steps of 2. Fig. 3.6 shows the high F_m state transition points, that is, the minimum $\Delta\omega$ required for good disaggregation performance, denoted by $\Delta\omega_0$.

In Fig. 3.6a, the value of the bound $\Delta\omega_0$ generally increases when either σ_1 or σ_2 increases. Since the definition of sensitivity index in Eq. (3.12) indicates ω 's and σ 's have the same dimension, the boundary is fitted as a three-variable linear equation which represent a plane in Fig. 3.6b as:

$$\Delta\omega_0 = a\sigma_1 + b\sigma_2 + c \quad (3.13)$$

where a is 1.67, b is 2.46 and c is 188. This equation will be used in the next section as a performance limit to predict disaggregation performance of the proposed algorithm.

3.4.6 Summary

From the above analysis the following conclusions can be made:

- there exists a clear $\Delta\omega$ limit between good algorithm performance and poor performance;
- the fluctuation of an appliance power load influences algorithm performance;
- when the ratio between the variances of the appliance load is high, the performance is poor;
- the performance limits can be expressed adequately using sensitivity measure d' ;
- for given variances of appliance loads, the minimum $\Delta\omega$ required for good performance can be estimated using Eq. (3.13).

Consequently, with the above-derived limits of the proposed algorithm, its performance can be predicted as well as both the success and failure of the approach can be explained given appliance load statistics.

3.5 Case Study

In this section, how the proposed algorithm works in practice is clarified, from the generation of the graph signal to final feature matching. As input, the aggregate active power readings of House 2 from REDD dataset [60] is used collected from April 28th 2011 to April 30th 2011, downsampled to 1 minute, as a typical example that does not have any gap. The aggregate power readings are shown in Fig. 3.11 (red curve).

In order to minimise the effects of load fluctuations without losing the state transition edges of all appliances, first, the events (candidate edges) are extracted by thresholding (Eq. (3.3)) on the aggregate load with $T_0 = 10W$. A segment of the resulting events is shown in Fig. 3.7 (top) as aggregate power versus time.

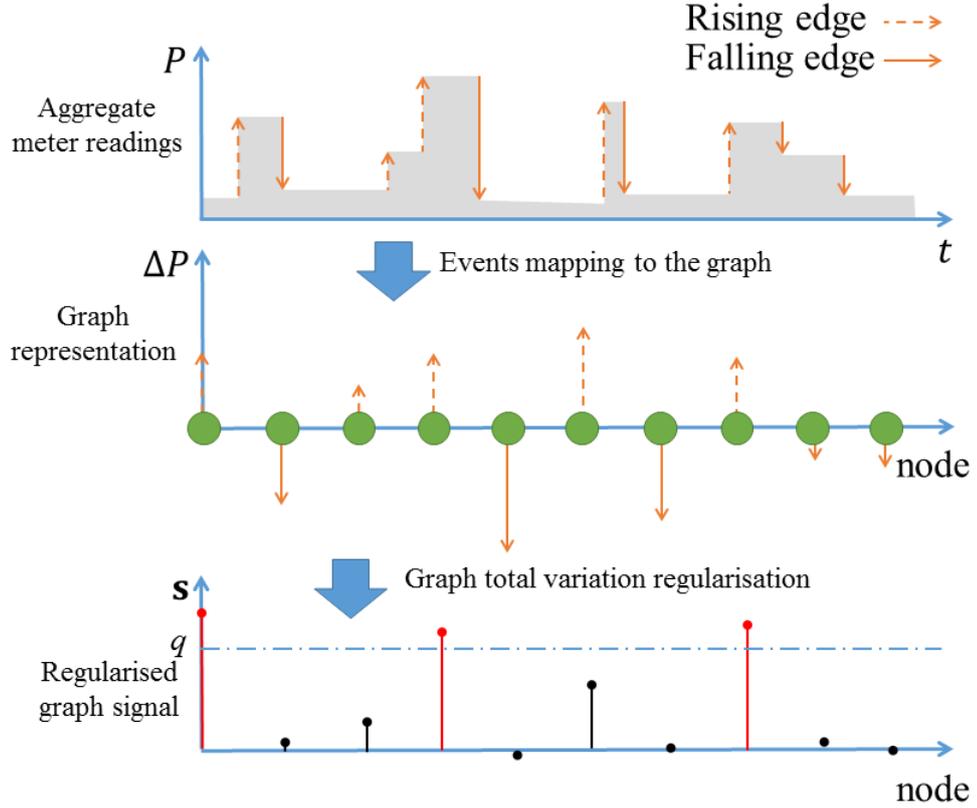


Figure 3.7: Graph generation for Stage 1 of clustering.

Graph Generation. All rising and falling edges are indexed by the nodes of

a graph (see the middle figure in Fig. 3.7). Each power edge ΔP_i in Fig. 3.7 (top) corresponds to one node v_i in the graph and the edge between nodes v_i and v_j is assigned using Eq. (2.5) with $x_i = \Delta P_i$ and $x_j = \Delta P_j$ and Euclidean distance measure.

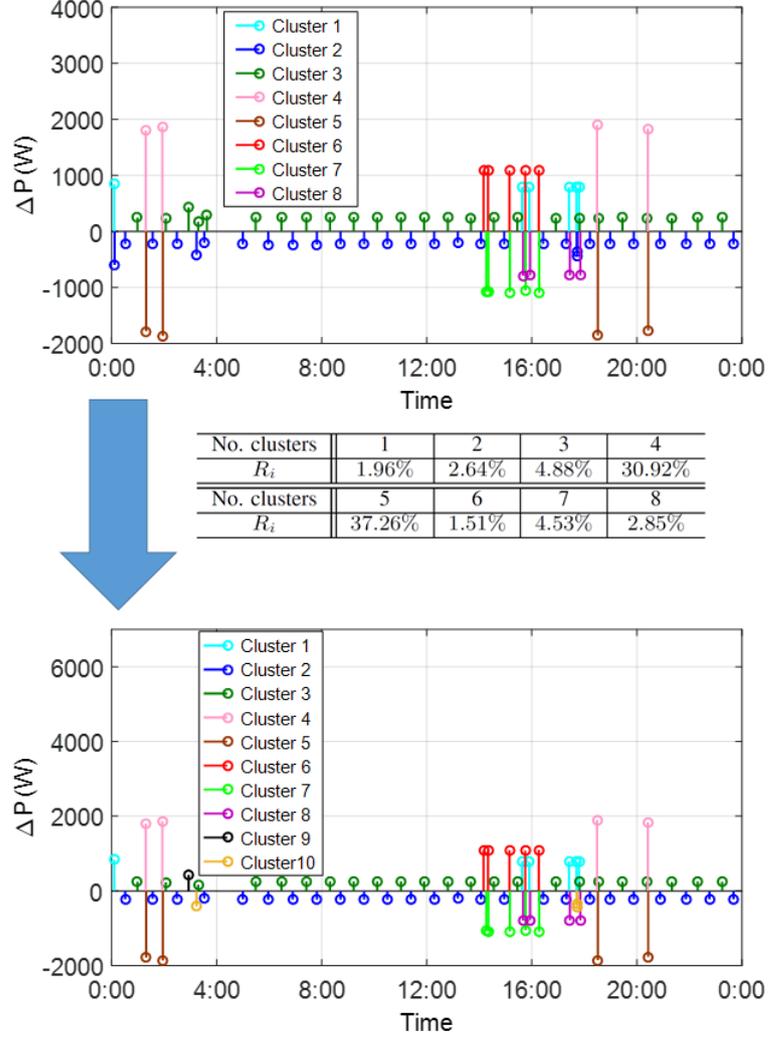


Figure 3.8: Clustering results.

Initial Clustering. The task is to group all edges similar to the first edge into the same cluster. To do that, a graph signal \mathbf{s} is formed where each sample of \mathbf{s} corresponds to one node in the graph. s_1 is set to 1 and $\mathbf{s}_{2:n}^*$ are calculated by Eq. (2.10) to ensure that the graph signal remains smooth. This step is intuitive since smooth changes between the edges are expected to be within the same cluster. Finally, all candidate

edges which are magnitude-wise similar to the first edge are grouped by thresholding of $s_{2:n}^*$ using a high threshold $q = 0.98$ (i.e., Edge i , $i > 1$, will be grouped in the cluster represented by the first edge only if $\frac{s_i}{s_1} \geq 0.98$), displayed as red edges in Fig. 3.7 (bottom). Note that time duration information is not a feature in this step. The same clustering procedure is repeated for the unclustered events until all events are clustered. As a result, in this example, eight “initial” clusters are formed shown in Fig. 3.8 (left).

Refined Clustering and Adaptive Thresholding. For each cluster, *RSD* R_i is used to denote the *quality* of the cluster defined in Eq. (3.4). The clusters with $R_i > K$, where $K = 10\%$, will be sub-clustered to make sure that each cluster has a high precision degree and is likely to contain the events of only one power state (Fig. 3.8 (middle)). For example, a cluster with a mean magnitude value of 150W should have a standard deviation of at most 15W to be an acceptable cluster. In this example, Clusters No.4 and No.5 need sub-clustering by halving ρ in Gaussian kernel weighting function for enhancing the clustering degree. At the end of clustering, small-sized clusters are merged into larger magnitude-wise clusters to keep a balance between the number of positive and negative clusters. After sub-clustering and merging, 10 clusters are obtained (see Fig. 3.8 (right)).

Feature Matching. In the next step, each positive cluster is matched to the magnitude-wise closest negative cluster. Then for each positive-negative cluster pair, each rising edge is paired to an optimal falling edge using feature matching, as explained in Subsection 3.3.3. This is demonstrated in Fig. 3.9, where red and blue impulses show positive and negative edges, respectively, and are fridge state transitions. The task is to find a falling edge corresponding to each rising edge. In this example, for one particular rising edge, two possible falling edge candidates are identified based on their time distance to the rising edge, identified within the dashed-line rectangles.

Fig. 3.10 demonstrates how pairing is performed using GSP, where both power and time information are treated as features. There exist two candidate falling edges at distances Δt_1 and Δt_2 , from the rising edge (see Fig. 3.10 (top)). Two graphs are built: (1) one to capture the similarity between absolute power values of the rising edge and candidate falling edges (the left-hand side on the second figure from the top

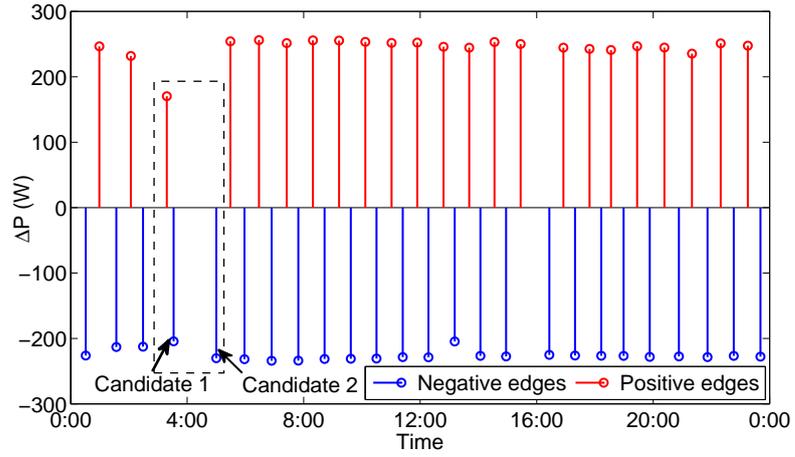


Figure 3.9: A typical positive-negative cluster pair.

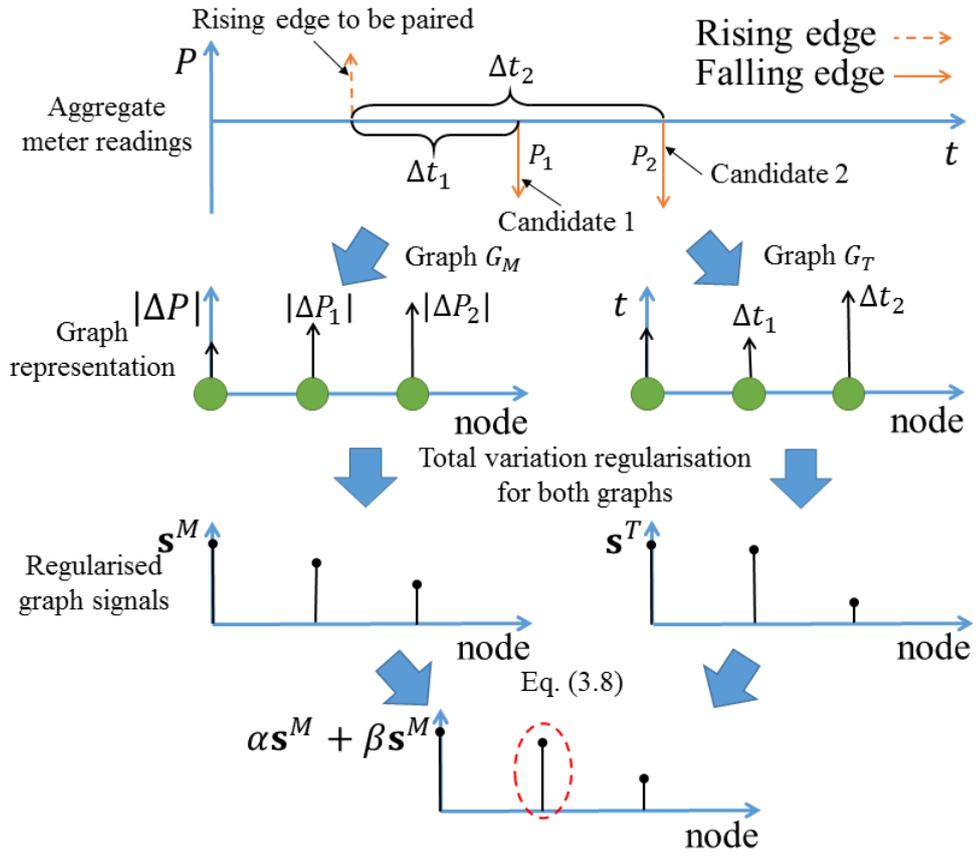


Figure 3.10: Graph example for feature matching.

in Fig. 3.10); (2) the second graph captures time duration between the rising edge and the candidate falling edges (the right-hand side on the second figure from the top in Fig. 3.10).

In the absolute power value graph, the first node's value is set to the difference between the mean of absolute values of all rising edges and the mean of the absolute values of all falling edges. The remaining nodes index the differences in absolute power value between the rising edge and the candidate falling edges. In the time interval graph, the first node is set as the median of the set containing the time interval between each rising edge and the following falling edge. The remaining nodes index the time duration between the rising edge and the falling edge candidates. The graph global smoothness minimisation solutions \mathbf{s}^{M*} and \mathbf{s}^{T*} for both power and time interval information are calculated using Eq. (3.2) and adjacency matrices \mathbf{A}^M and \mathbf{A}^T are obtained through Eqs. (3.6) and (3.7), respectively. See the third row in Fig. 3.10, where the left-hand side shows the power graph signal and the right-hand side shows the time interval graph signal, respectively. In the last step, the smoothness minimisation solution of both graphs is weighted and then summed using Eq. (3.8), where the maximum value corresponds to the optimal event pair. The optimal falling edge corresponding to the rising edge is circled in Fig. 3.10 (bottom).

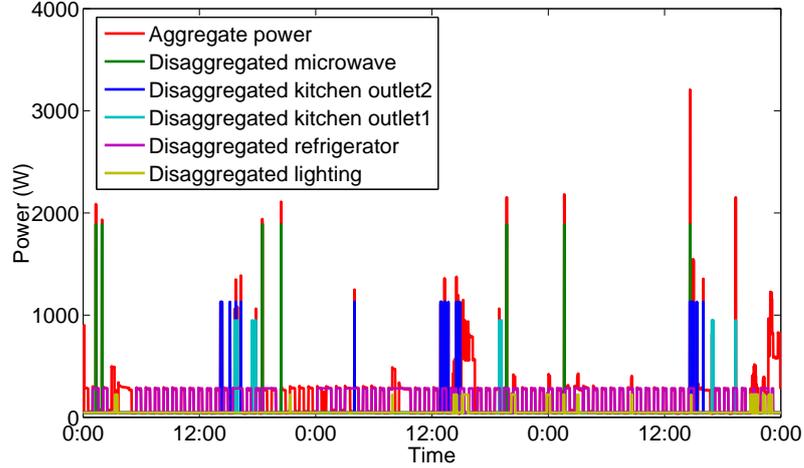


Figure 3.11: Disaggregation result for three typical days.

The disaggregation result for the three typical days illustrated by this example is

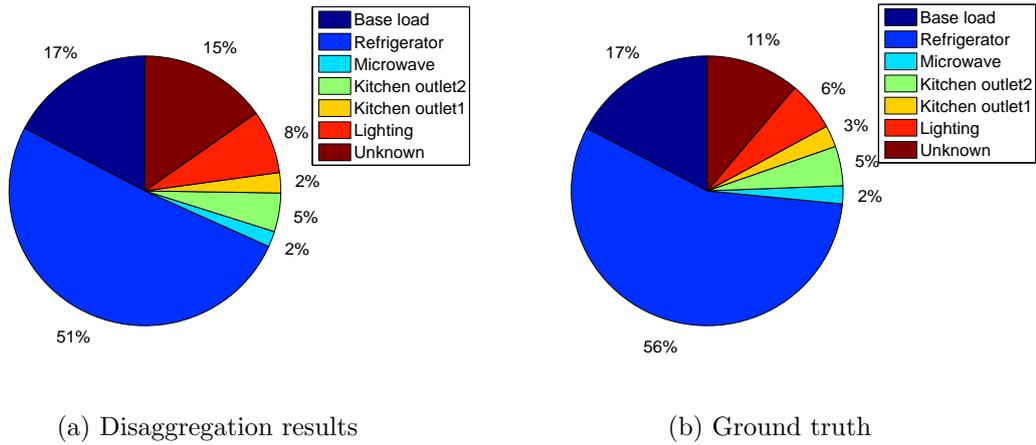


Figure 3.12: Pie charts of NILM results for House 2 from REDD dataset for three typical days.

shown in Fig. 3.11. Note that there is a good alignment between the true aggregate values and the disaggregated result. Fig. 3.12 presents the NILM accuracy for the three typical days. baseload is defined as the minimum power value over the three monitored days. The disaggregation error between the estimated (12.59kWh) and the real energy consumption (13.2kWh) is 4.6%. Unknown in the left pie chart denotes the difference between the true aggregate consumption and the sum of the disaggregated load and indicates the load that cannot be disaggregated. Unknown percentage in the right, ground truth, the pie chart is the difference between the measured aggregate consumption and the sum of loads of the individual, sub-metered appliances.

3.6 Results and Discussion

The performance of the proposed algorithm is demonstrated using active power readings from two open-access datasets: (1) the REDD dataset [60] of US houses, downsampled to 1 minute; (2) the UK REFIT dataset [61], with measurements collected every 6-8 seconds. Both datasets are supported by NILMTK [18] - a toolkit designed to help researchers evaluate the accuracy of NILM algorithms.

The REDD dataset is widely used for the evaluation of various NILM approaches [16, 25, 58, 59, 72]. REDD houses contain several appliances with only a few unknown

loads. REFIT dataset is chosen for simulation to evaluate various energy disaggregation/prediction solutions in recent works, such as [16, 24, 121], which is compatibility with NILMTK. Compared with other public datasets (e.g., REDD, AMPds), the advantages of REFIT dataset are obvious. Firstly, REFIT dataset contains 20 houses, providing more scenarios of in various levels of noisy setting. Secondly, for each house, validate period of measurements longer than one year is collected. It satisfies validation demands, as the data volume will reduce when generating hourly load profile. The last but not the least, REFIT dataset is generally more challenging and closer to the real case due to numerous unknown appliances and some measurement noise [61]. The results are shown only for appliances for which time-diaries are collected or whose usage was measured at the appliance level via appliance monitors; these appliances are referred to as known appliances.

K is empirically set for evaluating cluster *quality*, i.e., $RSD R_i$, to 10% to minimise $|R_i - K|$ for all appliances. Initial threshold T_0 is set to 10W to avoid detecting standby states without filtering any appliances' state-transition edge. The initial Gaussian kernel weighting function scaling factor ρ is set to 20 to avoid over-clustering for most domestic appliances, which work below 3000W. $\alpha = \beta = 0.5$ as power and time have the same influence when pairing rising and falling edges, and q is set to 0.98 to group only highly correlated samples, and reduce falsely clustered edges.

The evaluation metrics used in the experiments are PR , RE , F_m as in [20], all defined in Subsection 2.5.1. For comparison with recent approaches [25, 58], $Acc.$ (defined by Eq. (2.18)) is also used to demonstrate disaggregation accuracy.

3.6.1 Algorithm Performance

In order to enable like-for-like comparison with [22, 27, 40, 58, 72], REDD Houses 1, 2 and 6 are used for validation. House 2 contains the fewest power states among all three houses and has a low appliance complexity and low time complexity as defined in [122], implying that House 2 is relatively the easiest house to disaggregate. House 1 has a similar number of appliances as House 2, but more power states [122], and consequently is more challenging to disaggregate. House 6 has the largest number of

appliances of the three. Approximately, two weeks worth of consecutive data is used in this experiment. Note that all data is used for testing - there is no training in the proposed method.

Table 3.3: Performance of the proposed approach for House 1 from the REDD dataset.

Appliance	ATP	ITP	FP	FN	PR	RE	F_m
Microwave	174	105	21	13	0.89	0.6	0.72
Kitchen outlet	19	14	1	13	0.95	0.41	0.57
Washer dryer	191	31	11	78	0.95	0.64	0.76
Oven	29	4	2	26	0.94	0.49	0.64
Lighting	35	26	16	32	0.69	0.38	0.49
Refrigerator	513	50	20	127	0.96	0.74	0.84
Dishwasher	49	139	3	15	0.94	0.24	0.38
Bathroom GFI	39	11	9	10	0.81	0.65	0.72

Table 3.3 shows the performance for REDD House 1. Relatively low F_m results for dishwasher can be attributed to the similarity between the refrigerator load and one power state of the dishwasher. Indeed, the average working power of refrigerator is 201W and that of the low power state of the dishwasher is 210.6W, hence the two power states are likely to be grouped into the same cluster, as shown by the large ITP , which captures detected events that are wrongly labelled as refrigerator.

Table 3.4: Performance of the proposed approach for House 2 from the REDD dataset.

Appliance	ATP	ITP	FP	FN	PR	RE	F_m
Microwave	80	5	0	5	1	0.89	0.94
Kitchen outlet1	39	0	1	13	0.98	0.75	0.85
Kitchen outlet2	172	9	11	18	0.94	0.86	0.85
Stove	34	0	83	4	0.29	0.89	0.44
Refrigerator	595	17	110	161	0.84	0.77	0.8
Dishwasher	19	15	11	13	0.63	0.4	0.49
Lighting	70	0	110	9	0.64	0.89	0.74

The results for House 2 and 6 are shown in Tables 3.4 and 3.5, respectively. From Table 3.4, it can be observed that most appliances are disaggregated with accuracy $> 70\%$. The worst performance is achieved for the stove, due to a high FP . The poor performance for the heater and lighting in House 6, shown in Table 3.5, is due to the questionable ground truth data collected using sub-metering for these two appliances

Table 3.5: Performance of the proposed approach for House 6 from the REDD dataset.

Appliance	ATP	ITP	FP	FN	PR	RE	F_m
Microwave	10	0	3	0	0.77	1	0.87
Kitchen outlets	4	1	3	3	0.57	0.5	0.53
Stove	7	5	3	2	0.7	0.5	0.58
Refrigerator	439	8	56	132	0.89	0.76	0.82
Electronics	26	6	61	5	0.3	0.7	0.42
Heater	3	0	56	3	0.05	0.5	0.09
Air conditioner	44	9	0	1	1	0.81	0.9
Lighting	7	6	7	12	0.5	0.28	0.36
Outlets unknown	146	6	56	65	0.72	0.67	0.69

(sub-metering data is very noisy with many spikes).

Table 3.6: Performance of the proposed approach for House 8 from the REFIT dataset.

Appliance	ATP	ITP	FP	FN	PR	RE	F_m
Microwave	7	10	0	3	1	0.35	0.52
Toaster	4	1	2	1	0.67	0.67	0.67
Kettle	39	7	6	2	0.87	0.81	0.84
Refrigerator	18	0	2	0	0.9	1	0.95
Freezer	54	16	180	24	0.23	0.57	0.32
TV	4	0	180	6	0.02	0.4	0.04
PC	0	0	0	4	NaN	0	0
Washing machine	3	1	8	0	0.27	0.75	0.4

Next, the results for REFIT House 8 are presented in Table 3.6, disaggregated over three consecutive days when all seven known appliances were running. It can be seen that the proposed training-less approach showed very good performance for the kettle and refrigerator, but poor results for the TV, caused by TV being grouped in the same cluster as the freezer because TV and freezer have close active power range. Note that, the proposed approach shows average performance (across all appliances) of $F_m = 0.49$, which is better than the unsupervised HMM-based approach in [27], which uses training and shows $F_m = 0.46$ for the same house.

