People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

Ecole Nationale Polytechnique
Department of Electronics
Laboratory of Communication Devices
and Photovoltaic conversion

PhD Thesis

In Electronics

Presented by:

Selim Sahrane

Study of the realization of a Non-Intrusive Load Monitoring System

Thesis defended on December the 1st before the jury composed of:

A. BELOUCHRANI	Professor, ENP	President
M. HADDADI	Professor, ENP	Advisor
M. ADNANE	Professor, ENP	Co-advisor
L. HAMAMI	Professor, ENP	Examiner
L. HASSAINE	DR, CDER	Examiner
F. CHEKIRED	MRA, UDES	Examiner

ENP-2021

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

Ecole Nationale Polytechnique
Department of Electronics
Laboratory of Communication Devices
and Photovoltaic conversion

PhD Thesis

In Electronics

Presented by:

Selim Sahrane

Study of the realization of a Non-Intrusive Load Monitoring System

Thesis defended on December the 1st before the jury composed of:

A. BELOUCHRANI	Professor, ENP	President
M. HADDADI	Professor, ENP	Advisor
M. ADNANE	Professor, ENP	Co-advisor
L. HAMAMI	Professor, ENP	Examiner
L. HASSAINE	DR, CDER	Examiner
F. CHEKIRED	MRA, UDES	Examiner

ENP-2021

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique
Département d'Electronique
Laboratoire des Dispositifs de Communication
et de Conversion Photovoltaïque

Thèse de Doctorat

En Electronique

Présentée par :

Selim Sahrane

Etude de la réalisation d'un système NILM (Non-Intrusive Load Monitoring)

Soutenue le 1er décembre devant le jury composé de :

A. BELOUCHRANI	Professeur, ENP	Président
M. HADDADI	Professeur, ENP	Directeur
M. ADNANE	Professeur, ENP	Co-directeur
L. HAMAMI	Professeur, ENP	Examinateur
L. HASSAINE	DR, CDER	Examinateur
F. CHEKIRED	MRA, UDES	Examinateur

ENP-2021

ملخص:

تهدف مراقبة الحمل غير المتطفلة (NILM) ، والتي تسمى أيضًا ، تفصيل الحمل أو تفصيل الطاقة ، إلى استنتاج استهلاك الطاقة الكهربائية المستهلكة من طرف كل جهاز كهربائيي من الطاقة الإجمالية. تحد العديد من التحديات من نشر أنظمة NILM في المباني السكنية والتجارية. في هذا العمل نتعامل مع حالة تفصيل الطاقة في وجود أجهزة غير مستهدفة. تمثل الأجهزة الغيرمستهدفة في سياقنا الأجهزة الكهربائية التي ليس لدينا ملصقات لها أثناء مرحلة التدريب لخوارزمية NILM. ولكن هذه الأجهزة يضيف استهلاكها إلى الطاقة الإجمالية. في هذا العمل، نقدم طريقة لتفصيل الطاقة على نهج التصنيف متعدد العلامات وندرس تأثير الأجهزة غير المستهدفة على أداء تصنيف NILM. نظهر أن الأجهزة غير المستهدفة يمكن أن تؤثر سلبًا على أداء تصنيف NILM. وجدنا أيضًا ارتباطًا مهمًا بين التأثيرعلى أداء التصنيف NILM و معامل التداخل بين توزيعات طاقات الأجهزة المستهدفة و طاقات الأجهزة غيرالمستهدفة. يتم الحصول على النتائج باستخدام مجموعة بيانات لقياسات الطاقة متاحة للجمهور.

الكلمات المفتاحية: NILM ، معامل التداخل ، تفصيل الأحمال ، الأحمال غير المستهدفة.

Résumé

Le Non-Intrusive Load Monitoring (NILM), aussi appelé, désagrégation de charge, a pour but d'inférer l'énergie consommée par chaque appareil ou charge électrique à partir du signal de la puissance globale. Plusieurs challenges limitent le déploiement des systèmes NILM dans les bâtiment résidentiels et commerciaux. Dans le présent travail, nous traitons le cas de la désagrégation de charge en présence de charges non-ciblées. Les charges non-ciblées sont les charges électriques pour lesquelles nous ne disposons pas de labels durant la phase de l'entrainement de l'algorithme NILM. Or, ces charges peuvent exister dans un cas réel, et la consommation de ceux-ci s'ajoute au signal de la puissance globale. Dans ce travail, nous présentons notre méthode de désagrégation de charge basée sur une approche de classification multi-label, et étudions l'impact des charges non-ciblées sur la performance de désagrégation. Nous montrons que les charges non-ciblées peuvent négativement affecter la performance de désagrégation des systèmes NILM. Nous avons aussi trouvé une corrélation significative entre l'impact sur la performance de désagrégation et le coefficient de chevauchement entre la distribution de puissance des charges ciblées et la distribution de puissance des charges non-ciblées. Les résultats sont obtenus utilisant un jeu de données de mesures de puissance disponible publiquement.