The disaggregation results are also demonstrated for REFIT House 9 over a period of four consecutive days in Table 3.7. The proposed method scores over 85% F_m for dishwasher, washer dryer and kettle, where such identified high-consuming loads are approximately: dishwasher (high states) 2200 Watt; washer dryer (high states) 2500

Table 3.7: Performance of the proposed approach for House 9 from the REFIT dataset.

Appliance	ATP	ITP	FP	FN	PR	RE	F_m
Microwave	6	3	19	2	0.24	0.55	0.33
Dishwasher	17	1	2	1	0.89	0.89	0.89
Washer dryer	60	12	2	7	0.97	0.76	0.85
Refrigerator	168	33	307	39	0.35	0.7	0.47
TV	4	1	25	7	0.14	0.33	0.2
Kettle	52	9	2	0	0.96	0.85	0.9

Watt; and kettle 2800 Watt. The results are inline with the case study in Subsection 3.4.1, that is, the proposed approach achieves good performance when $\Delta\omega$ is large enough. However, the fluctuations during low working states of dishwasher and washer dryer are magnitude-wise close to working states of refrigerator and TV, resulting in high FP and poor classification results for both refrigerator and TV. The other reason for poor refrigerator performance is claimed based on the power measurement characteristics.

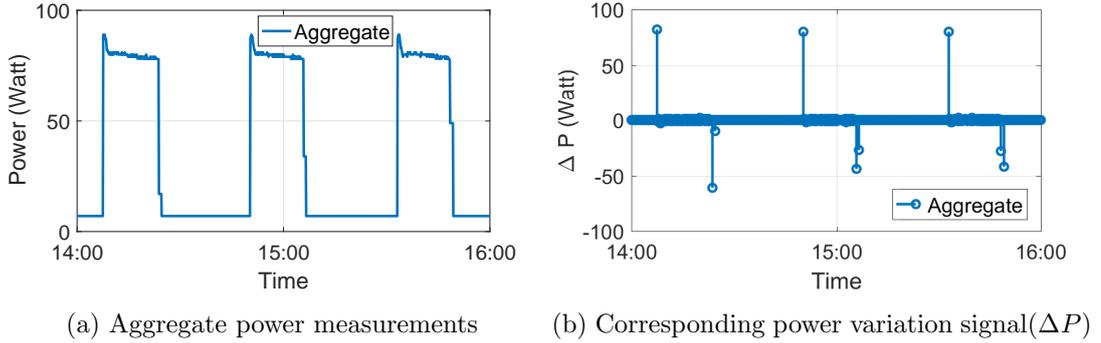


Figure 3.13: Example of aggregate readings for REFIT House 9 on 25th Feb. 2015

An example of aggregate power readings referring to refrigerator’s loads is demonstrated as Fig. 3.13a, where OFF events last more than one sample in time-series power signal. Thus, in edge detection, multiple consecutive segmented falling edges are captured instead of ideal, *sharp*, complete falling edges as shown in Fig. 3.13b. It leads to more FP events in both clustering and feature matching, and eventually, the clusters containing such FP events are labelled as refrigerator. The high FP and low F_m for microwave, is due to unknown loads with similar power range, not belonging to

any individual, sub-metered appliance category.

The algorithms were implemented in Matlab2014 and were executed on Intel Core i7-4700MQ CPU 2.40GHz machine running Windows 8.1. It takes 65.6 seconds to complete data filtering and clustering and 3.5 seconds to finish feature matching, i.e., 69.1 seconds in total, for processing 20,160 samples, which means roughly 3 milliseconds per sample. Note that optimised implementations would require less time.

3.6.2 Comparison with State-of-the-Art

In this section, the results are compared with some state-of-the-art NILM algorithms, proposed for low sampling rates and active power measurements.

First, the results, F_{m_U} , are compared to those of the supervised GSP-based approach in [40], F_{m_S} , and the results using the unsupervised HMM-based method of [22], F_{m_H} , as reported in [40]. All three methods are tested using the same data, while the two benchmark methods require additional data for training [22, 40].

Table 3.8: Performance of three NILM approaches for Houses 2 and 6 from the REDD dataset.

Appliance	House2			House6		
	F_{m_U}	F_{m_S}	F_{m_H}	F_{m_U}	F_{m_S}	F_{m_H}
Microwave	0.94	0.26	0.47	0.87	0.92	0
Kitchen outlet2	0.85	0.59	0.68	0.53	1	0
Stove	0.44	0.41	0.21	0.58	1	0
Refrigerator	0.8	0.63	0.9	0.82	0.54	0.88
Dishwasher	0.49	0.56	0.04	0.42	-	-
Heater	-	-	-	0.09	0.11	0.03
Air Conditioner	-	-	-	0.9	0.49	0.12

The results for the three NILM approaches are shown in Tables 3.8 for the REDD datasets. Overall, both GSP-based approaches perform significantly better than the HMM-based approach. The proposed unsupervised GSP-based approach performs, on average, as well as the supervised one of [40], but without the training and supervised labelling overhead.

The performance of the proposed method for the REDD houses is comparable to the three FHMM-based unsupervised approaches of [58] and two FMM-based approaches

Table 3.9: Performance comparison with [58] and [25] for House 2 from the REDD dataset. Note that only 5 top-consuming appliances are disaggregated in [58], while the proposed methods disaggregated seven appliances.

Approach	Acc.
Proposed method	77.2%
EM FHMM [58]	50.8%
F-HDP-HMM [58]	70.7%
F-HDP-HSMM [58]	84.8%
FHMM (without interaction) [25]	65.8%
FHMM (with interaction) [25]	66.5%

of [25] that all require training. Indeed, from Table 3.9 that shows the accuracy *Acc.* for House 2 from REDD dataset, It can be observed that the proposed method is more accurate than Expectation Maximization FHMM (EM-FHMM), Factorial Hierarchical Dirichlet Process (F-HDP) HMM as well as both FHMM approaches with and without interaction and performs slightly worse than F-HDP Hidden Semi-Markov Model (F-HDP-HSMM) approach of [58]. Note, however, that in [58], only top five power-drawing appliances are disaggregated, while both the proposed algorithm and that in [25] disaggregate seven appliances in House 2.

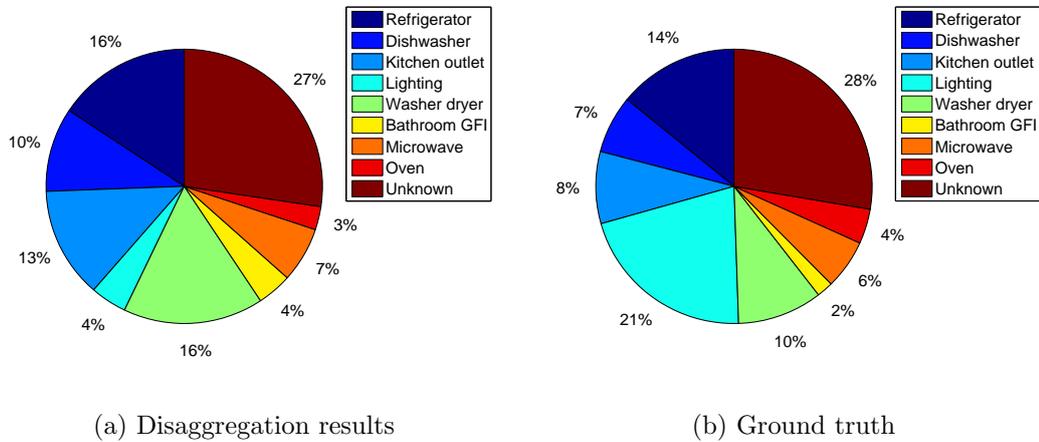


Figure 3.14: Pie charts of NILM results for House 1 from REDD datasets.

Next, the performance of the proposed approach is demonstrated for House 1 from REDD dataset with ground truth shown in Fig. 3.14b for comparison with a recent unsupervised approach in [72]. In contrast to [72], which uses the aggregated load of

six measured individual appliances, the proposed algorithm is applied to the measured aggregate data from the REDD dataset, downsampled to 1 minute, to demonstrate robustness to noise. Disaggregation is done for a period of 17 consecutive days. The resulting estimation error between the estimated (58.32kWh) and the real energy consumption (60.16kWh) is 3%, in the presence of noise, as opposed to the reported 2% in [72]. Additionally, eight appliances are identified in contrast to three ‘virtual’ appliances (i.e., appliances with similar power demand are grouped) in [72].

3.6.3 Performance Bound Analysis

Next, Eq. (3.13) is used to predict the performance of the proposed algorithm. The usefulness of the analytical study and the bound (Eq. (3.13)) are demonstrated for two appliances from the REFIT House 8: kettle, where disaggregation was successful, and TV where the algorithm did not work (see Table 3.6). First, the power load of each appliance is modelled using the Gaussian probability density function (PDF) (see Fig. 3.15), as is common practice [20, 22].

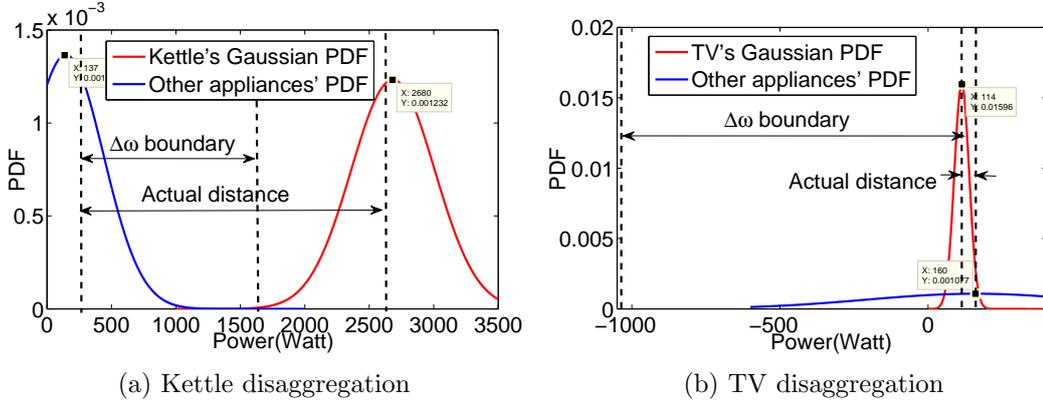


Figure 3.15: REFIT House 8 kettle and TV disaggregation.

The kettle’s standard deviation is 323.8W, while the standard deviation of the total aggregate signal without kettle is 292.2W and the actual difference in means of the two Gaussian distributions is found to be $\Delta\omega$ is 2537.1W which is greater than the $\Delta\omega$ boundary as shown in Fig. 3.15a. According to Eq. (3.13) the performance limit for $\Delta\omega_0$ is much lower - 1472.5W indicating correctly that disaggregation will work. From Table 3.6, the F_m value of kettle disaggregation is $0.84 > 0.8$, which confirms the

estimate.

Fig. 3.15b represents TV disaggregation. As standard deviation of TV and the remaining load, σ_1 and σ_2 , are equal to 25W and 370.5W, respectively, the $\Delta\omega_0$ limit given by Eq. (3.13) is 1141.2W. As, the actual distance $\Delta\omega$, in this case, is 46.8W, which is much smaller than the limit, the performance is predicted to be bad. Since the F_m value of TV in House 8 in Table 3.6 is 0.04, the poor performance confirms the prediction.

3.7 Summary

This chapter builds on the emerging GSP concepts to develop a novel, *blind*, unsupervised low-rate NILM approach. The main motivation comes from the fact that GSP does not require training, can accurately capture signal patterns that occur rarely, is robust to noisy data, and has low implementation complexity.

Based on the results from disaggregating aggregate loads measured from 4 real houses, the proposed training-less GSP-based NILM approach shows comparable performance with the supervised GSP-based NILM approach of [40] outperforming the unsupervised HMM-based method. The performance limits of the proposed algorithm are heuristically determined and the usefulness of this limit to estimate the disaggregation performance is demonstrated. Due to its low complexity, simple operation and minimal customer input (for initial labelling), the proposed algorithm can be applied on large scale as an embedded system as part of Consumer Access Device [49] with an online feedback interface that users can access.

The challenges of using GSP for tackling the NILM problem are: i) the previous GSP-based NILM approach which motivated this paper [40] was supervised and require a training dataset which may be unavailable in real-world applications; ii) the proposed GSP relies on graph total variation regularization, thus the disaggregation task is defined as an optimization problem containing both fidelity and smoothness terms, which is hard to solve directly; and iii) a too-large window size results in long execution time, reported in [100]. However, a small window size may affect NILM performance. Therefore, the GSP-based NILM approach is proposed in this Chapter to addresses

the challenges in the following ways. First, the proposed approach is training-less and requires only aggregate measurements. Secondly, a closed-form solution of graph total variation regularization can be calculated, as the optimisation is only applied to the smoothness term. The sliding window size is heuristically selected base on the data granularity.

After a study of the current algorithm and related results, improving event detection via pre-processing (denoising and filtering) is intuitively regarded as an effective way to achieve better performance. Additionally, in the current algorithm, if a single state transition lasts longer than the sampling period multiple consecutive events will be identified. Motivated by these findings, pre-processing and edge sharpening are proposed to reduce false detection in Chapter 4, where the unsupervised GSP-based algorithm is applied to more scenarios.

This chapter has further demonstrated the potential of GSP for load disaggregation where data granularity is less than 1 minute. Furthermore, the proposed algorithm is utilised for more challenging hourly profile disaggregation in Chapter 5, where the algorithm robustness is enhanced when dealing with appliance simultaneous operation by building an extra graph for the time-of-day as a feature.

Chapter 4

Event-based Non-intrusive Load Monitoring Improvement using Graph Signal Processing¹

Notations

K	Acceptable precision of a cluster
L_E	Window size for edge sharpening
L_G	Window size for graph filtering
L_M	Window size for median filtering
T_E	Threshold for edge sharpening
T_G	Threshold for bilateral graph filter
$\Delta\mathbf{P}$	Differential power signal
α	Trade-off factor between fidelity and smoothness terms
\mathbf{A}	$N \times N$ adjacency matrix
\mathbf{D}	$N \times N$ degree matrix
\mathbf{L}	$N \times N$ degree matrix
\mathbf{P}	Aggregate active power signal
Ω	Window of samples
\mathbf{s}	$N \times 1$ graph signal

¹This chapter is mainly based on the work that appeared in IEEE Access [123]

- \mathcal{M} A set of known appliances in the house
- \mathcal{V} A set of N vertices
- ρ Scaling factor for Gaussian kernel weighting
- q Threshold for acceptable graph signal estimation

4.1 Introduction

Motivated by the conclusion of the last chapter that data cleansing is required to help NILM approaches achieving better detection and disaggregation performance, in this chapter graph signal processing (GSP) is proposed as a tool that brings together low-level *signal* processing and application-driven *data* processing in order to improve the performance of various event-based NILM approaches, suitable for diverse electrical load datasets.

GSP-based approaches have recently been proposed for tackling the NILM problem, via supervised [16, 40], and unsupervised approaches proposed in Chapter 3. However, this prior work applied GSP at the data processing stage only, i.e., as a robust classification or clustering tool, without exploiting GSP's properties as effective physical signal filters [51], which can combat NILM sensitivity to measurement noise and the influence of unknown appliances. It is well recognized [15, 16, 50], that without appropriate processing of the physical measured signal, NILM will often not be accurate or successful, regardless of the effectiveness of the employed classification method. E.g., the performance of the unsupervised GSP-based approach proposed in the last chapter is significantly affected by state transitions lasting longer than the sampling period (see Table 3.7 and corresponding analytical contexts).

To address the above issues and enhance NILM methods, including the GSP-based NILM solutions proposed in Chapter 3 and [16, 40], two *universal* algorithms are proposed. First, capitalizing on recent advances in GSP filtering (see [51, 62, 63]), a novel signal processing approach is proposed to mitigate sensor noise and sharpen signal edges to improve detection of on/off appliance events, which in turn facilitates more effective feature extraction and classification in NILM. Two types of GSP filters are designed- one based on total variation regularization [62] and the other based on bi-

lateral filtering [64]. Since graph bilateral filtering results in a smoother output at the cost of occasionally filtering out true events, an algorithm is developed to select, automatically, the best filtering method. Second, relying on robust GSP-based *data* classification [16, 51, 63], a novel NILM result *refining* method is proposed, applicable to any NILM algorithm; this method is based on semi-supervised GSP-based feature matching to improve disaggregation results by removing confusion between appliances with similar power levels that are often misclassified by the initial NILM classification engine.

The methods are proposed as generic tools for improving the disaggregation performance of a range of NILM approaches, including supervised, semi-supervised and unsupervised NILM. The effectiveness of the proposed methods are demonstrated across three state-of-the-art NILM approaches, based on DT [27], supervised GSP [16] and unsupervised GSP NILM proposed in Chapter 3. Besides the aforementioned methods, the performance is also benchmarked against two additional NILM methods from the publicly available NILMTK toolbox based on HMM and Combinatorial Optimization (CO) [5], [18]. Results are validated using two open-access datasets of true power measurements: REDD [60] (US houses) and REFIT [61] (UK houses).

The rest of this chapter is organized as follows. The details of the proposed methods in Section 4.2. Section 4.3 shows experimental results with appropriate validation and benchmarking. Findings of this study are discussed in Section 4.4.

4.2 Proposed Signal Processing Algorithm for Enhancing NILM

As defined in Section 2.1.1, \mathbf{P} represents the household's aggregate active power consumption signal and \mathbf{P}^m refers to the active power consumption signal for any known appliance $m \in \mathcal{M}$. Then, $\Delta\mathbf{P}$ and $\Delta\mathbf{P}^m$ are corresponding variation signals, respectively.

To ensure good event detection by NILM, it is essential to reduce the influence of noise while keeping signal edges sharp. Therefore, a signal processing method is

designed, illustrated in Fig. 4.1, that takes advantage of the fact that the power signal should be piecewise smooth, i.e., jitter and spikes, due to sensor malfunction, transients and power noise, will be filtered out.

Firstly, L_M -length median filtering is applied to the aggregate power measurements P_i to remove outliers. L_M must be carefully chosen according to the signal attributes, such as granularity, to ensure that relevant events are not lost. Secondly, graph filtering is applied on ΔP_i to ensure piecewise smoothness of the power signal. Then, edge sharpening is used to merge unclear consecutive edges before NILM. Finally, NILM results for the appliances with similar operating power levels can be optionally refined.

A graph filtering is performed on overlapping sample windows Ω of size L_G of the power variation signal ΔP_i obtained after median filtering. Either a graph filter via graph total variation regularization or its variant based on bilateral filtering (BF) is exploited, depending on the magnitude range of the data in the present window. That is, let T_G be a pre-set magnitude threshold; if all samples, P_j (that is, the output signal values of the median filter), in the current window Ω , meet the condition that $|P_j| \leq T_G$, $j = 1, \dots, L_G$, then Ω will be filtered by a BF-based graph filter introduced in Subsection 4.2.2. Otherwise, the samples in window Ω will be filtered by the graph filter presented in Subsection 4.2.1.

4.2.1 Graph Filtering via Total Variation Regularization

Let $\mathbf{x} = \mathbf{P}$ be an N -length vector of noisy power signal measurements (after median filtering). That is, the input signal \mathbf{x} for graph filtering is set as the current aggregate power signal. Since \mathbf{x} is a time-series signal, a graph $\mathcal{G} = (\mathcal{V}, \mathbf{A})$ is designed, where adjacency matrix \mathbf{A} is given by Eq. (2.6). Then, the graph filtering-based denoising can be formulated as an optimisation problem over all possible graph signals \mathbf{s} [62]:

$$\arg \min_{\mathbf{s}} \frac{1}{2} \|\mathbf{s} - \mathbf{x}\|_2^2 + \alpha \frac{1}{2} \|\mathbf{s} - \mathbf{A}\mathbf{s}\|_2^2. \quad (4.1)$$

The cost function in Eq. (4.1) consists of a quadratic fidelity term to maintain similarity between denoised, output signal and the input signal \mathbf{x} , and the quadratic Laplacian

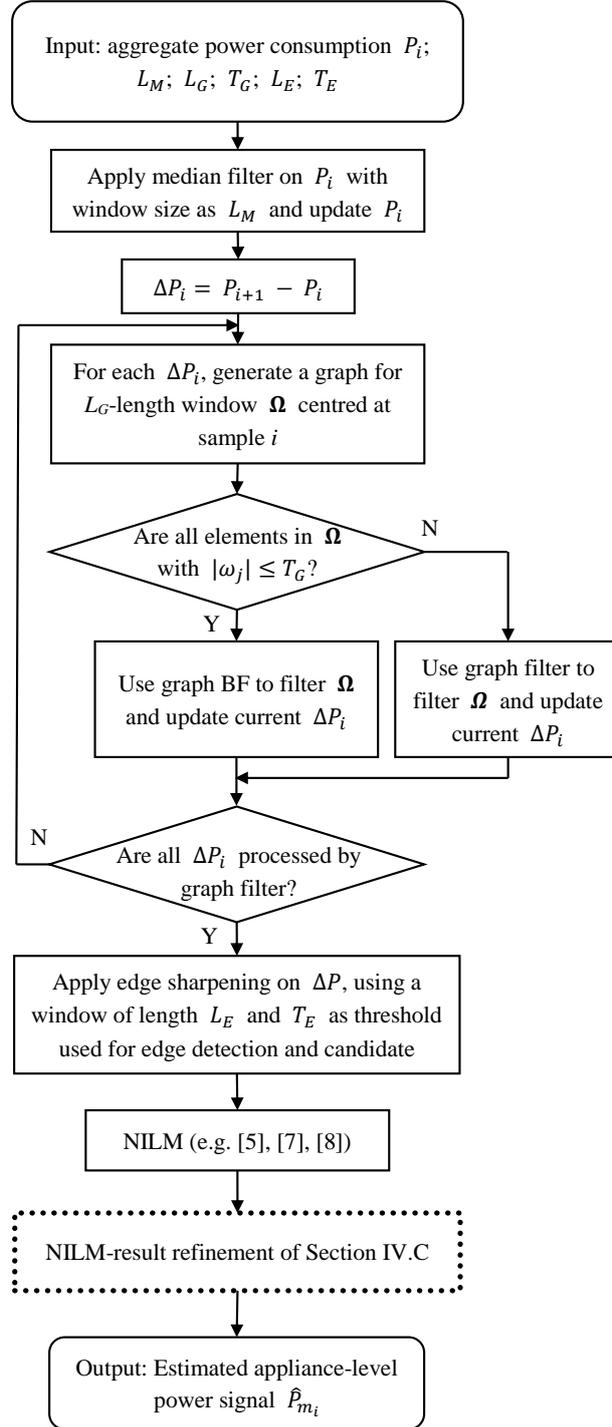


Figure 4.1: Flowchart of the proposed algorithm.

smoothness term to guarantee global smoothness of the output graph signal. α in Eq. (4.1) is chosen to trade-off fidelity and smoothness terms. The smoothness ensures that there are no sudden spikes in the signal that are characteristic of noisy measurement, appliance power value fluctuations around its mean that can cause misclassification, and transients. Thus, a piece-wise smooth signal over the underlying graph will be found, closest to the input signal \mathbf{x} .

Eq. (4.1) can be solved by calculating the first derivative of the cost function with respect to the filter input [62]:

$$\begin{aligned} \frac{\partial}{\partial x} \left(\frac{1}{2} \|\mathbf{s} - \mathbf{x}\|_2^2 + \alpha \frac{1}{2} \|\mathbf{s} - \mathbf{A}\mathbf{s}\|_2^2 \right) = \\ \frac{1}{2} \frac{\partial}{\partial x} ((\mathbf{s} - \mathbf{x})^*(\mathbf{s} - \mathbf{x}) + \alpha \mathbf{s}^*(\mathbf{I} - \mathbf{A})^*(\mathbf{I} - \mathbf{A})\mathbf{s}) = \\ (\mathbf{s} - \mathbf{x}) + \alpha(\mathbf{I} - \mathbf{A})^*(\mathbf{I} - \mathbf{A})\mathbf{s}, \end{aligned} \quad (4.2)$$

where $(.)^*$ denotes the conjugate transpose. If the first derivative is set in Eq. (4.2) to 0, the minimisation in Eq. (4.1) is solved as:

$$\tilde{\mathbf{s}} = (\mathbf{I} + \alpha(\mathbf{I} - \mathbf{A})^*(\mathbf{I} - \mathbf{A}))^{-1}\mathbf{x}. \quad (4.3)$$

Eq. (4.3) is a closed form solution which presents the exact filtered graph signal. Since the computational complexity of Eq. (4.3) is $O(N^3)$, where N is the number of samples in \mathbf{x} , the filtering is not practical when dealing with large scale entries [62]. Hence, in practice, filtering is performed on sliding windows Ω of manageable size L_G .

Note that for each input x_i , a graph is generated and the filtering is performed as described above. An example is shown in Fig. 4.2, where x_k is the central element in an L_G -length window $\Omega = \mathbf{x}_{i:j}$, with $j - i = L_G - 1$ and $k = (i + j)/2$. Note that each vertex v_i of \mathcal{G} corresponds to one power measurement x_i in Ω . Although $\tilde{\mathbf{s}}$ calculated by Eq. (4.3) gives the exact filtered signal for the whole $\mathbf{x}_{i:j}$, only the central element, x_k , is replaced by the corresponding filtered solution $\tilde{s}_{\frac{L_G+1}{2}}$. Similarly, the filtered output for x_{k+1} will be generated using a new graph whose entries are $\mathbf{x}_{i+1:j+1}$, and so on.

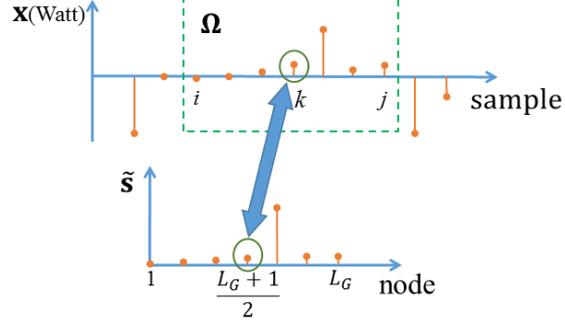


Figure 4.2: Example of graph filtering via total variation regularization, performed on window Ω of power measurements.

4.2.2 Graph-based Bilateral Filtering

Assume a graph $\mathcal{G} = (\mathcal{V}, \mathbf{A})$, with an adjacency matrix \mathbf{A} defined by Eq. (2.6), and a diagonal degree matrix $\mathbf{D} \in \mathbb{C}^{N \times N}$ defined in Section 2.4. For an input signal \mathbf{s} , the output signal after bilateral filtering is [64]:

$$\tilde{\mathbf{s}} = \mathbf{D}^{-1} \mathbf{A} \mathbf{s}, \quad (4.4)$$

where $\mathbf{D}^{-1} \mathbf{A}$ denotes the BF operator. Then obtained by inserting the BF operator into Eq. (2.9):

$$S_2(\mathbf{s}) = \frac{1}{2} \left\| \mathbf{s} - \mathbf{D}^{-1} \mathbf{A} \mathbf{s} \right\|_2^2. \quad (4.5)$$

Then Eq. (4.1) can be written as:

$$\arg \min_{\mathbf{s}} \frac{1}{2} \left\| \mathbf{s} - \mathbf{x} \right\|_2^2 + \alpha \frac{1}{2} \left\| \mathbf{s} - \mathbf{D}^{-1} \mathbf{A} \mathbf{s} \right\|_2^2. \quad (4.6)$$

The first derivative of the cost function in Eq. (4.6) is:

$$\begin{aligned} \frac{\partial}{\partial x} \left(\frac{1}{2} \left\| \mathbf{s} - \mathbf{x} \right\|_2^2 + \alpha \frac{1}{2} \left\| \mathbf{s} - \mathbf{D}^{-1} \mathbf{A} \mathbf{s} \right\|_2^2 \right) = \\ \frac{1}{2} \frac{\partial}{\partial x} \left((\mathbf{s} - \mathbf{x})^* (\mathbf{s} - \mathbf{x}) + \alpha \mathbf{s}^* (\mathbf{I} - \mathbf{D}^{-1} \mathbf{A})^* (\mathbf{I} - \mathbf{D}^{-1} \mathbf{A}) \mathbf{s} \right) = \\ (\mathbf{s} - \mathbf{x}) + \alpha (\mathbf{I} - \mathbf{D}^{-1} \mathbf{A})^* (\mathbf{I} - \mathbf{D}^{-1} \mathbf{A}) \mathbf{s}, \end{aligned} \quad (4.7)$$

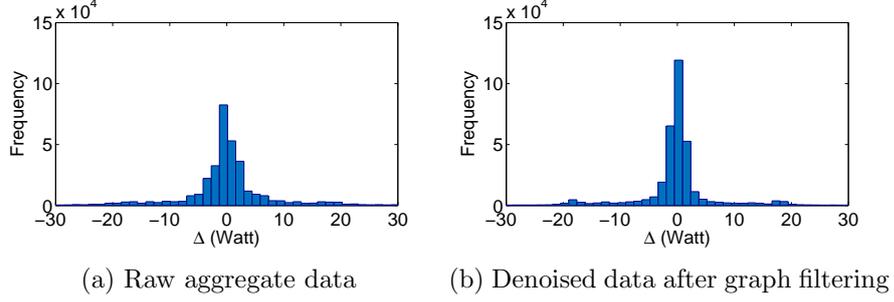


Figure 4.3: Histogram of $\Delta_i = |\Delta P_i - \sum_{m \in \mathcal{M}} \Delta P_i^m|$ for House 17 of REFIT dataset.

and after setting the first derivative to zero we get:

$$\tilde{\mathbf{s}}_{BF} = (\mathbf{I} + \alpha(\mathbf{I} - \mathbf{D}^{-1}\mathbf{A})^*(\mathbf{I} - \mathbf{D}^{-1}\mathbf{A}))^{-1}\mathbf{x}. \quad (4.8)$$

Compared with Eq. (4.3), Eq. (4.8) contains operator weights averaged by neighbouring graph nodes and gives *smoother* filtering results, useful for flattening small signal fluctuations. BF is commonly used in image denoising to smoothen/denoise each pixel by taking the weighted average of the nearby pixels [64]. However, the graph filter via BF will also split sharp edges into multiple segments, which will affect the edge detection accuracy. Therefore, only BF is applied to the entries in the windows which are magnitude-wise small, and which significantly suffer from signal fluctuations (see Fig. 2.3b).

An example is shown in Fig. 4.3, as a comparison of the distribution of the “noise”, i.e., the difference between the total measured power and the sum of all known loads, (a) before and (b) after BF graph filtering. The difference is significantly reduced, including the standard deviation which is reduced from 236.79 to 211.52.

4.2.3 Edge Sharpening

The essential step in identifying events to be classified is edge detection. To ensure successful edge detection, the final signal processing step is applying edge sharpening to the graph-filtered power variation signal. Edge sharpening is used to merge the consecutive rising edges or the consecutive falling edges caused by state changes lasting more than one sample in the time-series power signal. It can be applied to any event-

based NILM approach, e.g., [27], as these NILM approaches rely on accurate edge magnitude information during feature extraction.

Edge sharpening is performed on sample windows of size L_E samples, i.e., only the consecutive rising edges or falling edges within this window are allowed to be merged (see Fig. 4.1). T_E is the pre-set magnitude threshold for edge sharpening, i.e., only edges with magnitudes above T_E are merged.

4.2.4 Separating Similar Loads via NILM-result Refinement

Some appliances have similar operating power levels, for example, bathroom GFI (1608W) and microwave (1526W) in REDD House 1 (see Fig. 4.4a); and Toaster (902W) and Freezer spikes (909W) in REFIT House 17. Consequently, they are misclassified, causing disaggregation errors [16, 27]. In order to overcome this problem, a post-classification refinement step is proposed. The proposed graph-based refining method is generic and applicable to any event-based NILM.

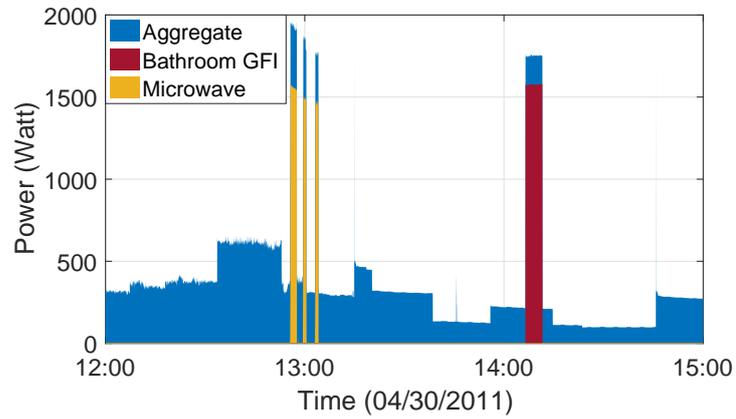
It is assumed that an event-based NILM algorithm was used to identify all appliance usage events and classify into appliances accordingly. The class associated with Appliance m contains two sets of detected rising and falling power signal edges that designate start and end of the appliance usage events, denoted by $\mathbf{\Pi}_P^m$ and $\mathbf{\Pi}_Q^m$, respectively. The task of the proposed method is then to refine the classification result using the knowledge acquired during the disaggregation process, by removing from $\mathbf{\Pi}_P^m$ and $\mathbf{\Pi}_Q^m$ power edges that do not belong to Appliance m and classifying them as Appliance n , which was labelled during NILM as the appliance with the most similar load to Appliance m .

Firstly, for two edges classified as belonging to Appliance m , i.e., caused by Appliance m being switched on and off, $\Delta P_i \in \mathbf{\Pi}_P^m$ and $\Delta P_j \in \mathbf{\Pi}_Q^m$, a *feature vector* is defined as:

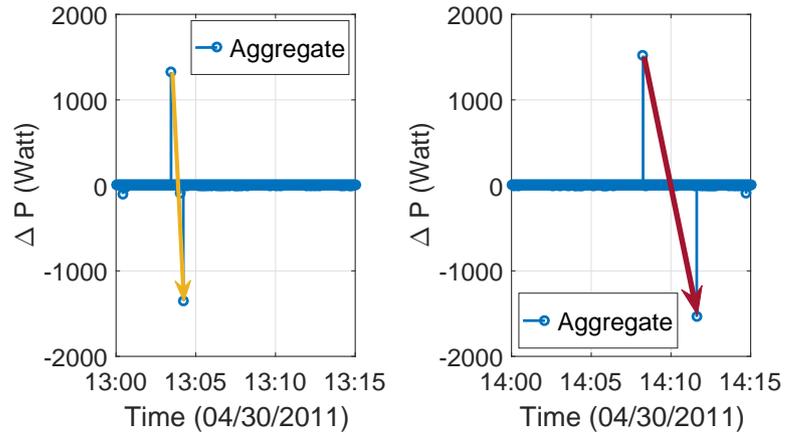
$$l_{i,j}^m = \sqrt{(\Delta t_{i,j})^2 + (|\Delta P_i| + |\Delta P_j|)^2}, \quad (4.9)$$

where $\Delta t_{i,j}$ is the time duration between the detected rising edge i and falling edge j .

An example of $l_{i,j}^m$ is shown in Fig. 4.4, where two appliances were switched on/off – microwave (switched on three times around 12:55, 13:00 and 13:04) and bathroom



(a) Aggregate power and individual load measurements for Bathroom GFI and microwave



(b) Power variation signal (ΔP_i) with a feature vector $l_{i,j}^m$ belonging to microwave (shown in yellow)
 (c) Power variation signal (ΔP_i) with a feature vector $l_{i,j}^m$ belonging to bathroom GFI (shown in red)

Figure 4.4: An example of vector $l_{i,j}^m$ for House 1 from the REDD dataset.

GFI (switched on around 14:08).

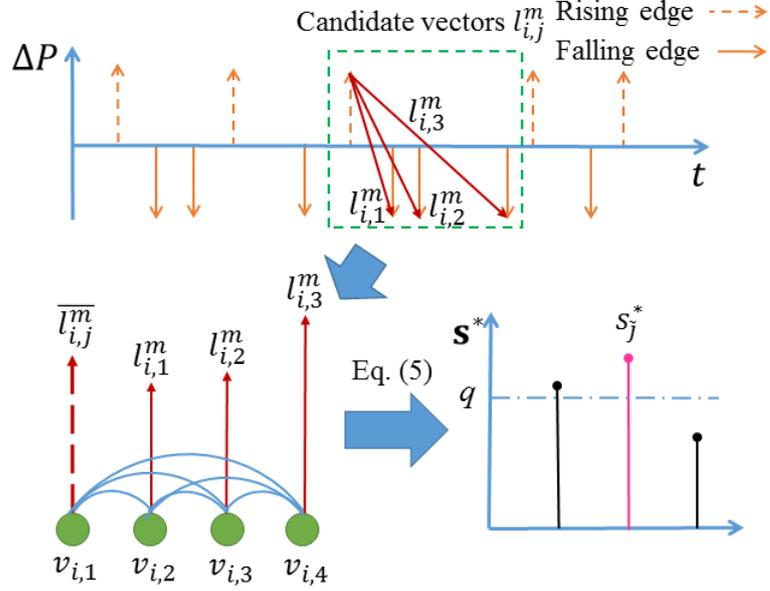


Figure 4.5: Graph generation. For a given rising edge, three vectors are selected as candidate falling edges (top). Graph with nodes corresponding to the reference vector and all candidate vectors (bottom, left). Graph total variation smoothness result s^* , with the matched falling edge corresponding to s_j^* where $\tilde{j} = 2$ (bottom, right).

Fig. 4.5 (top) shows disaggregated edges for Appliance m , containing rising edges (broken-line arrows) and falling edges (solid-line arrows). For each rising edge, all falling edges detected between that rising edge and the next rising edge (temporally) are regarded as candidates for the matched falling edge (dashed blocks in Fig. 4.5 (top)). A rising edge and a falling edge pair are a *match* if they represent the start and the end of the same appliance usage event.

For each rising edge $\Delta P_i \in \Pi_P^m$, a graph is generated, $\mathcal{G}_i = (\mathcal{V}_i, \mathbf{A}_i)$, where each vertex $v_{i,j}$, $j > 1$ (shown as a green solid circle in Fig. 4.5 (bottom, left)) in the graph corresponds to a candidate falling edge with a signal value $l_{i,j-1}^m$ defined in Eq. (4.9). The reference vector $\overline{l_{i,j}^m}$ denoted by vertex $v_{i,1}$, on the other hand, corresponds to the average $l_{i,j}^m$ for all i and j of the Appliance m class. Note that the graph adjacency matrix is defined as Eq. (2.5), where $x_1 = \overline{l_{i,j}^m}$ and $x_n = l_{i,n-1}^m$ for $n > 1$.

For the constructed graph \mathcal{G}_i , a graph total variation minimisation is performed as

in Eq. (2.7), where $k = 1$ and $\mathbf{s}_{1:k}$ in Eq. (2.13), corresponds the reference vector and is set to 1, leading to the solution \mathbf{s}^* . Using a threshold $0 < q \leq 1$, if there exists a solution greater than q , then the matched falling edge for a rising edge ΔP_i will be set to ΔP_j with $s_j^* \geq s_i^*$, $\forall j$. Fig. 4.5 (bottom, right) gives the result calculated by Eq. (2.13), where the selected edge is shown in red. For the case when $s_j^* \leq q$, the rising edge ΔP_i and all candidates ΔP_j will be classified as Appliance n .

The procedure is independently repeated for all rising edges $\Delta P_i \in \mathbf{\Pi}_P^m$ in the window. Note that because of efficiency and complexity, the proposed vector-feature matching is only designed for the refinement of a subset of loads likely to be misclassified, that is, for appliance classes with similar loads.

4.3 Experimental Validation

In this section, the experimental results are reported. Specifically, the proposed signal processing methods are tested in conjunction with three low-complexity, state-of-the-art event-based NILM approaches: 1. DT-based NILM of [27], 2. supervised GSP-based (SGSP) NILM [16], and 3. unsupervised GSP-based (UGSP) NILM proposed in Chapter 3. Additionally, the proposed methods are benchmarked with the baseload removal (BR) pre-processing method of [56], also in conjunction with NILM algorithms of [16,27]. The aforementioned results are also compared with the state-of-the-art state-based NILM algorithms, i.e., FHMM-based and CO-based methods from the NILMTK [18] with median filtering for pre-processing.

The same as in Chapter 3, active power readings can be used from two open-access datasets - the US REDD dataset [60], at 1Hz resolution, and the UK REFIT raw dataset [61], collected every 6 to 8 seconds. REFIT dataset is more challenging (see Table 4.5), as the houses contain noise due to numerous unknown appliances and some measurement noise. Two houses from each dataset are selected with varying levels of unknown appliance influence.

For all results presented, experiments were carried out over a full month for both REDD and REFIT houses; specifically the period 18/04/2011 - 24/05/2011 for REDD

House 1, 17/04/2011 - 22/05/2011 for REDD House 2, 01/10/2014 - 31/10/2014 for REFIT Houses 2 and 17. Note that the entire available dataset is used for two REDD houses. For REFIT houses, the same periods are used as in [16] to benchmark.

The abbreviations of domestic loads, using the labels provided in the datasets, to be disaggregated are as follows: B for Bathroom GFI; DW for Dishwasher; F for Fridge; KO for Kitchen outlet; L for Light; M for Microwave; O for Oven; WD for Washer dryer; S for Stove; FFZ for Fridge-freezer; K for Kettle; T for Toaster; WM for washing machine and FZ for Freezer.

In all results tables and figures, ‘P’ denotes the proposed algorithm, as illustrated in Fig. 4.1. For example, P-UGSP is the UGSP NILM method proposed in Chapter 3, used in conjunction with the proposed method, where the NILM block of Fig. 4.1 is UGSP NILM.

Table 4.1: Parameter setting used in the experiments.

Parameter	
L_M for REDD houses	31
L_M for REFIT houses	5
L_G	11
T_G	50 Watt
ρ in graph filtering via BF	30
α in graph filtering via total variation regularization	1
ρ in graph filtering via total variation regularization	$\max(\Omega)*0.3$
α in graph filtering via BF	$\max(\Omega)/20000$
T_E	100 Watt
L_E	5
ρ in NILM-refining	5
K	10%
q	0.98

Parameter values for the algorithms are chosen as discussed next (see Table 4.1). The window size for the median filter is defined as $L_M = 2\lfloor 15f \rfloor + 1$, where f is the dataset’s sampling frequency in Hz. L_M corresponds to an odd window size of roughly 30 seconds to avoid false flattening of short operational states and false segmentation of edges. Thus, L_M is set to 31 and 5 for REDD and REFIT datasets, respectively. L_G is heuristically fixed to 11 in graph filtering to trade-off complexity and perfor-

mance, exploiting, in this way, the correlation between sample x_i with samples $\mathbf{x}_{i-5:i+5}$. $T_G = 50\text{W}$ is set for all datasets, which implies that all detected power changes below 50W will be filtered using graph BF instead of graph filter. This significantly avoids over-smoothing and improves performance. ρ and α for graph filtering via BF, are heuristically fixed to 30 and 1, respectively. On the other hand, for graph filtering via total variation regularization, ρ and α are adaptively generated based on the input signal values. That is, for a current window Ω , $\rho = \max(\Omega) * 0.3$ and $\alpha = \max(\Omega)/20000$. T_E , an edge magnitude threshold for edge sharpening, is set to 100W for all datasets to minimise the influence of measurement noise and magnitude-wise low loads which usually have rapid state transitions. L_E is empirically fixed as 5 to reduce complexity and maximize edge sharpening performance. In the proposed NILM-result refining step, ρ is set to 5 to achieve high classification precision. K and q are the same as in Chapter 3 for evaluating cluster quality for all appliances and reducing falsely clustered edges, respectively.

The results are evaluated by comparing the output with the sub-metered measurements, as is common practice. The evaluation metrics used in this chapter are *F-measure* (F_m), *Acc^m*, *TER* and *DEM*, as described in Section 2.5. Note that for comparing results with the state-based NILM methods such as FHMM, confusion matrices for F_m calculation are built for ON/OFF state transitions as in [20].

Subsection 4.3.1 presents the results with and without the optional NILM-result refinement block, described in Subsection 4.2.4. Then, in Subsection 4.3.2, the F_m and *Acc* results for the proposed method are presented, shown in Fig. 4.1 with aforementioned benchmark algorithms. Finally, in Subsection 4.3.3, results obtained with the *TER* and *DEM* metrics are discussed to shed further insight into the effect of the proposed methods and their impact on the NILM algorithm performance.

4.3.1 Disaggregation Results using the Proposed NILM-result Refinement Approach

In this subsection, the performance of the proposed NILM-result refinement approach of Subsection 4.2.4 is highlighted for separating loads that are similar power-wise. From

Table 4.2, it can be seen that F_m and Acc^m improve by 0.13 and 0.07, respectively, on average, for all three disaggregation methods for similar appliance loads from REDD houses.

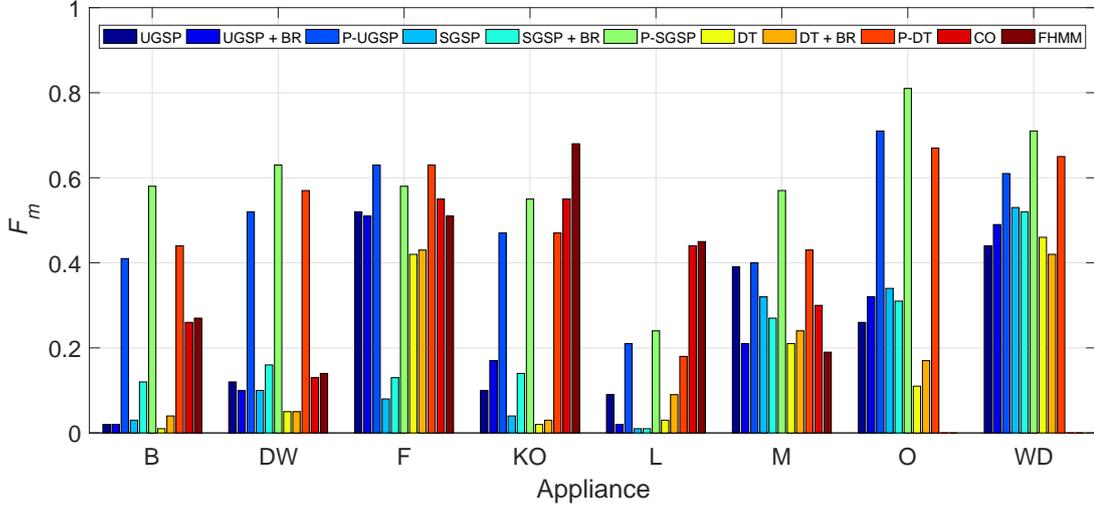
Table 4.2: Performance of the proposed NILM-result refinement for multiple appliances. N refers to disaggregation with the proposed methods before NILM-result refinement in Fig. 4.1. P refers to disaggregation with all steps of Fig. 4.1, including NILM-result refinement (discussed in Subsection 4.2.4).

Dataset			REDD				REFIT				
House			House 1		House 2		House 2		House 17		
Appliance			B	DW	KO1	M	FFZ	DW	WM	K	FFZ
UGSP	F_m	N	0.04	0.39	0.72	0.64	0.28	0.38	0.74	0.81	0.48
		P	0.41	0.52	0.82	0.69	0.42	0.79	0.76	0.84	0.5
	Acc^m	N	0.15	0.42	0.77	0.76	0.58	0.4	0.5	0.68	0.61
		P	0.44	0.66	0.89	0.82	0.77	0.42	0.53	0.79	0.66
SGSP	F_m	N	0.53	0.54	0.81	0.73	0.39	0.68	0.74	0.96	0.79
		P	0.58	0.63	0.9	0.83	0.59	0.73	0.77	0.96	0.82
	Acc^m	N	0.51	0.71	0.87	0.84	0.81	0.51	0.57	0.78	0.67
		P	0.53	0.72	0.86	0.85	0.8	0.67	0.61	0.8	0.7
DT	F_m	N	0.36	0.44	0.73	0.51	0.33	0.73	0.79	0.94	0.83
		P	0.44	0.57	0.85	0.78	0.54	0.73	0.78	0.95	0.82
	Acc^m	N	0.31	0.56	0.86	0.71	0.7	0.35	0.48	0.72	0.56
		P	0.39	0.58	0.86	0.76	0.73	0.61	0.52	0.77	0.67

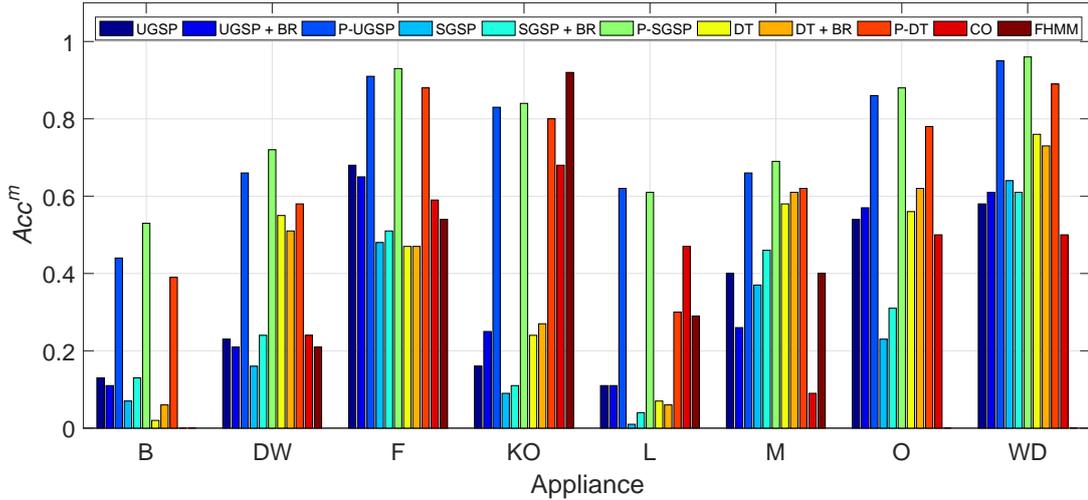
For REFIT houses, an average improvement of 0.08 for both F_m and Acc^m is observed. Significant improvement in the disaggregation of Bathroom GFI in REDD House 1 and Dishwasher in REFIT House 2 can also be observed. The proposed NILM-result refinement method further refines the clusters containing state transition events due to Bathroom GFI and Microwave but labelled as Bathroom GFI. Similarly, low-state transition events of Dishwasher from REFIT House 2 are further separated from the clusters of events labelled as Fridge-freezer during the proposed refinement step. REFIT House 17 contains an unknown appliance whose power level is similar to Kettle, leading to high ITP for Kettle. With the proposed GSP-based feature matching refinement, some events of Kettle are isolated and correctly labelled as ATP which resulted in a corresponding increase in both F_m and Acc^m .