Mots-clés : NILM, coefficient de chevauchement, désagrégation de charge, charges non-ciblées.

Abstract

Non-Intrusive Load Monitoring (NILM), also called, load disaggregation aims to infer load level electrical energy consumption from the aggregate power signal. Several challenges are limiting the deployment of NILM systems in residential and commercial buildings. In this work, we treat the case of energy disaggregation in the presence of non-targeted loads. Non-targeted loads in our context stand for electrical loads for which we do not have labels during the training phase of the NILM algorithm. However, those loads may exist in a real-world scenario, and their power consumption adds to the aggregate power signal. In this work, we present our load disaggregation method based on a multi-label classification approach and study the impact of non-targeted loads on the NILM disaggregation performance. We show that the non-targeted loads can negatively affect the disaggregation performance of NILM systems. We also found a significant correlation between the disaggregation performance impact and the overlapping coefficient between the targeted and non-targeted loads' power distributions. Results are obtained using a publicly available dataset of power measurements.

Keywords: NILM, overlapping coefficient, load disaggregation, non-targeted loads.

ACKNOWLEDGEMENT

This thesis has been conducted at the Laboratory of Communication Devices and Photovoltaic conversion in the Department of Electronics of Ecole Nationale Polytechnique (ENP) of Algiers.

I would like to thank the following people:

- My supervisor, Pr. Mourad Haddadi for giving me the freedom to choose a research topic and for his valuable advice.
- My second supervisor, Pr. Mourad Adnane for helping me in the writing phase of our articles and with whom I learned a lot about the process of scientific publishing.
- My mother for her support, patience and encouragement..

CONTENTS

Contents

List of figures

List of tables

General Introduction 13

1 Introduction to the NILM problem 15

1.1 Introduction 16

1.2 NILM working principle 16

1.3 NILM methods categorization 17

1.3.1 Event-based/eventless 18

1.3.2 Learning paradigm 18

1.3.3 Data granularity 19

1.4 Load types 20

1.4.1 On-Off loads 20

1.4.2 Finite state machine (FSM) loads 20

1.4.3 Continuously variable loads 21

1.4.4 Permanent consumer loads 21

1.5 Features 22

1.5.1 Steady-state features 22

1.5.2 Transient features 23

1.5.3 Non-traditional features 23

1.6 Datasets 24

1.7 NILM performance evaluation metrics 24

1.8 Conclusion 25

2 Review of NILM methods 27

2.1 Introduction 28

2.2 Hart's method 28

2.3 Hidden Markov Models based methods 29

2.4 Sparse Coding-based methods 32

2.5 Deep Learning-based methods 33

2.6 Other methods 35

2.7 Conclusion 35

3 Load disaggregation using Multiclass Classification 36

3.1 Introduction 37

3.2 Problem formulation 37

3.3 Proposed method 37

3.3.1 Method description 37

3.3.2 Data 38

3.3.3 Feature extraction 39

3.3.4 Feature selection 40

3.3.5 Training 42

3.3.6 Prediction 42

3.4 Results and discussion 43

3.5 Conclusion 46

4 Multi-label Classification for Load Disaggregation 48

4.1 Introduction 49

4.2 Data 49

4.3 Disaggregation method 49
4.3.1 Filtering 50
4.3.2 Discretization 51
4.3.3 Event detection 52
4.3.4 Features extraction 53
4.3.5 Classification 53
4.3.6 Data labeling 54
4.4 Performance evaluation 55
4.5 Results and discussion 56
4.5.1 Preprocessing results 56
4.5.2 Event detection results 57
4.5.3 Labeling results 57
4.5.4 Disaggregation performance 57
4.6 Conclusion 61

5 A Clustering-Based Event Detection Method for NILM 62
5.1 Introduction 63
5.2 Background 63
5.3 Data 65
5.4 Methods 65
5.4.1 Filtering 66
5.4.2 Data clustering 66
5.4.3 Events extraction 68
5.4.4 Evaluation metrics 68
5.5 Results and discussion 69
5.6 Conclusion 73