From this point, the proposed method (P) refers to the scheme with the NILM-result refinement, i.e., with all steps of Fig. 4.1.

4.3.2 F_m and Acc^m Appliance-level Results



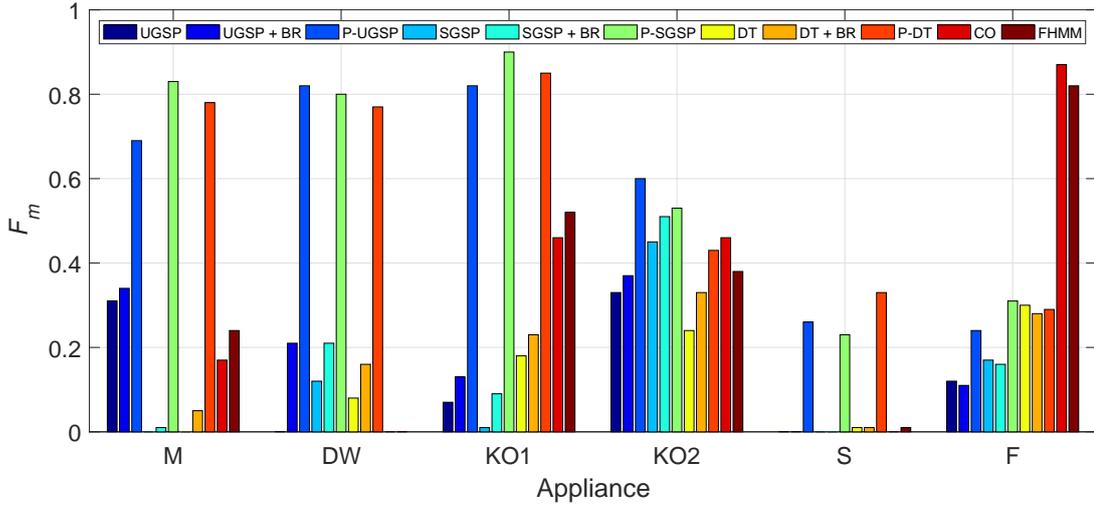
(a) F_m



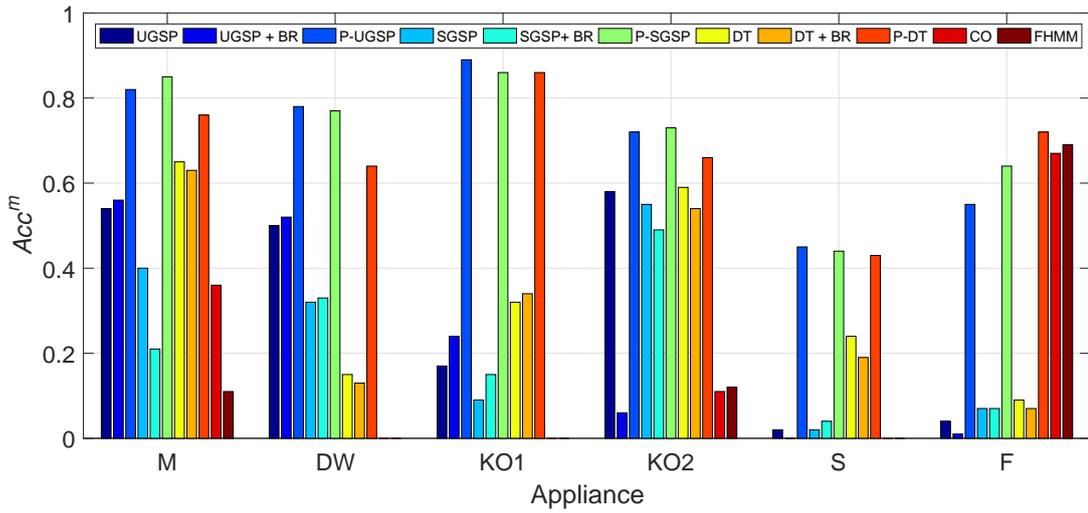
(b) Acc^m

Figure 4.6: Performance of the proposed method (P) with benchmarks UGSP, UGSP with BR [56], SGSP [16], SGSP [16] with BR [56], DT [27], DT [27] with BR [56], CO and FHMM for REDD House 1. Note that B is bathroom GFI; DW is dishwasher; F is fridge; KO is kitchen outlet; L is lighting; M is microwave; O is oven and WD is washer dryer.

The performance of the proposed and benchmark methods for House 1 from REDD dataset is shown in Fig. 4.6. The performance improvement due to the proposed algo-



(a) F_m



(b) Acc^m

Figure 4.7: Performance of the proposed method (P) with benchmarks UGSP, UGSP with BR [56], SGSP [16], SGSP [16] with BR [56], DT [27], DT [27] with BR [56], CO and FHMM for REDD House 2. Note that M is microwave; DW is dishwasher; KO is kitchen outlet; S is stove and F is fridge.

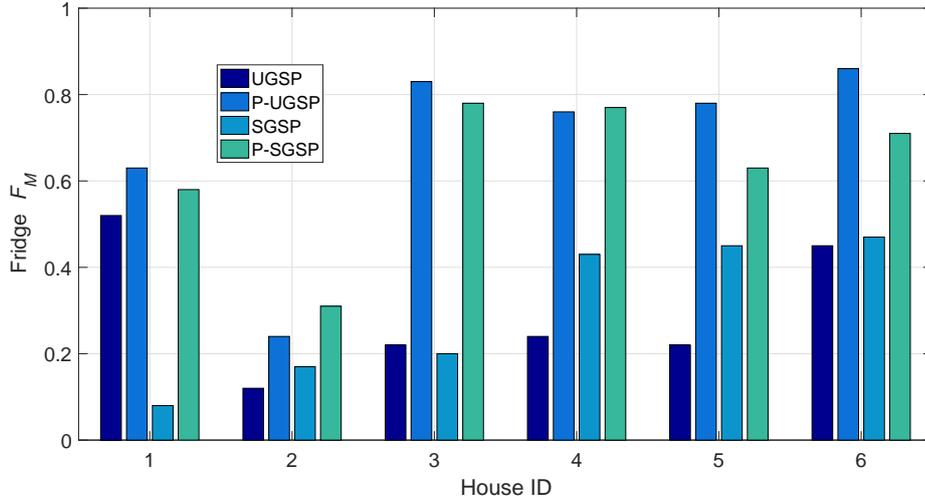
rithm is noticeable for all eight appliances for all three event-based NILM approaches, with average F_m improvement of 0.25 for P-UGSP over the results shown in Chapter 3, 0.4 for P-SGSP over [16] and 0.34 for P-DT compared to [27]. Similarly, an average Acc^m improvement of 0.39 for P-UGSP, 0.51 for P-SGSP and 0.25 for P-DT is observed

compared to their respective benchmarks. Worth noting is the oven (O), a significantly large load, which could hardly be disaggregated without the proposed methods with $F_m \leq 0.4$, but with the proposed method can achieve F_m classification accuracy between 0.7 and 0.8. This is because the state transition duration in the case of oven is often longer than the sampling period (1sec), affecting edge detection, that is, instead of detecting a single rising/falling edge due to the oven being switched on/off, multiple edges of smaller amplitude will be detected causing subsequent classification errors. However, the proposed GSP-filtering based method merges the segmented edges into a distinct edge transition. This is shown in Fig. 4.11a.

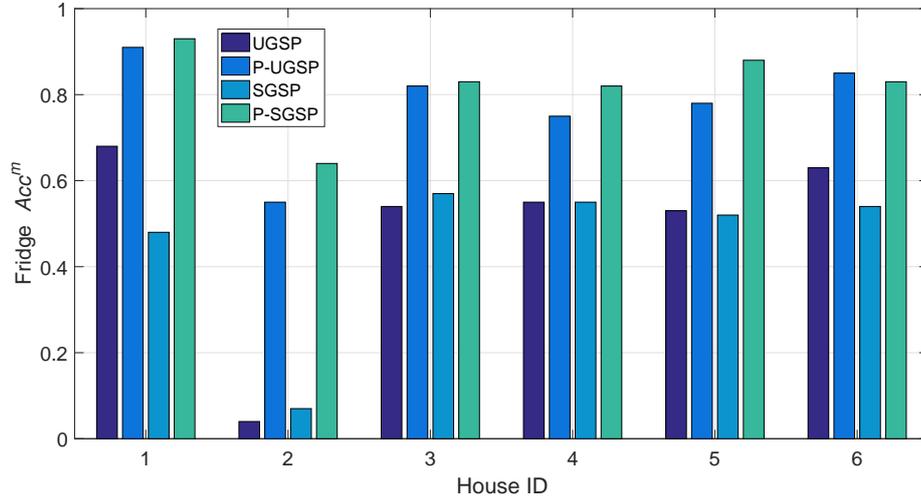
Similar results are obtained for REDD House 2, as can be seen from Fig. 4.7. In this case, the average Acc^m results among all appliances with the proposed method are 0.7 for P-UGSP, 0.73 for P-SGSP, and 0.68 for P-DT, which are significantly better than the benchmarks and competitive with NILM methods based on FHMM and its extensions [25], bearing in mind that the period of testing in [25] is unreported.

Furthermore, from Fig. 4.6 and 4.7, it can be seen that removal of the baseload (the BR method in [56]) does not show performance improvement compared to the results without BR, and may lead to worse performance for some appliances such as, Washer Dryer (WD) for disaggregation based on SGSP and DT, and Microwave (M) for disaggregation based on UGSP. This can be explained as follows: with data granularity of 1 sec, the appliance operational state transition edges can be captured by the BR method as consecutive smaller edges in power measurements; since BR identifies small power changes under 50 Watts as fluctuations of the baseload, some small discrete edges will be removed, leading to worse classification results. The performance of state-based NILM methods such as FHMM-based methods is possibly affected as low-value states will be removed while small edges are removed.

Similar results were obtained for other REDD houses, shown in Figs. 4.8 and 4.9, for four regularly used appliances present in all 6 REDD houses. The appliance-level NILM results for fridge are particularly demonstrated in Fig. 4.8 as fridge is a major consumer present in all houses. Significant improvements can be observed, 0.34 for F_m and 0.32 for Acc^m on average across houses. House-level results presented in Fig. 4.9 show F_m



(a) Fridge F_m

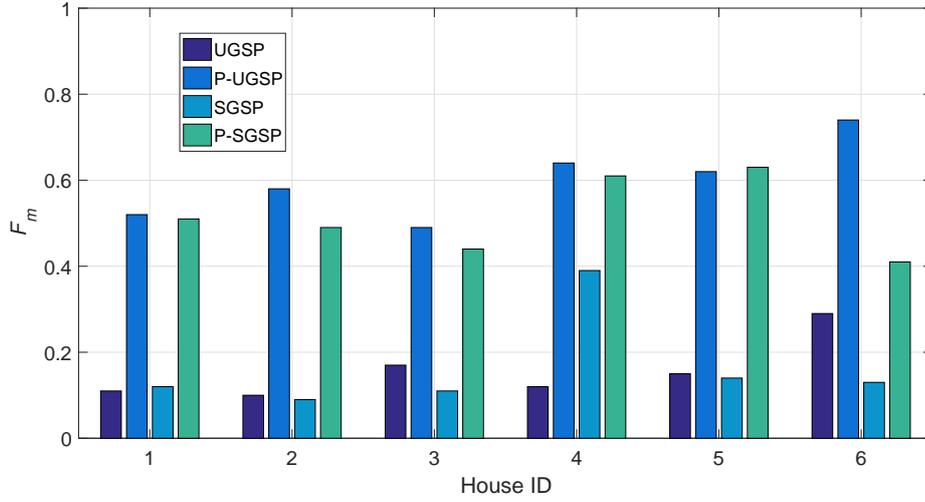


(b) Fridge Acc^m

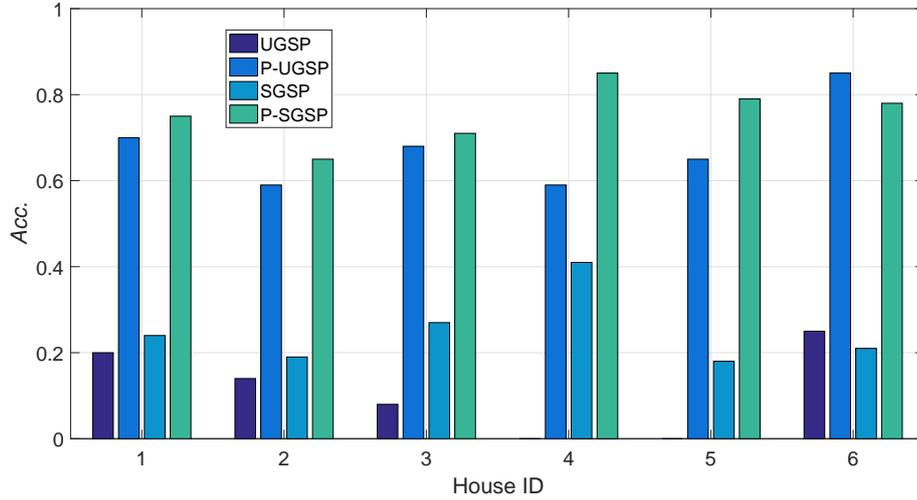
Figure 4.8: Performance of the proposed method with benchmarks UGSP, SGSP [16] for fridge in each house from REDD dataset.

and Acc improved by 0.35 and 0.54, respectively, with the proposed methods. Note that the accuracy obtained with the proposed methods varies only slightly across the houses. However, a much larger variation in performance is observed in the absence of the proposed method demonstrating robustness and consistency of the proposed approach to varying noise levels across houses.

The results in Fig. 4.9 are comparable to those of [19, 58, 71], taking into account



(a) House-level F_m

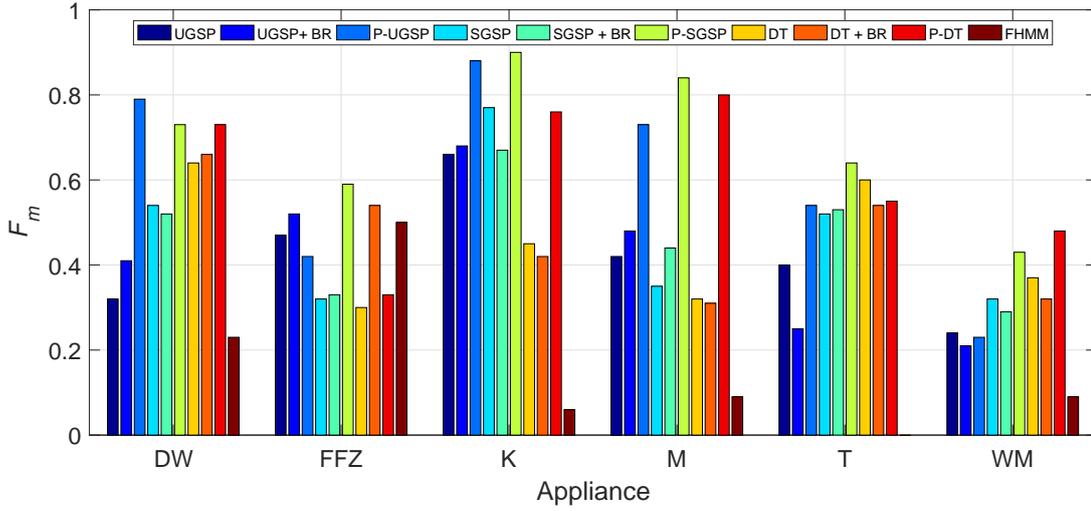


(b) House-level Acc^m

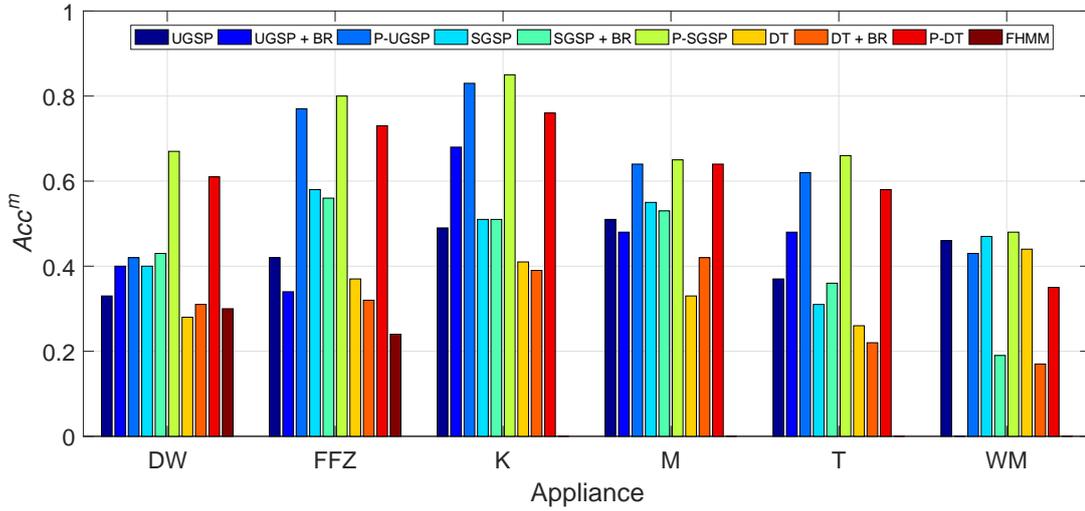
Figure 4.9: Performance of the proposed method with benchmarks UGSP, SGSP [16] for the four most common appliances (i.e., B, DW, F, WD) in each house from REDD dataset.

different sampling rates, the composition of a subset of appliances for house-level accuracy calculation, and the number of days/samples (sometimes unknown) used for training/testing.

The impact of the proposed algorithm on multiple NILM methods for REFIT House 2 is shown in Fig. 4.10. House 2 is typical in the REFIT dataset with two



(a) F_m



(b) Acc^m

Figure 4.10: Performance of the proposed method (P) with benchmarks UGSP, UGSP with BR [56], SGSP [16], SGSP [16] with BR [56], DT [27], DT [27] with BR [56] and FHMM for REFIT House 2. Note that DW is dishwasher; FFZ is fridge-freezer; K is kettle; M is microwave; T is toaster and WM is washing machine.

multi-state appliances. The F_m performance improvement of UGSP with and without the proposed method is significant for most appliances, except Fridge-freezer (FFZ) and Washing Machine (WM). Note that although the F_m of FFZ reduced a little by 0.05, caused by reduction of correctly identified state transition edges, many falsely

identified edges are avoided, resulting in 0.35 improvement in Acc^m and reduction in power consumption estimation error (see Subsection 2.5.3). The WM in REFIT House 2 is the only appliance for which the results have not improved in Acc^m , for two reasons - (a) a low number of WM events during the testing period of one month (16 runs in total); (b) large fluctuation of WM power load, resulting in poor performance for all tested methods.

Note that the largest improvement is observed for the Dishwasher (DW) because, in addition to the state transition edges being sharper for DW, multiple operational states of DW are well shaped after the proposed processing, enabling low-load events to be captured and distinguished from the cluster of Fridge-freezer. This is illustrated

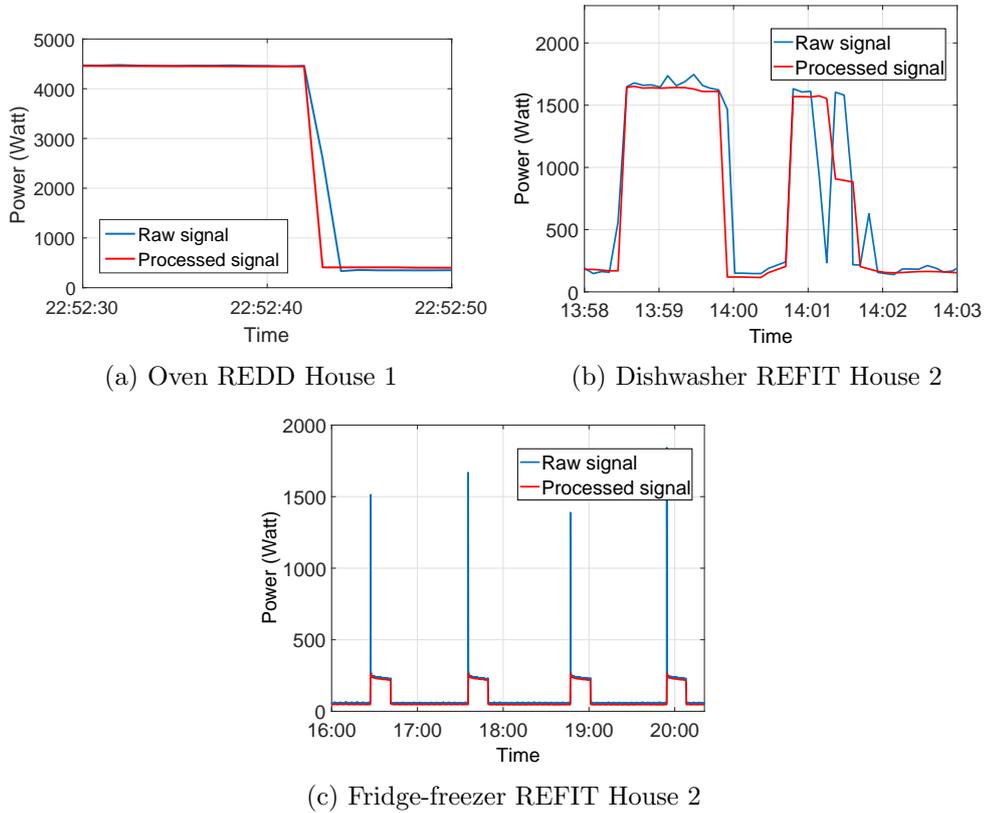


Figure 4.11: Typical appliance operation in the aggregate power consumption data before (shown in blue) and after processing via the proposed method (shown in red).

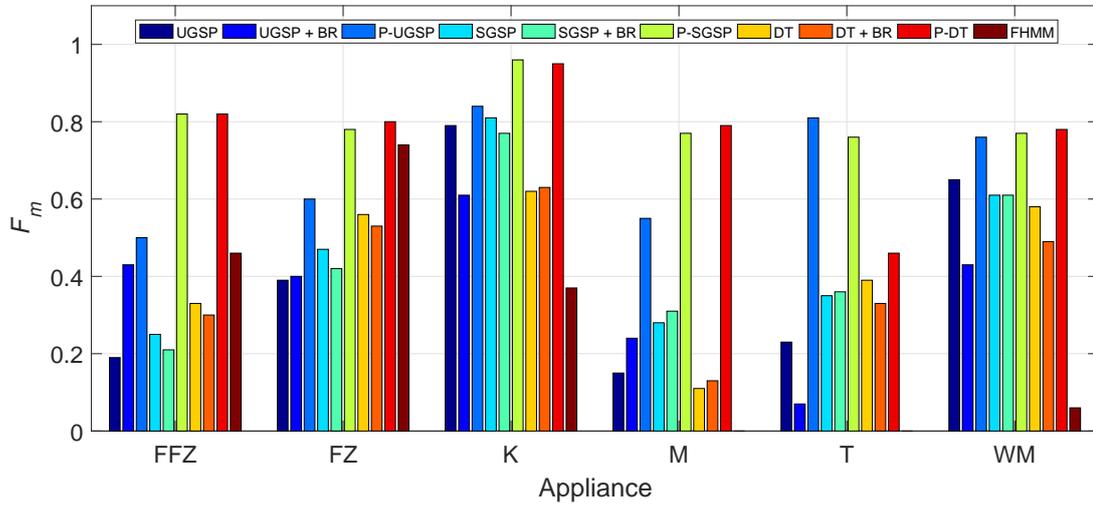
in Fig. 4.11c, which also shows an example of the raw and processed signal of Fridge-freezer; sharp spikes are removed, reducing the edge magnitude range and improving

this way the edge detection accuracy and classification precision.

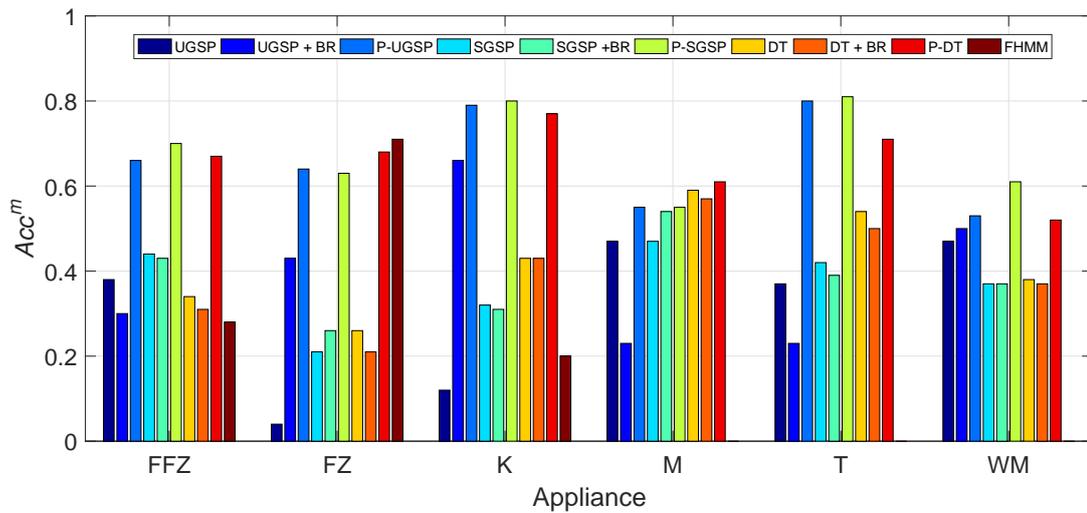
There is a significant improvement in the SGSP algorithm after applying the proposed methods, where F_m improves by 0.22 on average for all appliances. Similarly, the performance improvement between DT with and without the proposed method is noticeable for all appliances, besides Toaster, in REFIT House 2. Apart from DW in REFIT House 2, no significant improvement is observed for the case of multi-state appliances because the edges are not sufficiently denoised and sharpened as in the case of the REDD houses. This can be explained by the higher sampling rate of 1Hz for the REDD dataset, where there exist more state changes longer than 1 second which are split into multiple small segments; hence, the proposed method leads to higher gains. As for the REDD houses, baseload removal seems counterproductive for REFIT House 2 for most appliances. This is due to the consecutive low state signatures being removed during BR [56].

The FHMM-based NILM approach does not perform as well as the three event-based NILM approaches on REFIT House 2 except for Fridge since FHMM is good at identifying appliance operation cycles of fridges.

The performance of the proposed method is presented for House 17 from REFIT dataset in Fig. 4.12. In REFIT House 17, Washing Machine (WM) regularly contains significant power variations, resulting in a high number of FP s for magnitude-wise similar appliances, including Fridge-Freezer (FFZ), Freezer (FZ) and Toaster (T). Performance improvement of FFZ, FZ and T is noticeable in Fig. 4.12, since fluctuations of WM power signal are effectively reduced by the proposed method. Note that some very poor Acc^m results of FHMM were omitted. The performance of the proposed methods is comparable with the results in [24], which uses the cleaned version of REFIT data, where measurement errors were removed from the raw dataset as per [61]. The results, therefore, show that with the proposed signal processing of the raw measured signal, which would be available directly from the meter, comparable results can be obtained if a cleaned, measurement-error-free signal was used instead.



(a) F_m



(b) Acc^m

Figure 4.12: Performance of the proposed method (P) with benchmarks UGSP, UGSP with BR [56], SGSP [16], SGSP [16] with BR [56], DT [27], DT [27] with BR [56], CO and FHMM for REFIT House 17. Note that FFZ is fridge-freezer; FZ is freezer; K is kettle; M is microwave; T is toaster and WM is washing machine.

4.3.3 Further Insights with Additional Metrics

Table 4.3 shows that the normalised error between actual and total power consumption, TER as defined in Eq. (2.19), for all three NILM methods, is reduced with the proposed methods for most appliances. The proposed method not only removes outliers and spikes but also reshapes edges for better feature matching results. The relatively large estimation errors of FHMM-based method for Microwave, Toaster and Washing machine reflect over-estimation, which matches the corresponding poor performance of F_m and Acc^m in Fig. 4.10. Note that although BR significantly reduces the error for Dishwasher, it is at the expense of increased error for Fridge-freezer and Washing machine, since these two appliances are often confused with Dishwasher. Overall, as observed in the previous section, BR does not show performance improvement for the majority of appliances.

Table 4.3: Normalised total power consumption estimation error rate (TER) per appliance of the proposed method (P) benchmarked against UGSP, UGSP with BR [56], SGSP [16], SGSP [16] with BR [56], DT [27], DT [27] with BR [56] and FHMM for REFIT House 2.

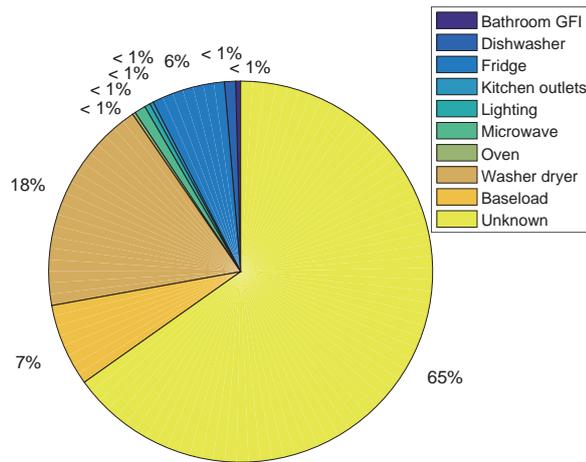
Appliance	DW	FFZ	K	M	T	WM
UGSP	0.73	0.1	0.34	0.43	0.83	0.17
UGSP + BR [56]	0.08	0.53	0.06	0.17	0.81	0.4
P-UGSP	0.66	0.31	0.05	0.34	0.82	0.09
SGSP [16]	0.47	0.19	0.27	0.52	0.9	0.32
SGSP [16] + BR [56]	0.13	0.42	0.11	0.16	0.88	0.52
P-SGSP	0.33	0.27	0.04	0.28	0.73	0.1
DT [27]	0.64	0.47	0.31	0.66	0.78	0.28
DT [27] + BR [56]	0.35	0.67	0.25	0.59	0.71	0.33
P-DT	0.49	0.31	0.15	0.63	0.42	0.22
FHMM	0.75	0.38	0.41	13.63	7.53	3

Table 4.4: Normalised total power consumption estimation error (TER) of UGSP and P-UGSP for REDD House 1.

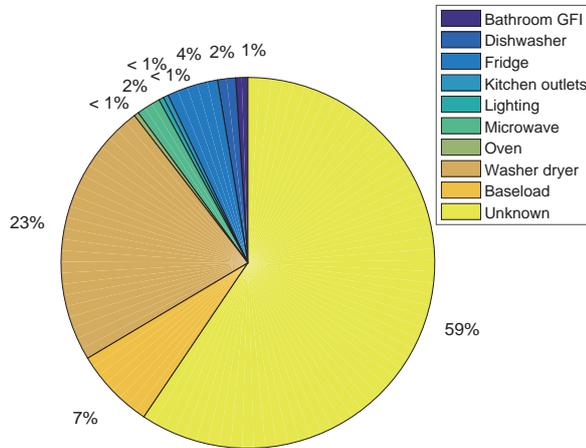
Appliance	B	DW	F	KO	L	M	O	WD	BL
UGSP in Chapter 3	0.32	0.53	0.26	0.41	0.84	0.53	0.37	0.29	0.37
P	0.7	0.22	0.1	0.15	0.86	0.03	0.01	0.11	0.37

Similar results are obtained for the REDD houses. Indeed, for REDD House 1,

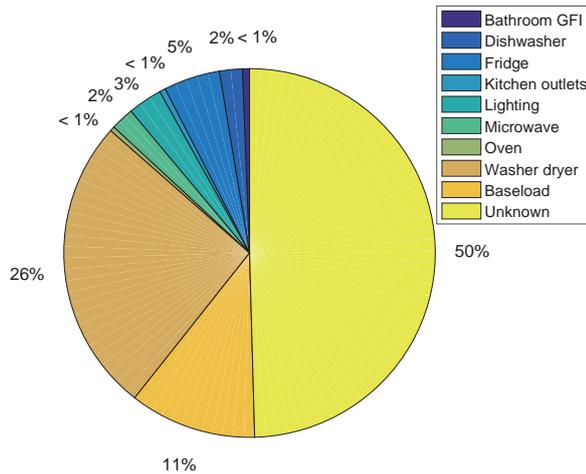
Event-based Non-intrusive Load Monitoring Improvement using Graph Signal Processing



(a) Results after disagggregation with UGSP



(b) Results after disagggregation with P-UGSP



(c) Ground truth

Figure 4.13: Disagggregation results, presented as percentage load contribution per appliance relative to the aggregate load, for House 1 from REDD datasets.

the difference between the disaggregated energy consumption and measured energy consumption per appliance is small as illustrated in Fig. 4.13 and Table 4.4 for UGSP in Chapter 3. The disaggregation results using pure UGSP, shown in Fig. 4.13a, are inline with the argument that the algorithm suffers from high sensitivity to disaggregation noise. With the proposed method, the total disaggregated energy is increased by 6%.

Similar to Fig. 4.6, it can be observed from Table 4.4 that for the majority of appliances, applying the proposed method to the event-based NILM approach proposed in Chapter 3 reduces TER . The TER increase for Bathroom GFI is due to a few microwave events misclassified as the events of Bathroom GFI, resulting in overestimation and high TER . With the benchmark, fewer events can be matched as operational state edge pairs, leading to lower estimated total consumption and lower TER . Both F_m and Acc^m results of Bathroom GFI in Fig. 4.6 indicate more accurate classification with the proposed method.

Table 4.5: F – *measure* performance for four selected houses.

Dataset		REDD		REFIT	
House		1	2	2	17
NM		0.24	0.35	0.68	0.58
DEM	P-UGSP	0.37	0.54	0.67	0.66
	P-SGSP	0.44	0.57	0.66	0.61
	P-DT	0.46	0.56	0.69	0.64
F_m	P-UGSP	0.67	0.72	0.59	0.53
	P-SGSP	0.69	0.73	0.61	0.62
	P-DT	0.7	0.68	0.59	0.6

Table 4.5 compares four selected houses, two from REDD and two from REFIT dataset, in terms of disaggregation noise using the noise measure (NM) of [19]. It can be seen that both REFIT houses have a much higher noise level compared to the REDD houses, making the disaggregation process very challenging. In addition, a comparison of overall disaggregation performance for the selected houses is shown. Note that the overall F_m values in Table 4.5 are calculated using equations (2.14), (2.15) and (2.16), by summing TP , FP and FN values for all appliances. From Table 4.5, disaggregation performance for REDD houses is generally better than that of REFIT houses, with higher F_m and lower DEM . Note that though REDD House 2 is ‘noisier’ than REDD

House 1 (based on NM inline with DEM values), the overall F_m performance for House 2 is better than that for House 1. This is because REDD House 1 contains more appliances than House 2, making the disaggregation problem more complex. For both REFIT houses, multiple disaggregation methods give DEM values which are 0.17 higher than those of REDD houses on average. Although NM values for REFIT houses are almost twice of those for REDD houses, corresponding overall F_m results are only 0.1 lower on average. One reason for this is fluctuation in baseload in REFIT houses, which increases NM , but does not significantly affect event-based NILM methods as only state transitions are considered.

4.4 Summary

This chapter addresses the challenging problem of mitigating the effect of measurement noise and unknown loads on load disaggregation (NILM) performance. Two signal processing methods based on GSP are proposed in conjunction with existing NILM approaches to improve any low-rate supervised and unsupervised event-based NILM classification and estimation accuracy. In particular, a graph-based filtering approach is proposed to clean the power signal before classification. The main motivation comes from the fact that event-based low-rate NILM approaches require clean power consumption measurements containing sharp and accurate state transition events. Besides the proposed graph-based filtering, a post-classification refinement method is proposed, which improves NILM by mitigating the effect of misclassification of loads with similar operational range.

The improvement is demonstrated in NILM performance with the proposed methods when applied to three distinct event-based NILM methods and across two real-world datasets with multiple houses with different levels of actual measured noise. The effect of sampling rate on graph filtering and edge sharpening is discussed and It is shown that the proposed method can significantly improve performance for smart meter data gathered at sampling rates of 1Hz and lower, allowing appliances which could not be detected by event-based NILM previously, possible. Assuming that the sampling

rate reduces to 15 minutes, 30 minutes or even 1 hour for commercial smart meters, fluctuations will be mitigated and state changes lasting longer than a sampling period will seldom occur, thus the proposed NILM improvement methods will no longer work. However, the problem becomes more challenging as detectable state changes on data with granularity less than 1 minute vanish while sampling rate falling to 1 hour. Thus in the next chapter, all the experience learnt from low-rate NILM is used to propose new approaches for disaggregating on data of lower sampling rate, that is, hourly electricity profile.

Chapter 5

Very Low-rate Electricity Profile Disaggregation¹

5.1 Introduction

As discussed before in Section 1.2, though various machine learning methods including [7, 16, 17, 25, 31, 32, 37, 47, 58, 65] have been applied to the NILM problem, most approaches focus on either high sampling rates in the order of kHz or MHz or sampling rates between 1sec and 1min. Disaggregation in these cases is usually performed via feature extraction and state transition modelling on active and/or reactive power data. However, mainly due to storage, data management, and privacy constraints [52], the resolution of load measurements available from roll-out smart meters is much lower, e.g., 15 minutes in Italy [53], 1 hour in Spain [54], 15 minutes or 1 hour in the US [55], 30 minutes in the UK [49], 1 hour in British Columbia and Ontario, Canada [52]. Hence, very low-rate (10-60 min) NILM is slowly gaining interest [28, 31, 32, 46, 125] since electricity meters deployed at scale in most countries tend to provide extremely low-rate measurements, at 15, 30 minutes or hourly granularity. Compared with power measurements of higher granularity, the energy consumption signal at very low granularity

¹This chapter is mainly based on the work that appeared in NILM Workshop [30] and Applied Energy [124]

features limited state transitions, fewer low-consuming appliances' feature patterns and a much higher probability of multiple appliances running simultaneously. Thus, lack of well-known features and increased appliance noise make very low NILM a challenging problem [28], which can be looked at as electricity usage profile disaggregation, since the input is the total energy use within fixed time intervals (e.g., in kiloWatt-hour), instead of active/reactive power readings collected at relatively high frequency in Watts or VARs.

This chapter focuses on solving the challenging very low-rate NILM problem, where the sampling period is at most 15 min. At first, a supervised K-NN based electricity usage profile disaggregation of energy measurements at 15 and 60 min granularity is proposed to identify a range of appliances. Relative standard deviation is proposed as a metric to determine which features are most useful for disaggregating particular appliances. Unlike [28,31,65], the disaggregation results are validated using three open-access datasets of true power measurements: REDD [60] (US houses), REFIT [61] (UK houses) and AMPds [66] (a Canadian house). For all datasets, the electricity energy profile of the aggregate load for 15-min and 1 hour in Watt-hour (*Wh*) is calculated. Although the proposed K-NN method can disaggregate up to 62% of the daily energy consumption [30], it is supervised and requires a large enough set of data for training.

Hence, a training-less very low-rate (hourly) OPT NILM approach is proposed aiming to estimate appliance-level consumption by finding the combination of appliance models generated from manufacture information, that minimises a cost function, after estimating and removing the baseload. A GSP-based NILM approach, adapted from the higher resolution GSP-based approach proposed in Chapter 3, is also proposed for hourly energy disaggregation, where an additional graph for time-of-day features is added. Unlike the proposed K-NN where knowledge of appliance usage pattern is required, the proposed OPT and GSP are training-free where GSP utilises time-of-day information. The proposed methods are evaluated on real aggregate profile data from REFIT houses against four state-of-the-art energy disaggregation methods, namely the widely used FHMM and CO, publicly available in the NILM Toolkit [18], DDSC [32] proposed for hourly data and a CNN-based approach, trained only on a small set of

aggregate profile data, where time information is considered as a feature. Both FHMM (and its variants) and CO are popular NILM solutions for disaggregation of low-rate active power (1Hz or less) e.g., [24, 31, 38].

Unlike the previously discussed NILM approaches tackling very low-rate NILM, OPT differs in the following ways. In the proposed optimisation-based approach, the tolerance is minimised between the aggregate energy consumption within each extracted duration and the sum of corresponding appliance-level consumption estimation, with consideration of one appliance running for multiple times within one sampling period and utilisation of manufacturer information. In contrast, in CO approach, the minimisation target is the tolerance between the aggregate measurements and the sum of instantaneous estimated appliance-level power per sample, based on instantaneous power for various operational states of appliance models trained from plug-level metering [5, 18]. Thus, in CO optimisation is carried out per sample independently, while in OPT samples in a particular load sequence are dependent when solving optimisation. The GSP-based approach proposed in [41] is supervised where graph edges of a single graph are weighted only based on smart metering readings. On the contrary, the GSP proposed in Chapter 5 is training-less based on the approach proposed in Chapter 3, where two graphs are generated for both energy profile and time.

Additionally, unlike many of the aforementioned literature, environmental data is not utilised. OPT and GSP are evaluated on public datasets that closely resemble real-life smart meter measurements that include many unknown appliances. Besides *detecting* appliance use (classification only) energy consumed is also estimated or disaggregated and results are provided 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.

The remainder of this chapter is organized as follows: Section 5.2 clarifies electricity profile extraction from existing high granularity datasets; the proposed supervised K-NN is presented in Section 5.3 with validation demonstrated on three datasets; Then two proposed approaches: OPT and GSP are described in Section 5.4 and Section 5.5, respectively. In Section 5.6, setup in the experiments is clarified including the data selected for validation and parameter settings. The results of all methods are

demonstrated and discussed in Section 5.7; Section 5.8 summarises the findings.

5.2 Hourly Energy Presentation Calculation

Since not in every dataset, hourly energy profile is collected like in [52,95], first it should be clarified how hourly energy profile data is calculated from existing public datasets of power measurements with a resolution less than 1 minute, i.e., real and reactive power, voltage and current on multiple phases within each collection interval.

For a daily window, $\mathbf{E}_{1:24}$ refers to 24 hourly load profile samples. E.g., E_{10} denotes total power consumption from 9:00am to 10:00am. Basically, the load profile is treated as an integral of power on time series. Here two calculation methods are listed depending on the way power measurements are collected.

5.2.1 Down-sampling assuming Fixed Sampling Intervals

In [99] and [100], very low-rate power consumption, in Watts, is calculated as the average of mean power values over N samples:

$$E_T = \frac{1}{N-1} \sum_{i=2}^N P_i. \quad (5.1)$$

Note that it starts at $t = 2$ because $t = 1$ is outside the sampling period, as shown in Fig. 5.1.

5.2.2 Down-sampling allowing Variable Sampling Intervals (Method 1)

Down-sampling 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 [61]. Therefore, in order to account for variability in sampling rates and ensuring accurate estimation of power consumption over a period comprising varying sampling rates, hourly consumption is calculated as follows. Referring to Fig. 5.1, the aim is to calculate the hourly profile E_T between

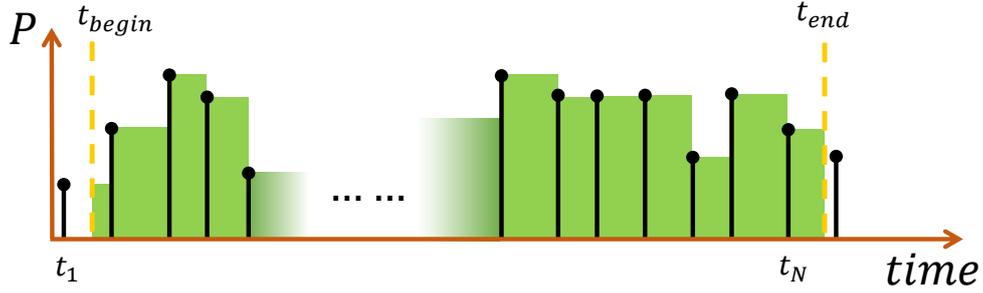


Figure 5.1: Hourly profile generation example for Method 1.

t_{begin} and t_{end} , as shown in Eq. (5.2).

$$E_T = P_1 \times (t_2 - t_{begin}) + \sum_{i=2}^{N-1} P_i \times (t_{i+1} - t_i) + P_N \times (t_{end} - t_N). \quad (5.2)$$

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. (5.2) for validation purposes.

5.2.3 Down-sampling allowing Variable Sampling Intervals (Method 2)

Another case is total energy usage is captured regularly by smart meters, where each power value equals to the quotient of energy increment from the last measuring instance and the corresponding period [126].

As an inverse procedure of power measurement calculation from temporal energy increments demonstrated in Fig. 5.2, where the period between instances t_{begin} and t_{end} refers to one whole hour, E_T is calculated as an integral to mitigate tolerance:

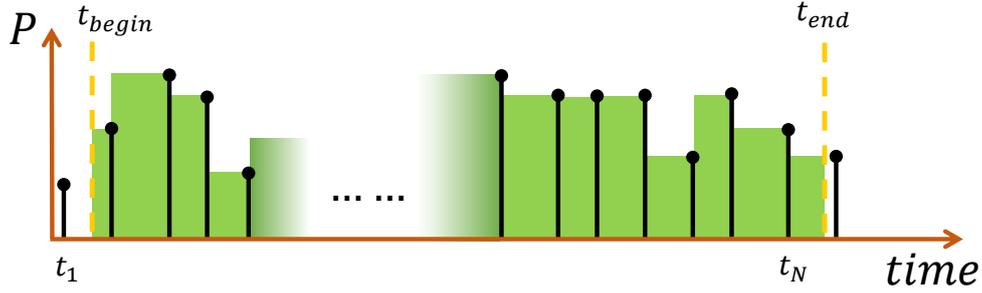


Figure 5.2: Hourly profile generation example for Method 2.

$$E_T = P_2 \times (t_2 - t_{begin}) + \sum_{i=2}^{N-1} [P_i \times (t_{i+1} - t_i)] + P_{N+1} \times (t_{end} - t_N), \quad (5.3)$$

5.2.4 Methods Discussion

Down-sampling by averaging is an appropriate method of generating hourly power usage presentation for the power measurements with an invariant sampled rate such as REDD dataset. While for the case measurements are collected with varying granularity such as REFIT dataset, down-sampling does not obtain hourly power characteristics precisely.

Table 5.1: Analytical comparison of multiple hourly profile calculation methods on House 4 from REFIT dataset

mean		standard deviation	
Eq. (5.2)	Eq. (5.1)	Eq. (5.2)	Eq. (5.1)
323.86	324.6	255.56	257.69

Table 5.1 demonstrates the mean and standard deviation values for hourly electricity profile signals L_T calculated using various methods on 90 weeks (12/10/2013–03/07/2015) of smart meter readings from REFIT House 4, containing 15120 attributes. Although the profile signals calculated via various methods have similar mean and standard deviation values, small errors on each sample influence energy profile characteristics and

algorithm robustness, especially for the approaches require parameter tuning and appliance modelling. Hence, an example is given as Fig. 5.3.

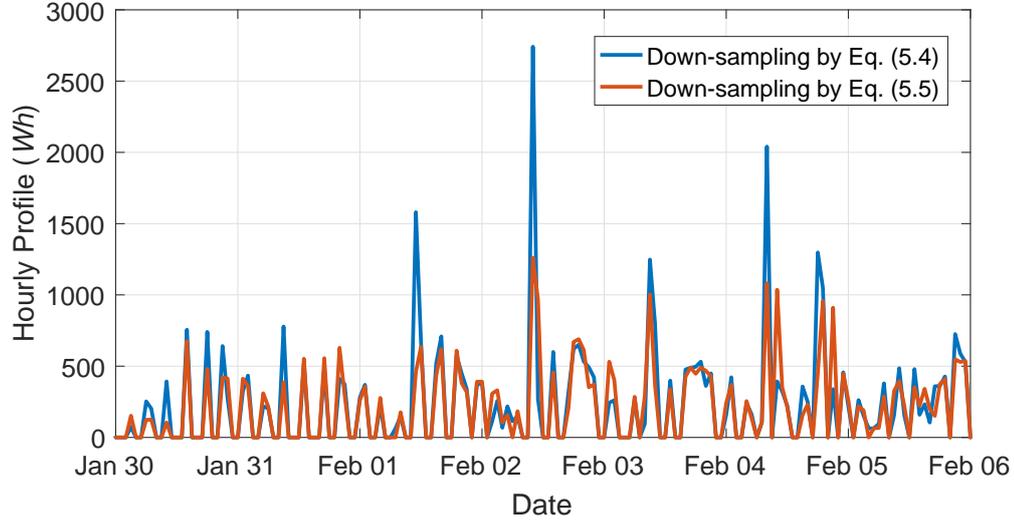


Figure 5.3: Hourly profile of House 4 from REFIT dataset calculated for a typical week.