6 Near Real-Time Low-Frequency Load Disaggregation 74
6.1 Introduction 75
6.2 Proposed method 77
6.2.1 Multi-label Classification 77

6.2.2 Multi-output Regression 78
6.2.3 Data 79
6.3 Results and discussion 79
6.4 Conclusion 83

7 Effect of Non-Targeted Loads on NILM Performance 85
7.1 Introduction 86
7.2 Background 86
7.3 Problem formulation 88
7.4 Methods 89
7.4.1 Computation of Pearson's correlation 91
7.4.2 Statistical significance test 91
7.5 Results and discussion 93
7.6 Conclusion 98

General Conclusion 100

Bibliography 102

LIST OF FIGURES

1.1 Figure showing the working principle of a NILM system. 16

1.2 Consumption patterns of different loads in the aggregate signal [1]. . . . 17

1.3 Representation of an appliance's consumption states using a Markov chain [2]. 18

1.4 Figure showing the operating cycle of a refrigerator from the REDD dataset [3]. 20

1.5 Figure showing the operating cycle of a dishwasher from the REDD dataset [3]. 21

1.6 Figure showing the operating cycle of a laptop from the ECO dataset [4]. 22

1.7 Figure showing the power draw of smoke alarms from the REDD dataset [3]. 23

1.8 Block diagram showing the learning model development process for NILM. 26

2.1 Bloc diagram showing the different steps of Hart's NILM algorithm [1]. 29

2.2 Data clusters of each load in the complex power space (P-Q) [1]. 30

2.3 Figure showing how an HMM is used to model the power consumption of a refregirator [5]. 32

3.1 Figure showing an example of the aggregate power signal when only the fridge and dishwasher are used. 38

3.2 Heatmap for std, mean, min, max, med, var. 41

3.3 Scatter plot of mean power versus std of power for each appliance. . . . 41

3.4 Comparison between predicted and ground truth energy breakdowns. . 46

4.1 Block diagram of the proposed NILM approach. 50

4.2 Discretization of the filtered aggregate signal $S_{agg,f}$ produces the signal $S_{agg,d}$. The horizontal lines represent the bins' edges e. The spaces between the horizontal lines represent the bins' width w. 52

4.3 Effects of median filtering and discretization on the raw aggregate signal. The raw aggregate power signal is first filtered than discretized. . . 58

4.4 Detection signal used for event detection, obtained by differentiating the filtered and discretized aggregate signal. 59

4.5 Comparison between the constant and adaptive thresholding methods. The adaptive labeling method detects (high state) all the operating cycles while the constant method detects only parts of the signal which are greater than the constant threshold $Thr = 50W$. 60

5.1 Block diagram showing the steps of an event-based NILM. 63

5.2 Example of operational state changes events in the aggregate power signal of a household. 65

5.3 Block diagram showing the steps of our event detection method. 66

5.4 Example of the resulting segmentation of the aggregate signal into different steady power consumption states using kmeans clustering algorithm. Each cluster contains samples from the aggregate signal that have similar power values. 67

5.5 Fridge consumption cycle in the aggregate signal before and after applying the median filter. 70

5.6 Figure showing the obtained result using our event-detection method. Each red marker represents a detected event. 71

5.7 Performance results of our method using the Kmeans algorithm. 71

5.8 Performance results of our method using the Mean Shift algorithm. . . 72

6.1 Block diagram of the proposed method. 77

6.2 Figure showing the disaggregation result obtained using our method. The power signals of the fridge, kitchen outlet, and microwave are disaggregated from the aggregate power signal. 82

6.3 Comparison between the estimated energy by our method for each load, and the corresponding actual energy consumption. 83

7.1 Active power of the targets, non-targets, and aggregate signals. 89

7.2 Block diagram showing the design of experiment 1 and experiment 2. . 90

7.3 In green, the overlap between two probability density functions, $f_1(p)$ and $f_2(p)$. 91

7.4 Figure showing the steps followed in the statistical significance test. . . 92

7.5 Scatter plot of the values (after standardization) used in the hypothesis test. The x-axis represents the performance impact I and the y-axis represents the overlapping index (OVL) computed for the non-targets' signal and the targets' signal. The blue line represents the fitted regression line. 98