Fig. 5.3 demonstrates that there is a difference between hourly electricity profile signals E_T calculated using the two down-sampling approaches over one week (30/01/2014–05/02/2014) of smart meter readings from REFIT House 4. Based on data collection for REFIT dataset [61], in this chapter, Eq. (5.2) is used in the experiment to generate both and hourly aggregate energy profile for simulation and corresponding appliance-level profile as ground truth for evaluation. Profile signal with granularity as 15 minutes can be calculated a similar way, by setting $t_{end} - t_{begin}$ to 15 minutes.

5.3 Proposed K-NN based Approach

This section describes the proposed algorithm based on K-NN. Let’s start with a brief background of K-NN and then move on to defining feature categories used in the proposed algorithm. Finally, the performance of the proposed K-NN approach for three open-access datasets is demonstrated and discussed.

5.3.1 K-NN Principles

K-NN is a time-series classification method where test samples are classified by a majority vote of neighbours via distance calculation between samples' attributes and corresponding features of instances in the training database using a distance metric [65]. Popular distance metrics include Euclidean, Manhattan, Hamming, DTW, etc. Let y be a test sample. Then, the distances between y and all samples in the training dataset x_1, \dots, x_K are calculated, and the minimum distance is found:

$$d_y = \min \{d(y, x_1); d(y, x_2); \dots; d(y, x_K)\}, \quad (5.4)$$

where $d(\cdot, \cdot)$ is a distance measure.

5.3.2 Feature Extraction

Let's assume, as in [31], that Individual appliance monitoring (IAM) measurements of individual loads are available for training. Note that if sub-metering data is not available, as it is the case for smart metering nationwide rollouts, time-wise features extracted from a time-of-use diary can be used to estimate magnitude-wise features from the aggregate load like in [28, 65].

In many countries, including Spain and Italy, smart meter measurements are collected every 24 hours. Therefore, daily disaggregation is performed on 24-hour long windows, where for hourly readings, each window contains $n = 24$ energy samples. Within each time sample, an appliance can be in either OFF state, if it was not running at all during that hour, or in the ON state, otherwise. Thus, for a duration of one window, there are 2^n possible daily combinations of ON and OFF states. To reduce complexity, the number of candidate ON-OFF state patterns is limited by filtering out invalid combinations based on appliance time usage profile, e.g., refrigerators are always ON, which is not the case with an electric heater.

Features are extracted as *a priori* inputs to K-NN, as in previous works [28, 31, 65], depending on the datasets and algorithms used. Note that $\mathbf{E_P}$ and $\mathbf{E_Q}$ are used to denote active and reactive power totally consumed, calculated through Method 2. From

$\mathbf{E_P}$, using the training dataset, the categories of features are intuitively designed as follows: 1. Average daily ON duration; 2. Maximum daily ON duration; 3. Minimum daily ON duration; 4. Average daily switched-ON time; 5. Average daily switched-OFF time; 6. Median time-of-day for daily running; 7. Average consumed energy per day; 8. Maximum consumed energy per day; 9. Minimum consumed energy per day; 10. Variance of consumed energy per day; 11. Average daily total energy consumed. When both energy and reactive power consumption measurements are available, for each appliance, an additional feature can be used – the average ratio between active and reactive measurements used in the ON state. Since only a subset of features is useful for disaggregating individual loads, an adaptive feature refining step is proposed based on the assumption that the subset of useful features should be extracted from attributes with high precision and low variability. RSD defined in Eq. (3.4) is used, as in Chapter 3, to represent the *quality* of each feature, where features are selected based on constant threshold G for evaluating RSD values. Note that additional features extracted from other attributes, such as weather and occupancy information used in [31], are available in the proposed algorithm, only if they result in small RSD .

Table 5.2 lists the selected features for several appliances from the AMPDs dataset using $G = 0.5$. The abbreviations used for domestic loads considered in this chapter are as follows: HWU is hot water unit; CW is clothes washer; DW is dishwasher; SNE is security/network equipment; HP is heat pump; UT is utility room; EWB is electronics workbench; GR is garage; FZ is freezer; KO is kitchen outlet; F is fridge and EH is electrical heater. From Table 5.2, the majority of features extracted from $\mathbf{E_P}$ for most appliances are of *high quality*, e.g., they have low RSD . On the other hand, the features extracted from $\mathbf{E_Q}$ for HWU and UT have low precision and high RSD , so are not used. Apart from daily total consumption, no feature can be extracted from $\mathbf{E_P}$ or $\mathbf{E_Q}$ for SNE due to its low-consumption, which is attributed to the baseload.

5.3.3 Feature Matching

Fig. 5.4 illustrates how d_y in Eq. (5.4) is calculated for the proposed K-NN based electricity profile disaggregation algorithm. During training, a set of aforementioned daily

Table 5.2: Features considered for appliances from the AMPds dataset.

	Features	HWU	CW	DW	SNE	HVAC	HP	UT	EWB	GR
E_P	Average daily ON duration	✓	✓	✓		✓		✓	✓	✓
	Maximum daily ON duration	✓	✓	✓		✓		✓	✓	✓
	Minimum daily ON duration	✓	✓	✓		✓		✓	✓	✓
	Average daily switched-ON time	✓						✓		
	Average daily switched-OFF time	✓				✓	✓	✓		
	Median time-of-day for daily running					✓	✓	✓		
	Average consumed energy per day	✓	✓	✓		✓	✓	✓	✓	✓
	Maximum consumed energy per day	✓	✓	✓			✓	✓	✓	✓
	Minimum consumed energy per day	✓	✓	✓		✓	✓	✓	✓	✓
	Variance of consumed energy per day	✓		✓			✓	✓	✓	✓
	Average daily total consumption	✓	✓	✓	✓	✓	✓	✓	✓	✓
E_Q	Average daily ON duration			✓					✓	✓
	Maximum daily ON duration			✓					✓	✓
	Minimum daily ON duration			✓					✓	✓
	Average daily switched-ON time									✓
	Average daily switched-OFF time					✓	✓			✓
	Median time-of-day for daily running					✓	✓			✓
	Average consumed energy per day		✓	✓		✓	✓		✓	✓
	Maximum consumed energy per day		✓	✓		✓	✓			✓
	Minimum consumed energy per day		✓	✓		✓	✓		✓	✓
	Variance of consumed energy per day			✓			✓			
Average daily total consumption		✓	✓	✓	✓	✓		✓	✓	
E_P & E_Q	Average ON state magnitude ratio between E_{Q_i} and E_{P_i}	✓	✓	✓		✓	✓	✓	✓	✓

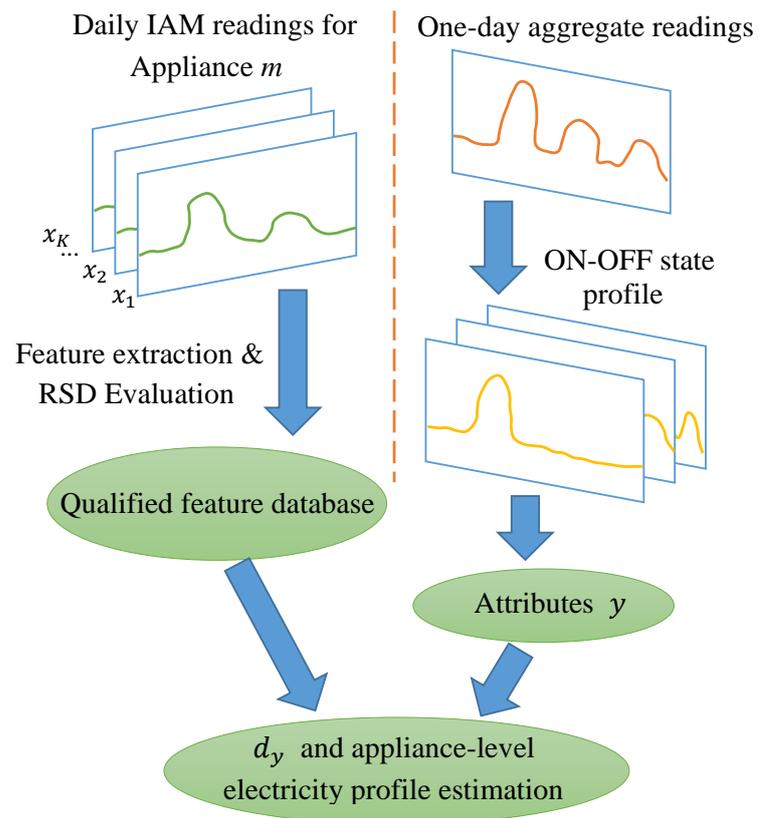


Figure 5.4: Flow chart of the proposed algorithm.

features is obtained, extracted and shortlisted (after *RSD* evaluation) for Appliance m , as discussed in Subsection 5.3.2 - left of Fig. 5.4. During testing (right of Fig. 5.4), the same features are extracted as those in the qualified feature database. Then K nearest neighbours are defined as K daily readings, x_1, \dots, x_K , from the training dataset, whose features have the shortest distances (calculated using Eq. (5.4) to the testing candidate ON-OFF state pattern y for Appliance m . For daily disaggregation and hourly sampling rates, y and x_i 's are all $n = 24$ -length vectors. The resulting minimum distance d_y among all possible candidates y and selected neighbours x classifies appliances as per the training set.

5.3.4 Experimental Validation and Discussion

The performance of the proposed approach is demonstrated using (1) \mathbf{P} of the US REDD dataset, (2) \mathbf{P} of the UK REFIT dataset (Cleaned) [61] and (3) both \mathbf{P} and \mathbf{Q} of Canadian AMPds dataset (Version 2) [66]. REDD dataset offers the best sub-metering coverage but provides the fewest measurements. REFIT houses include fewer submetered loads than AMPds and aggregate load measurements are considerably noisier than in the other two datasets, due to unknown appliances. Since the datasets' original measurements' sampling rates are 1sec, 6-8sec, and 1min, the 15-min or hourly energy $\mathbf{E_P}$ is calculated using Method 2. As time-of-day is a key feature in the proposed algorithm, only whole-day available measurements are used, e.g., if measurement samples are missing within a 24-hour window of a day, that day is not included in the experiments. Based on the aforementioned data selection rules and individual appliance usage frequency, the following portions of data are empirically selected: 19/04/2013–01/05/2013 (13 days in total, 7 days for training and 6 days for testing) for REDD House 2; 28/10/2013–02/06/2014 (90 days in total, 30 days for training and 60 days for testing) for REFIT House 1; 28/10/2012–27/04/2013 (180 days in total, 90 days for training and 90 days for testing) for the AMPds house. Fridge and freezer measurements in REFIT House 1 are merged. Aggregate measurements from the AMPds dataset includes the main house, garage and a rental suite. As is common practice by utilities for summarising energy use, kWh is used to present the results. In all

experiments, G for evaluating RSD is empirically set to 0.5.

Table 5.3: Performance of the proposed method for 15-minute data.

Appliance	REDD House 2			REFIT House 1		
	FZ	KO	BL	F+FZ	EH	BL
Est. (kWh)	11.6	1.47	3.08	61.67	209.16	151.98
IAM (kWh)	12.32	1.47	2.88	63.08	253.71	101.96
Est./Total	36.68%	4.65%	9.74%	7.18%	24.34%	17.69%

Tables 5.3 and 5.4 show the disaggregation results for 15-min and 1-hour electricity usage profile, respectively, for REDD House 2 Freezer (FZ), Kitchen Outlet (KO) and baseload (BL), and REFIT House 1 Fridge&Freezers (F+FZ), Electrical Heater (EH) and BL. IAM row, used as ground truth, shows the actual submetered energy in kWh. Est./Total shows the percentage contribution of the estimated energy consumed by an individual load towards the aggregate (measured) load.

The proposed approach can disaggregate 51% and 62.5%, for 15-min and 1-hour granularity, respectively, of REDD House 2 total load and about 49% and 54%, for 15-min and 1-hour granularity, respectively, of REFIT House 1 total load. The amount of energy consumed that can be accounted for due to individual loads is slightly lower for hourly granularity measurements. As the Est. and IAM rows for each appliance show, the disaggregated energy for each of the selected appliances is close to the actual energy consumed. The disaggregated energy for KO and BL for REDD House 2 is overestimated, which is inline with very low-rate disaggregation results based on sparse coding [32] where the performance of Fridge/Freezer is generally good but the overestimation problem generally exists for short-duration and low-energy loads.

Table 5.4: Performance of the proposed method for hourly data.

Appliance	REDD House 2			REFIT House 1		
	FZ	KO	BL	FZ	EH	BL
Est. (kWh)	9.52	1.6	8.67	43.53	235.62	197.42
IAM (kWh)	12.32	1.47	7.11	45.49	255.97	205.56
Est./Total	30.1%	5.07%	27.41%	4.96%	26.85%	22.5%

Table 5.5 shows the disaggregation results for hourly electricity usage profile for the AMPds house, where $+\mathbf{E}_Q$ refers to disaggregation with \mathbf{E}_Q available as a feature. When considering both \mathbf{E}_P and \mathbf{E}_Q instead of \mathbf{E}_P only, the algorithm performs worse

Table 5.5: Performance of the proposed method for the AMPds dataset for hourly data.

		HWU	CW	DW	SNE	HVAC
Est. (kWh)	$\mathbf{E_P}$	11.61	11.62	45.77	78.46	224.2
	$+\mathbf{E_Q}$	11.71	14.02	45.24	78.57	222.89
IAM (kWh)		16.85	10.23	36.78	86.97	252.11
Est./Total	$\mathbf{E_P}$	0.41%	0.41%	1.63%	2.79%	7.98%
	$+\mathbf{E_Q}$	0.42%	0.50%	1.61%	2.80%	7.94%
		HP	UT	EWB	GR	BL
Est. (kWh)	$\mathbf{E_P}$	456.77	108.95	57.34	3.29	212.74
	$+\mathbf{E_Q}$	477.74	108.95	57.34	3.29	212.74
IAM (kWh)		552.44	111.04	56.6	3.54	179.36
Est./Total	$\mathbf{E_P}$	16.26%	3.88%	2.04%	0.12%	7.58%
	$+\mathbf{E_Q}$	17.01%	3.88%	2.04%	0.12%	7.58%

for CW and better for HP, due to strong correlation between $\mathbf{E_P}$ and $\mathbf{E_Q}$ for HP, demonstrated in Fig. 5.5. The features derived from $\mathbf{E_Q}$ of CW have low *RSD* resulting

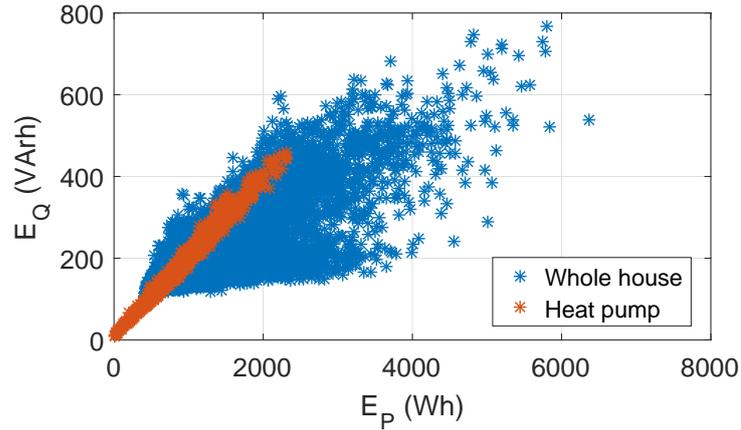


Figure 5.5: $\mathbf{E_Q}$ versus $\mathbf{E_P}$ for the same time.

in overestimation. The proposed approach can disaggregate 43% of AMPds house total electricity consumption given $\mathbf{E_P}$ and 44% given both $\mathbf{E_P}$ and $\mathbf{E_Q}$. Overall, with hourly NILM, the inclusion of $\mathbf{E_Q}$ as a feature does not seem to improve results significantly.

Then let's move on to the other two unsupervised approaches by first describing the proposed approach to solve very low-rate load profile disaggregation problem, based on OPT, then clarifying the novelty of GSP approach proposed for hourly disaggregation against the original version in Chapter 3.

5.4 Proposed Optimisation-based Approach (OPT)

Our proposed OPT approach does not require any training for the model. It requires no other information besides smart meter consumption readings E_i , and the appliance manufacturer information, i.e., typical energy consumption per operation cycle or 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, Eq. (2.1) is rewritten as

$$E_i = \sum_{m \in \mathcal{M}} E_i^m + \sum_{m \in \mathcal{N}} E_i^m + n_i, \quad (5.5)$$

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:

$$\arg \min_{\{E_i^m\}, m \in \mathcal{N}} \sum_{i=1}^N \left| \hat{E}_i - \sum_{m \in \mathcal{N}} E_i^m \right|, \quad (5.6)$$

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:

$$\hat{E}_i = E_i - \sum_{m \in \mathcal{M}} E_i^m. \quad (5.7)$$

OPT directly solves the optimisation problem in Eq. (5.6) after the always-on load profile is estimated and removed from aggregate profile signal as in Eq. (5.7).

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. (5.6) (Alg. 2), which are discussed in the following subsections.

5.4.1 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 [127], always-on loads are estimated by assuming that the electricity consumption between 12 AM and 5 AM 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 12 AM and 5 AM. E.g., for hourly sampling rate, τ is 5, as it corresponds to the 12 AM-5 AM 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.

Note that always-on load is not constant as it fluctuates over time (e.g., the refrigerator has cooling and stand-by stages, whose duration depends on its usage) containing local minima and maxima. Therefore the influence of rare appliance usage by residents during 12 AM-5 AM on always-on load removal is attempted to be mitigated. To do that, a threshold γ is set 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.

L_{max} and L_{min} are defined, 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 $E_i - L_{max} \leq \gamma$, it is estimated that only always-on appliances were operating during the sampling period i . Thus such load for always-on appliances is estimated, $\sum_{m \in \mathcal{M}} E_i^m$, as E_i , and therefore, from Eq. (5.7), $\hat{E}_i = 0$. Based on this assumption, if $|E_i - L_{max}| \leq |E_i - L_{min}|$, always-on load is defined 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,

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 $\sum_{m \in \mathcal{M}} E_i^m$ to L_{max} ;
- 16 **else**
- 17 $\sum_{m \in \mathcal{M}} E_i^m$ to L_{min} ;
- 18 **else if** $\sum_{m \in \mathcal{M}} E_{i-1}^m$ is set above to L_{min} **then**
- 19 $\sum_{m \in \mathcal{M}} E_i^m$ to L_{max} ;
- 20 **else**
- 21 $\sum_{m \in \mathcal{M}} E_i^m \leftarrow L_{min}$;
- 22 **calculate** $\hat{\mathbf{E}}$ using Eq. (5.7);
- 23 **return** $\sum_{m \in \mathcal{M}} \mathbf{E}^m$, $\hat{\mathbf{E}}$;

$$\hat{E}_i = E_i - L_{max} \text{ or } \hat{E}_i = E_i - L_{min}.$$

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, $\sum_{m \in \mathcal{M}} E_l^m$ is set 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.

5.4.2 Appliance Modelling

A model of each appliance $m \in \mathcal{N}$ is built using manufacturer information on appliance power rating. These appliances are split 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:

$$\hat{\mathbf{E}}^m = \begin{cases} \boldsymbol{\alpha}^m \bar{W}^m, & \text{for } m \in \mathcal{N}_1, \\ \boldsymbol{\beta}^m \bar{W}^m, & \text{for } m \in \mathcal{N}_2, \end{cases} \quad (5.8)$$

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 (5.8), the following example is given:

$$\hat{\mathbf{E}}^m = \left[\underbrace{\alpha_1^m, \alpha_2^m, \alpha_3^m}_{Run1}, \alpha_4^m, \alpha_5^m, \underbrace{\alpha_6^m, \alpha_7^m, \dots}_{Run2} \right] \times \bar{W}^m. \quad (5.9)$$

Eq. (5.9) indicates Appliance m runs two times within the selected period. Hence, based on the defined rules, $\alpha_1^m + \alpha_2^m + \alpha_3^m = 1$; $\alpha_6^m + \alpha_7^m = 1$ and $\alpha_4^m = \alpha_5^m = 0$. Then,

$$\hat{\mathbf{E}}^m = \left[\underbrace{\alpha_1^m \bar{W}^m, \alpha_2^m \bar{W}^m, \alpha_3^m \bar{W}^m}_{Run1}, 0, 0, \underbrace{\alpha_6^m \bar{W}^m, \alpha_7^m \bar{W}^m}_{Run2}, \dots \right], \quad (5.10)$$

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

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**
12 $\{\alpha^{m*}, \beta^{m*}\} = \arg \min_{\alpha^m, \beta^m} \left\| \hat{\mathbf{E}}_{seq} - \sum_{m \in \mathcal{N}_1} \alpha^m \bar{W}^m - \sum_{m \in \mathcal{N}_2} \beta^m \bar{W}^m \right\|$;
13 **subject to**
14 $\{ \sum_{i=1}^n \alpha_i^m \} = \mathbf{F}_m; 0 \leq \alpha_i^m \leq 1$;
15 **foreach** $m \in \mathcal{N}_2$ **do**
16 $\beta_i^m \in \{0, 1\}$;
17 $T_{opt}^* \leftarrow \left\| \hat{\mathbf{E}}_{seq} - \sum_{m \in \mathcal{N}_1} \alpha^{m*} \bar{W}^m - \sum_{m \in \mathcal{N}_2} \beta^{m*} \bar{W}^m \right\|$;
18 **if** $T_{opt}^* < T_{opt}$ **then**
19 $T_{opt} \leftarrow T_{opt}^*$; $\alpha_{opt}^m \leftarrow \alpha^{m*}$; $\beta_{opt}^m \leftarrow \beta^{m*}$;
20 **calculate** $\hat{\mathbf{E}}_{seq}^m$ using Eq. (5.8) with α_{opt}^m and β_{opt}^m ;
21 **return** $\hat{\mathbf{E}}^m$

5.4.3 Disaggregation via Optimisation

The OPT disaggregation algorithm is given in Alg. 2. The optimisation problem in (5.6) is NP-complete [44]. An approximate solution is found using CVX [128] 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 [129]. Constraints are heuristically set to reduce the number of candidates and optimisation complexity.

The entire sequence is split 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 the implementation $\sum_{i=1}^n \alpha_i^m$ is fixed, 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 N_0 length and contains as its m element the value of $\sum_{i=1}^n \alpha_i^m$.

Then Line 11, corresponding to Eq. (5.6), 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 5.4.2, 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 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. (5.8).

5.5 Graph Signal Processing - based Proposed Approach

This GSP-based unsupervised approach is built on the one proposed in Chapter 3 by generating an additional graph for the time-of-day feature, motivated by NILM

approaches [30, 99, 130] where time-of-day usage pattern is exploited.

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 $\mathbf{s}^* \leftarrow \mathbf{L}_{2:n,2:n}^\# (-s_1) \mathbf{L}_{1,2:n}^T \frown \mathbf{s}^*$;
- 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}

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 Gaussian kernel weighting function as 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 Chapter 3, based on total variation regularization upon graphs, where \frown notates the concatenation of two vectors.

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} ,
 ρ_E , ρ_T and q ;
- 5 $f \leftarrow f + 1$;
- 6 **foreach** \mathbf{C}_f **do**
- 7 **calculate** RSD_f using Eq. (3.4);
- 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 Chapter 3, 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 Chapter 3;
- 17 **return** $\hat{\mathbf{E}}^m$

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 by Eq. (3.4). For the rejected clusters with RSD_f higher than a heuristically set threshold T_{RSD} , the clustering is enhanced by halving ρ_E as in Chapter 3. Once ρ_E is halved the clustering is repeated until ρ_E becomes very small (regulated by a parameter κ close to zero). In Chapter 3, feature matching is required for pairing positive and negative resulting clusters and ON/OFF transition events in paired clusters since signal changes are used to build the graph. However, since hourly aggregate profile $\mathbf{E} \in \mathbb{R}^+ \cup \{0\}$ is used to build the graph in this paper, feature matching is not needed. Finally, as in Chapter 3, each disaggregated cluster is labelled as one category by comparing with appliance signature,

and corresponding appliance-level load profile is estimated. Note that time (not time-of-day) is used as a feature in Chapter 3, firstly for pairing ON/OFF events in feature matching, but not for clustering, and secondly, edges of the graph in Chapter 3 represent the correlation of duration among ON/OFF candidates, while in this paper each edge in graph \mathcal{G}^T represents the time-of-day difference of two corresponding samples.

5.6 Experimental Setup

5.6.1 Experimental Data

The REFIT dataset [61] is used for evaluation. This dataset was chosen for the following reasons: (i) supported by NILMTK and used in recent literature, [24, 121], to ease 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 unknown 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. (5.2). For all results presented, experiments were carried out for REFIT Houses 4 and 8, with a low-level and a higher-level of unknown appliance noise [19], 0.02 and 0.24, respectively. The experimental period of testing and training (where applicable) is presented in Table 5.6.

Table 5.6: 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)

5.6.2 Benchmark Setup

While the NILMTK toolbox [18] has an embedded resampling tool to generate hourly power samples by picking the last sample in each hour, it is not a true hourly load profile data and aggregated hourly consumption is lost. Instead, the hourly profile data is imported with corresponding timestamps into NILMTK for benchmarking against FHMM and CO. All parameters in DDSC implementation in this paper are chosen as suggested in [33].

5.6.3 Appliances Disaggregated

The following domestic loads are disaggregated and denoted in abbreviated form in the remainder 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.

5.7 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 5.6, and using the following evaluation metrics: daily Acc^m and daily MR^m .

5.7.1 OPT Parameters

For OPT, the sequence is split 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

Table 5.7: 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-consumed-per-run	760Wh	-	-	-	-

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 \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 5.7.

The parameters for the appliance models, defined in Section 5.4.2, are shown in Tables 5.8 and 5.9. As discussed in Section 5.4.2, 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 5.8: Appliance models generated by the proposed OPT 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 5.9: Appliance models generated by the proposed OPT 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

Where the energy consumption per cycle (Wh/cycle) is not available, it is possible to determine it by learning from the training data. 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 5.8 and 5.9.

As explained in Section 5.4.1, OPT estimates energy consumed by F, FZ, FFZ and baseload via always-on consumption estimation steps and therefore results show always-on loads 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.

5.7.2 GSP Parameters

The scaling factors for weighting graph edges: ρ_E is initialised to 10 and ρ_T is fixed to 0.005. κ is set to $\rho_E/2^{10}$. Trade-off factor λ is empirically set to 0.5. $T_{RSD} = 10\%$ and $q = 0.98$, as defined in Chapter 3, for maintaining high clustering quality. In order to mitigate long execution time, as reported in [100], the upper limit of window size in GSP clustering is set to 1344, which is equivalent to the period of 8 weeks for hourly profile measurements.

5.7.3 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.

5.7.4 Daily Consumption Accuracy and Match Rate Performance

Daily disaggregation accuracy Acc^m performance for Houses 4 and 8 are presented in Tables 5.10 and 5.11, respectively. Daily match rate MR^m results can be observed in Tables 5.12 and 5.13 for Houses 4 and 8, respectively. Note that weekly and monthly disaggregation accuracy Acc^m performance for Houses 4 and 8 are demonstrated in the Appendix.

DDSC performance is slightly worse than reported results in [32] but this is expected as results in Tables 5.10 to 5.13 were tested on houses containing many unknown

Table 5.10: Daily Acc^m results of the proposed methods with benchmarks 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	-	0.5	0.45	-	-	-	0.5	0.5
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 5.11: Daily Acc^m results of the proposed methods with benchmarks 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	-	-	0.5	0.36	-	-	-	0.4	-
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

appliances as opposed to the aggregate of known appliances in [32]. Since DDSC attempts to learn appliance dictionaries by mitigating the difference between the weekly profile and the production of dictionaries and appliance-level weekly activation, it is sensitive to the noise due to unknown appliances. The results explain why the sum of sub-metering measurements is used instead of real aggregate readings in [32], as discussed in Chapter 2. DDSC is also observed to have the worst performance of all benchmark and proposed algorithms.

Table 5.10 shows that FHMM performs well in estimating consumption accuracy for F, FZ and FFZ, which is in line with NILM results reported previously [16, 18]. However, in noisier House 8, as shown in Table 5.11, 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 5.10 and 5.11, 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, thus clusters labelled as TV contain a large amount of mis-clustered or unknown loads, resulting in over-estimation for TV but under-estimation for other appliances, as shown in Fig. 5.8. While in House 8, the usage frequencies for various appliances are more balanced. Hence, fewer events are falsely clustered to 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 to disaggregate such small loads, also shown in Figs. 5.8 and 5.9. However, FHMM and CO achieve good performance for PC in both houses as expected, in both metrics and Figs. 5.8 and 5.9.

As shown in Tables 5.10 and 5.11, CNN outperforms others in estimating daily consumption accuracy for always-on and long-lasting appliances (F, FZ, FFZ, TV, PC) in both houses. Especially for PC and TV that operate in a house-dependant hourly usage pattern, incorporating time information, CNN performs better than others under the supervision of sub-metering measurements.

It can be observed that WM and WD Acc^m performance is relatively poorer than other appliances. Similar daily match rate MR^m results can be observed from Tables 5.12 and 5.13, respectively. As defined in Subsection 5.4.2, 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 outperform on Acc^m and MR^m for WM in House 4, its energy consumption estimation is the closest to the energy actually consumed.

Table 5.12: Disaggregation daily MR^m results (%) of the proposed methods with benchmarks 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	16	0	10	9	14	15	0	0
GSP	80			36	0	29	21	34
OPT	87			26	49	56	49	54
CNN	86	90	87	41	55	64	50	55

Table 5.13: Disaggregation daily MR^m results (%) of the proposed methods with benchmarks 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	0	8	1	4	5	1	4
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

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 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 demonstrates a better metric to assess disaggregation performance as shown by Figs. 5.6 to 5.9, which show estimated energy consumed vs ground truth. E.g., for House 4, MR^m of GSP for PC is 0, inline with Fig. 5.8 - 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.

5.7.5 Analysis of Estimated vs. Actual Energy Consumption to explain Metrics

First, let's look at always-on appliances, as shown in Figs. 5.6 and 5.7, 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 disaggregation due to similarity in energy signature. The purpose of these figures, together with Figs. 5.8 and 5.9 which show energy estimated w.r.t ground truth for all other appliances is to better understand over- and under-estimation of consumption during disaggregation because this is not fully captured by the performance metrics.

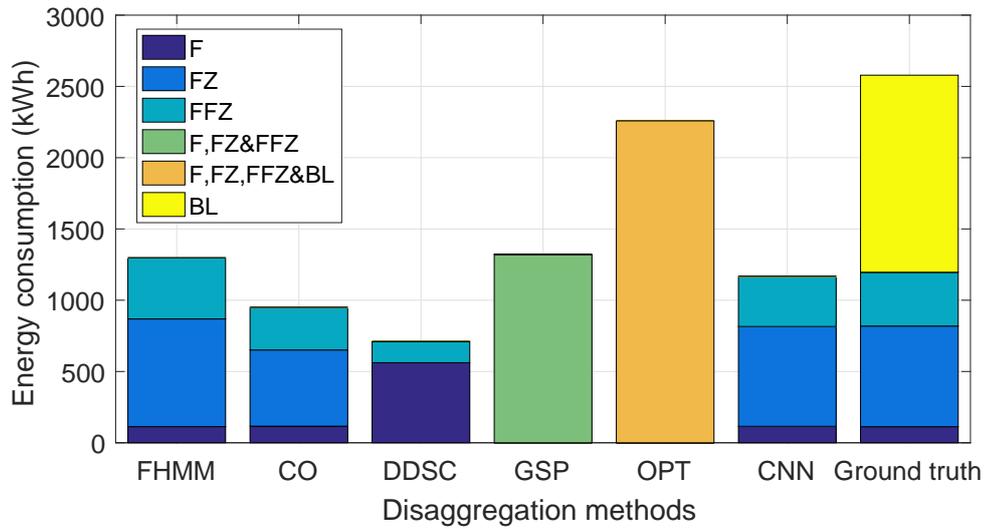


Figure 5.6: Disaggregated energy consumption of the proposed methods with benchmarks for always-on appliances in REFIT House 4.

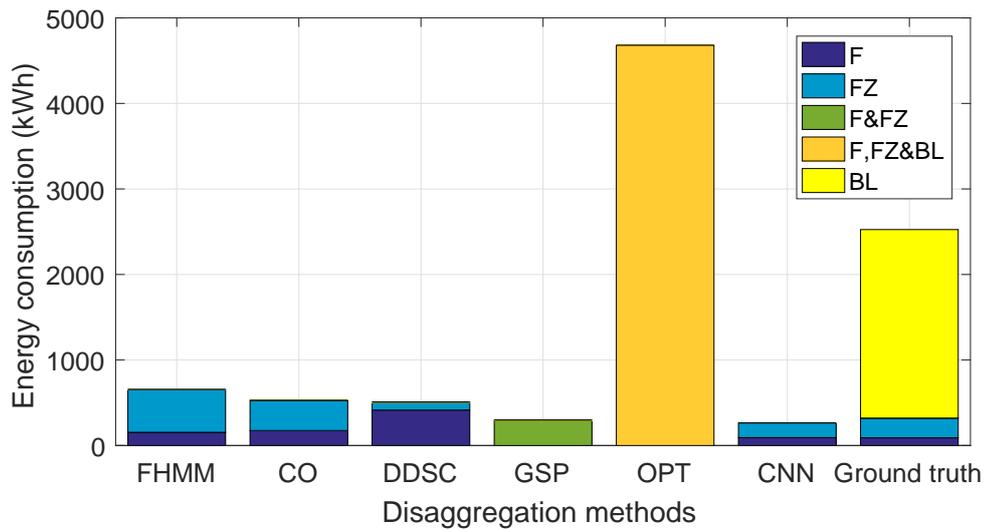


Figure 5.7: Disaggregated energy consumption of the proposed methods with benchmarks for always-on appliances in REFIT House 8.

DDSC performance is poor because it significantly overestimates freezer consumption as shown in Figs. 5.6 and 5.7. FHMM and CO generally slightly overestimate 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 over-estimates always-on loads for House 8 due to 'noise' from unknown appliances contributing significantly to BL at odd hours.

Next, the relative performance of all algorithms is discussed in terms of energy consumption of other appliances, namely washing machine, washer dryer, PC, TV, microwave, toaster and kettle, as shown in Figs. 5.8 and 5.9. In both houses, all benchmarking algorithms significantly over-estimate or under-estimate 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 unknown appliance models or a noise model results in over-estimation. This weakness is not obvious when the data is clean, i.e., where unknown loads do not exist as in [32, 100]. In particular, over-estimation is the most significant in CO performance of all approaches used, as its basis is ideal disaggregation as Eq. (2.1) with generally small measurement noise and all appliance being known. Inline with Acc^m and MR^m , the bar charts indicate DDSC is the most susceptible to real-world noise due to unknown loads among all benchmarks for both houses.

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 under-estimation of PC and M and over-estimation for TV, as shown in Fig. 5.8. 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 under-estimation.

OPT and CNN both generally perform best for all appliances. From Figs. 5.8 and 5.9, OPT performs better for House 4 while CNN performs better for House 8. For both houses, supervised CNN slightly outperforms unsupervised OPT for most appliances apart from M. Although in CNN the training set for M is sufficient due to plenty of M usage, the periodicity is overly learnt, resulting in over-estimation for M. Indeed, \hat{E}^m obtained by CNN for each appliance contains more or less periodic components, thus under-estimation is not observed in the CNN results.

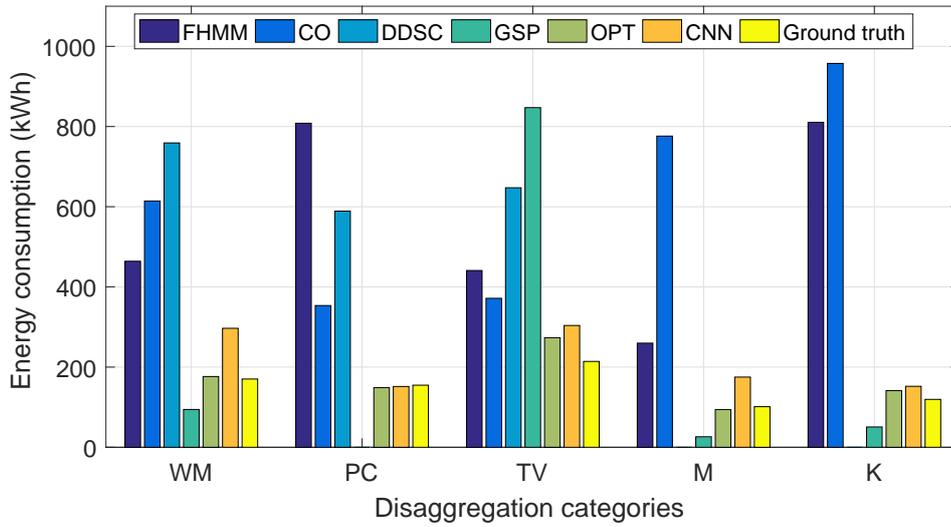


Figure 5.8: Disaggregated energy consumption of the proposed methods with benchmarks for other appliances in REFIT House 4.

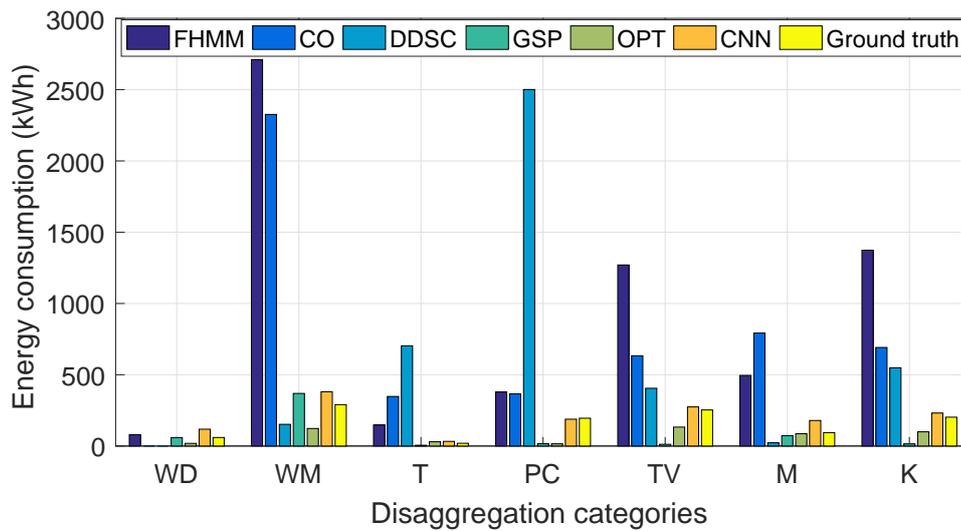


Figure 5.9: Disaggregated energy consumption of the proposed methods with benchmarks for other appliances in REFIT House 8.

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.

5.7.6 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 5.14: Execution time for all methods validated on both houses

		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 5.14 for all methods, where the runtime for FHMM and CO is inline with those reported in [100]. 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 [100], 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 [100]. Moreover, both OPT and CNN trade-off efficacy and efficiency, and execute within short periods of time.

5.8 Summary

This chapter contributes to the research gap of very low-rate energy profile disaggregation (15-60min) by proposing three disaggregation methods. First, a supervised K-NN based electricity usage profile disaggregation solution for daily appliance-level energy feedback is proposed. Unlike K-NN classifiers of [28, 65], appliance time usage profile is considered in the proposed K-NN method to extract useful features. Furthermore, *RSD* is used to evaluate the *quality* of each feature and customise feature selection

per appliance. After validation on three datasets for up to 3 months, It is shown that the proposed K-NN algorithm successfully disaggregates appliance energy consumption when compared to the individual, appliance-specific, energy measurements and can disaggregate up to 62% of the daily energy consumption from the total noisy electricity usage profile with 15-min and 60-min granularity.

Then two unsupervised approaches are proposed for hourly energy profile disaggregation, via OPT and GSP adapted from the GSP-based NILM approach proposed in Chapter 3. Both proposed unsupervised methods are validated on real-world noisy REFIT dataset together with four state-of-the-art power/energy disaggregation methods: FHMM and CO implemented in NILMTK [18], DDSC proposed in [32] and CNN proposed in Appendix 7.1. Disaggregation results for all methods are demonstrated in multiple metrics, instead of classification results. It can be observed from experimental results that most benchmarking methods are sensitive to noise and their performances drop significantly compared with those on more ideal cases. Generally, the proposed OPT outperforms other methods except for CNN, where manufacturer information is required in OPT instead of a training set of data. Although CNN performs better than OPT for most appliances, it requires plug-level training data.

Chapter 6

Conclusions and Future Work

This chapter brings summary and concluding remarks for the entire work presented in this thesis and perspective for future work.

6.1 Conclusions

Encouraged by the world-wide deployment of smart metering equipment, benefits of fine-grained energy feedback and weaknesses of prior NILM research, this thesis focuses on providing solutions for three challenging problems in the area of load disaggregation. Such problems refer to low-rate NILM (1-60sec sampling interval), result improving for general event-based low-rate NILM and very low-rate NILM (15-60min sampling interval).

Chapter 3 proposes a novel, *blind*, unsupervised low-rate NILM approach, built on the emerging GSP concepts and tested on REDD and REFIT datasets (residential active power readings from US and UK). The main motivation comes from the fact that GSP is training-free, can accurately capture signal patterns that occur rarely, is robust to noisy data, and has low implementation complexity. The results from disaggregating aggregate loads measured from four real houses indicate that the proposed training-less GSP-based NILM approach has comparable performance with the supervised GSP-based NILM approach of [40] outperforming the unsupervised HMM-based method. Moreover, the performance limits of the proposed GSP algorithm are heuris-

Conclusions and Future Work

tically determined and the usefulness of this limit is demonstrated to estimate the disaggregation performance. Due to its low complexity, simple operation and minimal customer input (for initial labelling), the algorithm can be applied on large scale as an embedded system as part of Consumer Access Device [49] with an online feedback interface that users can access. This paper has further demonstrated the potential of GSP for load disaggregation.

Then, for improving event-based low-rate NILM performance, in Chapter 4 various methods are designed before and after NILM procedure based on the nature of this problem, which objectives are reducing measurement noise in NILM entries and misclassification of loads with similar working power levels in NILM results, respectively. In order to address the challenge of mitigating the effect of measurement noise and unknown loads on NILM performance. Two signal processing methods based on GSP are proposed in conjunction with existing NILM approaches to improve any low-rate supervised and unsupervised event-based NILM classification and estimation accuracy. In particular, a graph-based filtering approach is proposed to clean the power signal before classification. The main motivation comes from the fact that event-based low-rate NILM approaches require clean power consumption measurements containing sharp and accurate state transition events. Besides the proposed graph-based filtering, a post-classification refinement method is proposed to improve NILM by mitigating the effect of misclassification of loads with similar operational range.

The improvement in NILM performance is demonstrated with the proposed methods when applied to three distinct event-based NILM methods and across real-world REDD and REFIT datasets with multiple houses with different levels of actual measured noise. The effect of sampling rate on graph filtering and edge sharpening is discussed and it is shown that the proposed method can significantly improve performance for smart meter data gathered at sampling rates of 1Hz and lower, allowing appliances which could not be detected by event-based NILM previously, possible.

Finally, the more challenging very low-rate load disaggregation problem is tackled where electricity consumption profile is collected every 15 minutes or longer in both supervised and unsupervised ways. A supervised K-NN based electricity usage profile

Conclusions and Future Work

disaggregation solution is proposed for daily appliance-level energy feedback. Unlike K-NN classifiers of [28, 65], appliance time usage profile is considered in the proposed K-NN method to extract useful features. Furthermore, *RSD* is used to evaluate the *quality* of each feature and customise feature selection per appliance. The validation on REDD, REFIT, AMPds datasets for up to 3 months indicates that the proposed algorithm successfully disaggregates appliance energy consumption when compared to the individual, appliance-specific, energy measurements and can disaggregate up to 62% of the daily energy consumption from the total noisy electricity usage profile with 15-min and 60-min granularity.

Opposite to the supervised K-NN approach, that is, avoiding usage of sub-metered measurements, two hourly energy profile disaggregation approaches are proposed, one performs this problem as combinatorial optimisation and the other is varied from the unsupervised approach for low-rate power disaggregation proposed in Chapter 3. Both proposed unsupervised methods are validated on real-world noisy REFIT dataset together with four state-of-the-art power/energy disaggregation methods: FHMM and CO implemented in NILMTK [18], DDSC proposed in [32] and CNN proposed in Appendix 7.1. Disaggregation results for all methods are demonstrated in multiple metrics, instead of classification results. It can be observed from experimental results that most benchmarking methods are sensitive to noise and their performances drop significantly compared with those on more ideal cases. Generally, the proposed OPT outperforms other methods except for CNN, where manufacturer information is required in OPT instead of a training set of data. Although CNN performs better than OPT for most appliances, it requires plug-level training data.

The scenarios applicable to the proposed methods are concluded in the following. In Chapter 3, the GSP-based approach is proposed as an unsupervised low-rate (at 1sec-1min granularity) NILM solution for the case where only low-rate aggregate measurement are available. That is, there is no plug-level training data, extra survey or environment data. However, GSP is susceptible to measurement noise and requires an extra labelling process after clustering.

The pre-processing approach proposed in Chapter 4 is applied before low-rate load

Conclusions and Future Work

disaggregation, for improving the performance by mitigating different types of noise present in the aggregate measurements. The performance improvement for the NILM methods making use of transient (power change) feature is more significant than that for other state-based NILM methods. Note that the pre-processing is not suitable for NILM approaches featuring high level of ‘fluctuation’, ‘transient spike’, etc., which is defined as ‘noise’ in the proposed pre-processing. The proposed NILM-result refining approach is applied as a further disaggregation step applied to the cluster of state transient events obtained by the primary load disaggregation method, where events of two appliances with similar working power ranges but distinguished running durations are grouped in the same cluster. The NILM-result refining approach requires either the plug-level data or the appliance model for at least one of the two appliances. Note that the methods proposed in Chapter 4 are not suitable for improving load disaggregation performance at very low rates (15min-1hour) due to the absence of state transient features.

For tackling the very low-rate (at 15min to 1hour granularity) energy disaggregation problem, three approaches are proposed in Chapter 5. The proposed K-NN requires appliance-level training data and extracts features from reactive energy consumption data if available. On the contrary, both proposed OPT and GSP-based methods do not require a set of data for training. However, make/model, rated power or energy-consumption-per-use for appliances are required for OPT via a one-off survey. The GSP-based method proposed for hourly energy disaggregation in Chapter 5 is low-complexity and requires pure aggregate hourly profile readings for billing purpose, but its disaggregation accuracy is lower than OPT and CNN.

6.2 Future Work

Although lots of works have contributed to the problems concerned in this PhD work, there are still potentials of improvement in various ways, worthy being researched more deeply and extensively. Future work is listed in the following.

- The robustness of current low-rate NILM algorithm via unsupervised GSP can

Conclusions and Future Work

be enhanced as it still can not separate simultaneous state transitions of multiple appliances. Besides, the performance of the proposed GSP for multi-state appliances can also be improved, by, e.g., utilising extra multi-state appliance models generated through transfer learning.

- The features of time used in proposed K-NN can be used in low-rate NILM algorithm based on unsupervised GSP for generating extra graphs, where trade-off factors as weights of each feature can be learnt if the training set is available.
- Weights in the adjacency matrix of graphs can be selected based on training using CNN instead of Gaussian kernel weighting function.
- For NILM result improving methods, adaptive parameter selection derived from measurements can be investigated and robustness can be enhanced by applying the proposed algorithm to other energy usage measurements (e.g., reactive power) with various sampling rates.
- For the supervised hourly profile disaggregation via K-NN, future work includes weighting features based on *RSD* or other metrics; improvement of ON-OFF states prediction rules to trade-off efficiency; widening the set of loads that can be estimated reliably; transfer learning from similar houses and appliance ownership and usage profile.
- Since permanent noise modelling is proposed in OPT, it can be combined with other benchmarks and expected to reduce the complexity of further disaggregation.
- An investigation is valuable for analysing benchmarks' NILM performance on energy measurements at a distinct sampling rate, from 1 sec to 1 hour.
- Investigating scalable or transferable NILM approaches, comprising unsupervised and supervised methods, that can work with minimal training data and can work on any unseen dataset.

Conclusions and Future Work

- For the proposed OPT, it would be beneficial to undertake a sensitivity analysis to assess the effect of the noise level in the house, i.e., the percentage of energy consumed by unlabelled appliances, on disaggregation performance.
- The efficiency of the proposed OPT should be systematically improved as it is currently exponential in the number of appliances, with heuristically set constraints to reduce the computational time.

Chapter 7

Appendix

7.1 Convolution Neural Network

In this section, the CNN architecture used for benchmarking in Chapter 5 is introduced. The CNN architecture is motivated by recent deep neural network architectures [7], [37] and [39] for low-rate power disaggregation and implemented by a co-author Minxiang Ye. It can be observed from Table 2.1 that all neural networks in [7,37,39] are proposed for load disaggregation on the smart meter readings of resolution from 1 second to 1 minute. However, the CNN architecture applied in Chapter 5 is proposed for estimating appliance hourly profile after training on very low-rate aggregate profile data. To enhance disaggregation performance on the hourly profile, time-of-day information is explicitly used as an additional feature, which differs from the aforementioned neural network architectures for load disaggregation.

As shown in Fig. 7.1, the 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 hours sliding window; (2) encoded cyclical continuous absolute time features of the fourth hour in the window. In this paper, he heuristically sets window length $Q = 7$ hours to cover at least two working periods of a washing machine. Given a sequence of 7 hours' aggregate power consumption, he employs 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

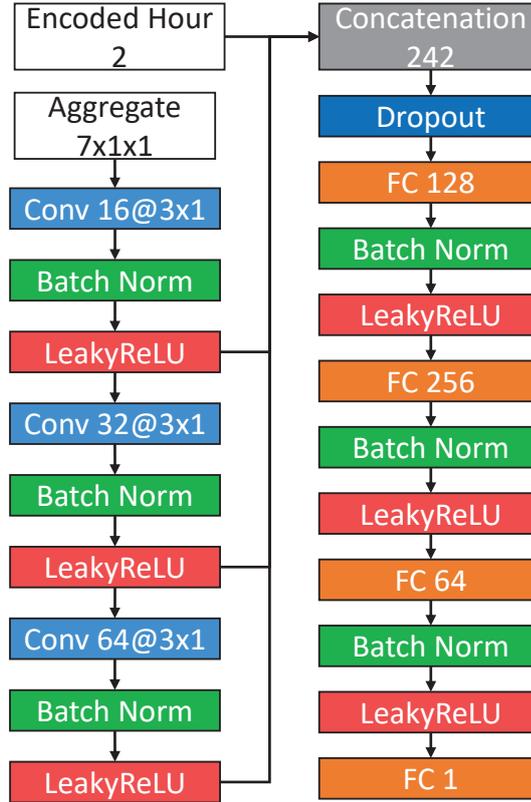


Figure 7.1: Proposed CNN architecture for very low-rate disaggregation. Both windows of 7-hour long aggregate data and the 4-th hour’s encoded time information are fed into the proposed network. ‘Conv $x@y$ ’ refers to a 2D convolutional layer with x filters each with kernel size y and fixed stride size= 1×1 . ‘FC x ’ means the fully connected layer with x =number of neurons. ‘Concatenation x ’ refers to a flatten operation that reshapes a matrix into a vector array with x values. ‘Batch Norm’ is a batch normalisation layer [131]. ‘LeakyReLU’ refers to a leaky version of a rectified linear unit activation function [132], with a fixed slope coefficient= 0.3 . ‘Dropout x ’ means that the dropout layer [133] randomly sets a fraction rate= x of input units to zero at each update during the training phase.

distribution of appliance usage, he encodes 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 both time features into the network at the first layer, he merges the target hour’s time features with the corresponding spatial features extracted by the 3 CNN blocks via down-sampling in the time domain. To effectively train the network, “skip connection” is adopted to merge the outputs of the 3 CNN blocks with the time features that accelerates the

training, enabling an ensemble of 7, 5, 3 hours' 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, he also adopt batch normalization [131], dropout [133], leaky ReLU [132] and l_2 -norm penalty on weights [134] to obtain better regularization performance regardless of weights initialisation.

From the experimental results in Section 5.7, the CNN outperforms other methods for most appliances. CNN is especially good at estimating always-on and long-lasting appliances, that operate in a house-dependant hourly usage pattern, because it incorporates time-of-day information. Note that comparing with unsupervised methods such as OPT, supervised CNN requires sufficient data for training. Thus, if the training data is sufficient, CNN is the best performing solution for very low-rate load disaggregation among all NILM approaches validated in Chapter 5. The difference of usage periodicity for one appliance between its training period and testing period might lead to overestimation or underestimation. For example, the reduction of microwave usage results in overestimation for its disaggregated consumption, as its usage frequency is overly learnt. Since periodic components exist for most domestic appliance usage, the utilisation of time-of-day feature in CNN improves the very low-rate load disaggregation performance for most appliances.

7.2 Weekly and Monthly Acc^m Results for REFIT House 4 and 8 on Hourly Granularity

In this section, more experimental results are demonstrated in various metrics for enriching the results presented in Section 5.7. The disaggregation accuracy Acc^m for all proposed and benchmarking methods at lower granularities, as per week and per month, are shown in Tables 7.1 and 7.2 for REFIT House 4 and in Tables 7.3 and 7.4 for REFIT House 8. The weekly and monthly disaggregation accuracy Acc^m is calculated in a similar way as Eq. 2.22, however, weekly/monthly load profile is utilised instead of the daily load profile.

Appendix

Table 7.1: Weekly Acc^m results of the proposed methods with benchmarks for REFIT House 4.

App.	F	FZ	FFZ	WM	PC	TV	M	K
FHMM	0.93	0.92	0.87	0.57	-	0.53	0.3	-
CO	0.96	0.88	0.89	-	0.36	0.63	-	-
DDSC	-	0.5	0.45	-	-	-	0.5	0.5
GSP	0.88			0.72	0.5	-	0.61	0.71
OPT	0.94			0.8	0.77	0.74	0.79	0.8
CNN	0.95	0.95	0.94	0.62	0.8	0.77	0.63	0.8

Table 7.2: Monthly Acc^m results of the proposed methods with benchmarks for REFIT House 4.

App.	F	FZ	FFZ	WM	PC	TV	M	K
FHMM	0.94	0.92	0.89	0.79	-	0.54	0.3	-
CO	0.97	0.88	0.9	-	0.37	0.64	-	-
DDSC	-	0.5	0.59	-	-	0.03	0.5	0.5
GSP	0.89			0.77	0.5	-	0.61	0.71
OPT	0.94			0.91	0.82	0.79	0.85	0.86
CNN	0.96	0.96	0.94	0.63	0.84	0.79	0.63	0.84

Table 7.3: Weekly Acc^m results of the proposed methods with benchmarks for REFIT House 8.

App.	F	FZ	WD	WM	T	PC	TV	M	K
FHMM	0.75	0.56	0.7	-	-	0.67	-	-	-
CO	0.53	0.72	0.5	-	-	0.56	0.25	-	-
DDSC	-	-	0.5	0.36	-	-	-	0.4	-
GSP	0.86		0.73	0.81	0.63	0.54	0.52	0.87	0.54
OPT	0.65		0.63	0.71	0.71	0.54	0.76	0.89	0.75
CNN	0.91	0.86	0.42	0.79	0.62	0.94	0.91	0.55	0.92

Table 7.4: Monthly Acc^m results of the proposed methods with benchmarks for REFIT House 8.

App.	F	FZ	WD	WM	T	PC	TV	M	K
FHMM	0.78	0.56	0.84	-	-	0.69	-	-	-
CO	0.53	0.73	0.5	-	-	0.57	0.24	-	-
DDSC	-	-	0.5	0.51	-	-	0.34	0.46	-
GSP	0.87		0.85	0.84	0.64	0.54	0.52	0.89	0.54
OPT	0.65		0.65	0.71	0.73	0.54	0.76	0.93	0.75
CNN	0.93	0.87	0.55	0.82	0.66	0.95	0.93	0.54	0.93

Bibliography

Comparing the daily accuracy results and corresponding weekly and monthly results for both houses, it can be observed that the disaggregation accuracy at a lower granularity is always better than that at a higher granularity. Therefore, if the hourly load disaggregation results are unnecessary for the consumers, the energy disaggregation feedback can be provided per week or per month with higher accuracy.

Bibliography

- [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/overview>, July 2014, updated Jan. 20, 2020.
- [3] K. C. Armel, A. Gupta, G. Shrimali, and A. Albert, “Is disaggregation the holy grail of energy efficiency? the case of electricity,” *Energy Policy*, vol. 52, pp. 213–234, 2013.
- [4] P. Xu, J. Shen, X. Zhang, X. Zhao, and Y. Qian, “Case study of smart meter and in-home display for residential behavior change in shanghai, china,” *Energy Procedia*, vol. 75, pp. 2694–2699, 2015.
- [5] G. Hart, M. I. of Technology. Energy Laboratory, and E. P. R. Institute, *Nonintrusive Appliance Load Data Acquisition Method: Progress Report*. MIT Energy Laboratory, 1984. [Online]. Available: <https://books.google.co.uk/books?id=gYIYtwAACAAJ>
- [6] 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.
- [7] 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.

Bibliography

- [8] 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.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] H. Rashid, P. Singh, V. Stankovic, and L. Stankovic, "Can non-intrusive load monitoring be used for identifying an appliance's anomalous behaviour?" *Applied Energy*, vol. 238, pp. 796–805, 2019.
- [14] L. Stankovic, C. Wilson, J. Liao, V. Stankovic, R. Hauxwell-Baldwin, D. Murray, and M. Coleman, "Understanding domestic appliance use through their linkages to common activities," in *8th international conference on energy efficiency in domestic appliances and lighting*, 2015.
- [15] A. Zoha, A. Gluhak, M. Imran, and S. Rajasegarar, "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey," *Sensors*, vol. 12, no. 12, pp. 16 838–16 866, 2012.

Bibliography

- [16] 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.
- [17] R. Bonfigli, S. Squartini, M. Fagiani, and F. Piazza, “Unsupervised algorithms for non-intrusive load monitoring: An up-to-date overview,” in *2015 IEEE 15th International Conference on Environment and Electrical Engineering (EEEIC)*. IEEE, 2015, pp. 1175–1180.
- [18] 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.
- [19] 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.
- [20] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han, “Unsupervised disaggregation of low frequency power measurements,” in *Proceedings of the 2011 SIAM international conference on data mining*. SIAM, 2011, pp. 747–758.
- [21] J. Z. Kolter and T. Jaakkola, “Approximate inference in additive factorial hmms with application to energy disaggregation,” in *Artificial intelligence and statistics*, 2012, pp. 1472–1482.
- [22] O. Parson, S. Ghosh, M. Weal, and A. Rogers, “Non-intrusive load monitoring using prior models of general appliance types,” in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
- [23] M. Zhong, N. Goddard, and C. Sutton, “Latent bayesian melding for integrating individual and population models,” in *Advances in neural information processing systems*, 2015, pp. 3618–3626.

Bibliography

- [24] 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.
- [25] M. Aiad and P. H. Lee, “Unsupervised approach for load disaggregation with devices interactions,” *Energy and Buildings*, vol. 116, pp. 96–103, 2016.
- [26] K. Basu, V. Debusschere, S. Bacha, U. Maulik, and S. Bondyopadhyay, “Non-intrusive load monitoring: A temporal multilabel classification approach,” *IEEE Transactions on industrial informatics*, vol. 11, no. 1, pp. 262–270, 2015.
- [27] 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 (SmartGridComm)*. IEEE, 2014, pp. 535–540.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] 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.

Bibliography

- [32] 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.
- [33] 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.
- [34] H. Shao, M. Marwah, and N. Ramakrishnan, “A temporal motif mining approach to unsupervised energy disaggregation: Applications to residential and commercial buildings,” in *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.
- [35] A. G. Ruzzelli, C. Nicolas, A. Schoofs, and G. M. O’Hare, “Real-time recognition and profiling of appliances through a single electricity sensor,” in *2010 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*. IEEE, 2010, pp. 1–9.
- [36] S. Biansoongnern and B. Plangklang, “Nonintrusive load monitoring (nilm) using an artificial neural network in embedded system with low sampling rate,” in *2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. IEEE, 2016, pp. 1–4.
- [37] 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.
- [38] 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.

Bibliography

- [39] 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.
- [40] 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.
- [41] 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.
- [42] 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.
- [43] G. Tang, Z. Ling, F. Li, D. Tang, and J. Tang, “Occupancy-aided energy disaggregation,” *Computer Networks*, vol. 117, pp. 42–51, 2017.
- [44] G. W. Hart, “Nonintrusive appliance load monitoring,” *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.
- [45] 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.
- [46] 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.
- [47] 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.

Bibliography

- [48] 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.
- [49] D. of Energy and C. Change, “Smart metering equipment technical specifications version 2,” 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.
- [50] M. Weiss, A. Helfenstein, F. Mattern, and T. Staake, “Leveraging smart meter data to recognize home appliances,” in *2012 IEEE International Conference on Pervasive Computing and Communications*. IEEE, 2012, pp. 190–197.
- [51] 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.
- [52] 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.
- [53] 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.
- [54] 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.
- [55] V. Y. Pillitteri and T. L. Brewer, “Guidelines for smart grid cybersecurity,” NIST Interagency, Tech. Rep. (NISTIR)-7628 Rev 1, 2014.

Bibliography

- [56] K. Kumar, R. Sinha, M. G. Chandra, and N. K. Thokala, “Data-driven electrical load disaggregation using graph signal processing,” in *2016 IEEE Annual India Conference (INDICON)*. IEEE, 2016, pp. 1–6.
- [57] C. Zhang, D. Florêncio, and P. A. Chou, “Graph signal processing-a probabilistic framework,” *Microsoft Res., Redmond, WA, USA, Tech. Rep. MSR-TR-2015-31*, 2015.
- [58] M. J. Johnson and A. S. Willsky, “Bayesian nonparametric hidden semi-markov models,” *Journal of Machine Learning Research*, vol. 14, no. Feb, pp. 673–701, 2013.
- [59] O. Parson, S. Ghosh, M. Weal, and A. Rogers, “An unsupervised training method for non-intrusive appliance load monitoring,” *Artificial Intelligence*, vol. 217, pp. 1–19, 2014.
- [60] 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.
- [61] 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.
- [62] S. Chen, A. Sandryhaila, J. M. Moura, and J. Kovacevic, “Signal denoising on graphs via graph filtering,” in *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2014, pp. 872–876.
- [63] A. Sandryhaila and J. M. Moura, “Discrete signal processing on graphs,” *IEEE transactions on signal processing*, vol. 61, no. 7, pp. 1644–1656, 2013.
- [64] A. Gadde, S. K. Narang, and A. Ortega, “Bilateral filter: Graph spectral interpretation and extensions,” in *2013 IEEE International Conference on Image Processing*. IEEE, 2013, pp. 1222–1226.

Bibliography

- [65] K. Basu, V. Debusschere, A. Douzal-Chouakria, and S. Bacha, “Time series distance-based methods for non-intrusive load monitoring in residential buildings,” *Energy and Buildings*, vol. 96, pp. 109–117, 2015.
- [66] 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.
- [67] M. Dong, P. C. Meira, W. Xu, and W. Freitas, “An event window based load monitoring technique for smart meters,” *IEEE Transactions on Smart Grid*, vol. 3, no. 2, pp. 787–796, 2012.
- [68] A. S. Bouhouras, P. A. Gkaidatzis, E. Panagiotou, N. Poulakis, and G. C. Christoforidis, “A nilm algorithm with enhanced disaggregation scheme under harmonic current vectors,” *Energy and Buildings*, vol. 183, pp. 392–407, 2019.
- [69] E. P. Loukas, K. Bodurri, P. Evangelopoulos, A. S. Bouhouras, N. Poulakis, G. C. Christoforidis, I. Panapakidis, and K. C. Chatzisavvas, “A machine learning approach for nilm based on odd harmonic current vectors,” in *2019 8th International Conference on Modern Power Systems (MPS)*. IEEE, 2019, pp. 1–6.
- [70] N. Batra, R. Baijal, A. Singh, and K. Whitehouse, “How good is good enough? re-evaluating the bar for energy disaggregation,” *arXiv preprint arXiv:1510.08713*, 2015.
- [71] N. Henao, K. Agbossou, S. Kelouwani, Y. Dubé, and M. Fournier, “Approach in nonintrusive type i load monitoring using subtractive clustering,” *IEEE Transactions on Smart Grid*, vol. 8, no. 2, pp. 812–821, 2017.
- [72] D. Egarter and W. Elmenreich, “Autonomous load disaggregation approach based on active power measurements,” in *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. IEEE, 2015, pp. 293–298.

Bibliography

- [73] A. Cominola, M. Giuliani, D. Piga, A. Castelletti, and A. E. Rizzoli, “A hybrid signature-based iterative disaggregation algorithm for non-intrusive load monitoring,” *Applied energy*, vol. 185, pp. 331–344, 2017.
- [74] N. Pathak, N. Roy, and A. Biswas, “Iterative signal separation assisted energy disaggregation,” in *2015 Sixth International Green and Sustainable Computing Conference (IGSC)*. IEEE, 2015, pp. 1–8.
- [75] F. Paradiso, F. Paganelli, D. Giuli, and S. Capobianco, “Context-based energy disaggregation in smart homes,” *Future Internet*, vol. 8, no. 1, p. 4, 2016.
- [76] A. Faustine, N. H. Mvungi, S. Kaijage, and K. Michael, “A survey on non-intrusive load monitoring methodologies and techniques for energy disaggregation problem,” *arXiv preprint arXiv:1703.00785*, 2017.
- [77] S. Kang and J. W. Yoon, “Classification of home appliance by using probabilistic knn with sensor data,” in *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, 2016, pp. 1–5.
- [78] S. Singh and A. Majumdar, “Non-intrusive load monitoring via multi-label sparse representation based classification,” *IEEE Transactions on Smart Grid*, 2019.
- [79] P. Xiao and S. Cheng, “Neural network for nilm based on operational state change classification,” *arXiv preprint arXiv:1902.02675*, 2019.
- [80] M. Zeifman and K. Roth, “Nonintrusive appliance load monitoring: Review and outlook,” *IEEE transactions on Consumer Electronics*, vol. 57, no. 1, pp. 76–84, 2011.
- [81] L. Pereira and N. Nunes, “Performance evaluation in non-intrusive load monitoring: Datasets, metrics, and tools—a review,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 6, pp. 1–17, 2018.
- [82] F. K.-K. Kee, Y. S. Lim, J. Wong, and K. H. Chua, “Non-intrusive load monitoring (nilm)—a recent review with cloud computing,” in *2019 IEEE International*

Bibliography

- Conference on Smart Instrumentation, Measurement and Application (ICSIMA)*.
IEEE, 2019, pp. 1–6.
- [83] M. Zhong, N. Goddard, and C. Sutton, “Signal aggregate constraints in additive factorial hmms, with application to energy disaggregation,” in *Advances in Neural Information Processing Systems*, 2014, pp. 3590–3598.
- [84] S. Patten, “Unsupervised disaggregation for non-intrusive load monitoring,” in *2012 11th International Conference on Machine Learning and Applications*, vol. 2. IEEE, 2012, pp. 515–520.
- [85] K. Anderson, J. M. Moura, and M. Berges, “Unsupervised approximate power trace decomposition algorithm,” in *Proc. 2nd Int. Non-Intrusive Appliance Load Monitor. Workshop*, 2014, pp. 5–8.
- [86] M. Devlin and B. P. Hayes, “Non-intrusive load monitoring using electricity smart meter data: A deep learning approach,” in *2019 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 2019, pp. 1–5.
- [87] C. Liu, A. Akintayo, Z. Jiang, G. P. Henze, and S. Sarkar, “Multivariate exploration of non-intrusive load monitoring via spatiotemporal pattern network,” *Applied energy*, vol. 211, pp. 1106–1122, 2018.
- [88] N. Batra, A. Singh, and K. Whitehouse, “Neighbourhood nilm: A big-data approach to household energy disaggregation,” *arXiv preprint arXiv:1511.02900*, 2015.
- [89] B. Wild, K. S. Barsim, and B. Yang, “A new unsupervised event detector for non-intrusive load monitoring,” in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2015, pp. 73–77.
- [90] H. Gonçalves, A. Ocneanu, M. Bergés, and R. Fan, “Unsupervised disaggregation of appliances using aggregated consumption data,” in *The 1st KDD Workshop on Data Mining Applications in Sustainability (SustKDD)*, 2011.

Bibliography

- [91] D. García, I. Díaz, D. Pérez, A. A. Cuadrado, M. Domínguez, and A. Morán, “Interactive visualization for nilm in large buildings using non-negative matrix factorization,” *Energy and Buildings*, vol. 176, pp. 95–108, 2018.
- [92] S. Barker, S. Kalra, D. Irwin, and P. Shenoy, “Nilm redux: The case for emphasizing applications over accuracy,” in *NILM-2014 workshop*. Citeseer, 2014.
- [93] A. Dejamkhooy, A. Ahmadpour, and S. Pourjafar, “Non-intrusive appliance load disaggregation in smart homes using hybrid constrained particle swarm optimization and factorial hidden markov model,” *Journal of Energy Management and Technology*, vol. 3, no. 4, pp. 52–64, 2019.
- [94] K. Osathanunkul and K. Osathanunkul, “Different sampling rates on neural nilm energy disaggregation,” in *2019 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT-NCON)*. IEEE, 2019, pp. 318–321.
- [95] 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.
- [96] N. MacMackin, L. Miller, and R. Cariveau, “Modeling and disaggregating hourly effects of weather on sectoral electricity demand,” *Energy*, vol. 188, p. 115956, 2019.
- [97] 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.
- [98] 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.

Bibliography

- [99] 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.
- [100] 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.
- [101] 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.
- [102] A. M. Alonso, F. J. Nogales, and C. Ruiz, “A single scalable lstm model for short-term forecasting of disaggregated electricity loads,” *arXiv preprint arXiv:1910.06640*, 2019.
- [103] 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 (Smart-GridComm)*. IEEE, 2014, pp. 511–516.
- [104] 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.
- [105] 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.
- [106] C. C. Yang, C. S. Soh, and V. V. Yap, “A systematic approach to on-off event detection and clustering analysis of non-intrusive appliance load monitoring,” *Frontiers in Energy*, vol. 9, no. 2, pp. 231–237, 2015.

Bibliography

- [107] H. Araghi, M. Sabbaqi, and M. Babaie-Zadeh, “ k -graphs: An algorithm for graph signal clustering and multiple graph learning,” *IEEE Signal Processing Letters*, vol. 26, no. 10, pp. 1486–1490, 2019.
- [108] A. Sandryhaila and J. M. Moura, “Classification via regularization on graphs,” in *2013 IEEE global conference on signal and information processing*. IEEE, 2013, pp. 495–498.
- [109] C. Yang, Y. Mao, G. Cheung, V. Stankovic, and K. Chan, “Graph-based depth video denoising and event detection for sleep monitoring,” in *2014 IEEE 16th international workshop on multimedia signal processing (MMSP)*. IEEE, 2014, pp. 1–6.
- [110] J. Pang, G. Cheung, A. Ortega, and O. C. Au, “Optimal graph laplacian regularization for natural image denoising,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 2294–2298.
- [111] C. Yang, A. Kerr, V. Stankovic, L. Stankovic, P. Rowe, and S. Cheng, “Human upper limb motion analysis for post-stroke impairment assessment using video analytics,” *IEEE Access*, vol. 4, pp. 650–659, 2016.
- [112] A. Sandryhaila and J. M. Moura, “Discrete signal processing on graphs: Frequency analysis,” *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, 2014.
- [113] A. Sandryhaila and J. M. Moura, “Classification via regularization on graphs,” in *2013 IEEE Global Conference on Signal and Information Processing*. IEEE, 2013, pp. 495–498.
- [114] C. Yang, G. Cheung, and V. Stankovic, “Estimating heart rate via depth video motion tracking,” in *2015 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2015, pp. 1–6.
- [115] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

Bibliography

- [116] B. Huang, M. Knox, K. Bradbury, L. M. Collins, and R. G. Newell, “Non-intrusive load monitoring system performance over a range of low frequency sampling rates,” in *2017 IEEE 6th International Conference on Renewable Energy Research and Applications (ICRERA)*. IEEE, 2017, pp. 505–509.
- [117] 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.
- [118] S. Arora and B. Barak, *Computational complexity: a modern approach*. Cambridge University Press, 2009.
- [119] J. Gao, S. Giri, E. C. Kara, and M. Bergés, “Plaid: a public dataset of high-resolution electrical appliance measurements for load identification research: demo abstract,” in *proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*. ACM, 2014, pp. 198–199.
- [120] S. Giri and M. Bergés, “An energy estimation framework for event-based methods in non-intrusive load monitoring,” *Energy Conversion and Management*, vol. 90, pp. 488–498, 2015.
- [121] 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.
- [122] D. Egarter, M. Pöchacker, and W. Elmenreich, “Complexity of power draws for load disaggregation,” in *5th DA-CH+ Energy Informatics Conference in conjunction with 7th Symposium on Communications for Energy Systems (ComForEn)*, 2016, p. 81.
- [123] 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.

Bibliography

- [124] B. Zhao, M. Ye, L. Stankovic, and V. Stankovic, “Non-intrusive load disaggregation solutions for very low-rate smart meter data,” *Applied Energy*, 4 2020.
- [125] M. Wytock and J. Z. Kolter, “Contextually supervised source separation with application to energy disaggregation,” in *Twenty-eighth AAAI conference on artificial intelligence*, 2014.
- [126] M. Simonov, “Coarse-grained cycle-accurate electricity metering,” in *IEEE PES Innovative Smart Grid Technologies, Europe*. IEEE, 2014, pp. 1–5.
- [127] 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.
- [128] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 2.1,” <http://cvxr.com/cvx>, Mar. 2014.
- [129] P. Tseng, “An infeasible path-following method for monotone complementarity problems,” *SIAM Journal on Optimization*, vol. 7, no. 2, pp. 386–402, 1997.
- [130] 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.
- [131] 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.
- [132] 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.
- [133] 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.

Bibliography

- [134] P. Lemberger, “On generalization and regularization in deep learning,”
arXiv:1704.01312, 2017.