LIST OF TABLES

1.1 "Table showing some NILM datasets and theiir attributes." 24

3.1 Number of operating cycles used for training and prediction 39

3.2 Results of the mutual information estimation for each feature $V_{parameter,i}$ with the target variabe y 40

3.3 Performance comparison between our method and the Seq2point method [6]. 44

3.4 Results of the performance evaluation 45

4.1 Performance comparison between our method and the MLCDDL method [7]. 61

5.1 Performance results for our method using K-means, Mean Shift, and Mini-batch K-means clustering algorithms. 72

6.1 Performance results for each load. 81

6.2 Overall performance results of the proposed NILM method. 81

6.3 Performance comparison between our method and the PALDi method [8]. 81

7.1 Targets and non-targets for experiment 1. 94

7.2 Targets and non-targets for experiment 2. 95

7.3 Performance results for experiment 1. 96

7.4 Performance results for experiment 2. 97
7.5 Table showing how each non-targeted load impacts the NILM performance in household 5. 97
7.6 Values used for the statistical significance test. 98

GENERAL INTRODUCTION

The worldwide increase in energy demand and climate change has resulted in new technologies which aim to reduce the use of fossil fuels as well as reducing the overall energy consumption. If actions are not undertaken, the CO2 emissions will double by 2050 [9]. Residential buildings consume up to 40% of total energy [9]. Studies show that consumers don't know the necessary actions which will reduce their energy bill [10]. Furthermore, 55.2% of people do prefer reducing the usage of inefficient appliances while only 11.7% prefer replacing their old appliances. This highlights the importance of providing relevant feedback information to the consumer on his energy consumption. The most common type of feedback is provided through energy bills (e.g., KWatt/hour) [10] which does not provide detailed information to the user. The effect of energy feedback on household consumption is covered in detail in [11]. Furthermore, providing appliance-level consumption information can result in more than 12% of energy savings [12] [13] [14].

Non-intrusive load monitoring (NILM) also called load or energy disaggregation aims to infer the energy consumption of single appliances from the aggregate energy use measured at the power source interface [1]. One instrumented point is sufficient to get the energy consumption of each appliance. Intrusive load monitoring (ILM) on the other hand uses one measuring system for each appliance which has the advantage to be more accurate but is more expensive and difficult to deploy at a large scale. This is why NILM is preferred when it comes to load disaggregation [15]. Each type of appliance or electrical load is different in the way it consumes electricity due to its internal circuitry and therefore has what is called "an appliance signature" [1]. A NILM system will typically rely on machine learning and signal processing techniques in order to extract features from each appliance's signature and to disaggregate the total energy signal by identifying different signatures.

The goal of our thesis work is to study and simulate the different steps of a Non-Intrusive Load Monitoring system using real power measurements using the Python programming language.

Thesis Organization

The thesis is organized as follows:

Chapter 1 presents the general problem of load disaggregation.

Chapter 2 discusses NILM state of the art methods.

In Chapter 3, we describe our multiclass classification approach to load disaggregation.

In, Chapter 4, we describe our multi-label-based disaggregation method.

In, Chapter 5, we present our event-detection method based on a data clustering algorithm.

Chapter 6 treats our near real-time Non-Intrusive Load Monitoring approach using multi-label classification and multi-output regression.

In Chapter 7, we study the effect of non-targeted loads on the performance of NILM systems.

Finally, a General Conclusion summarizes the thesis, and highlights the main contributions of the research. It also contains recommendations for future works.

CHAPTER 1

INTRODUCTION TO THE NILM PROBLEM

Contents

1.1 Introduction . . . 16

1.2 NILM working principle . . . 16

1.3 NILM methods categorization . . . 17

1.3.1 Event-based/eventless . . . 18

1.3.2 Learning paradigm . . . 18

1.3.3 Data granularity . . . 19

1.4 Load types . . . 20

1.4.1 On-Off loads . . . 20

1.4.2 Finite state machine (FSM) loads . . . 20

1.4.3 Continuously variable loads . . . 21

1.4.4 Permanent consumer loads . . . 21

1.5 Features . . . 22

1.5.1 Steady-state features . . . 22

1.5.2 Transient features . . . 23

1.5.3 Non-traditional features . . . 23

1.6 Datasets . . . 24

1.7 NILM performance evaluation metrics . . . 24

1.8 Conclusion . . . 25

1.1 Introduction

The constant increase in energy consumption and carbon dioxide emissions has inducted the need to find solutions to reduce global energy consumption. Studies have shown that buildings' electricity consumption can be reduced by up to 15% using energy management methods [13]. Non-intrusive load monitoring aims to reduce electricity consumption by providing appliance-level consumption feedback to users. Providing household users with this type of feedback can be an effective solution to reduce energy consumption in buildings [16].

In this chapter, we describe the NILM problem. Then, we give a brief history of the NILM and discuss the most important works in the literature that helped the advancement of NILM.

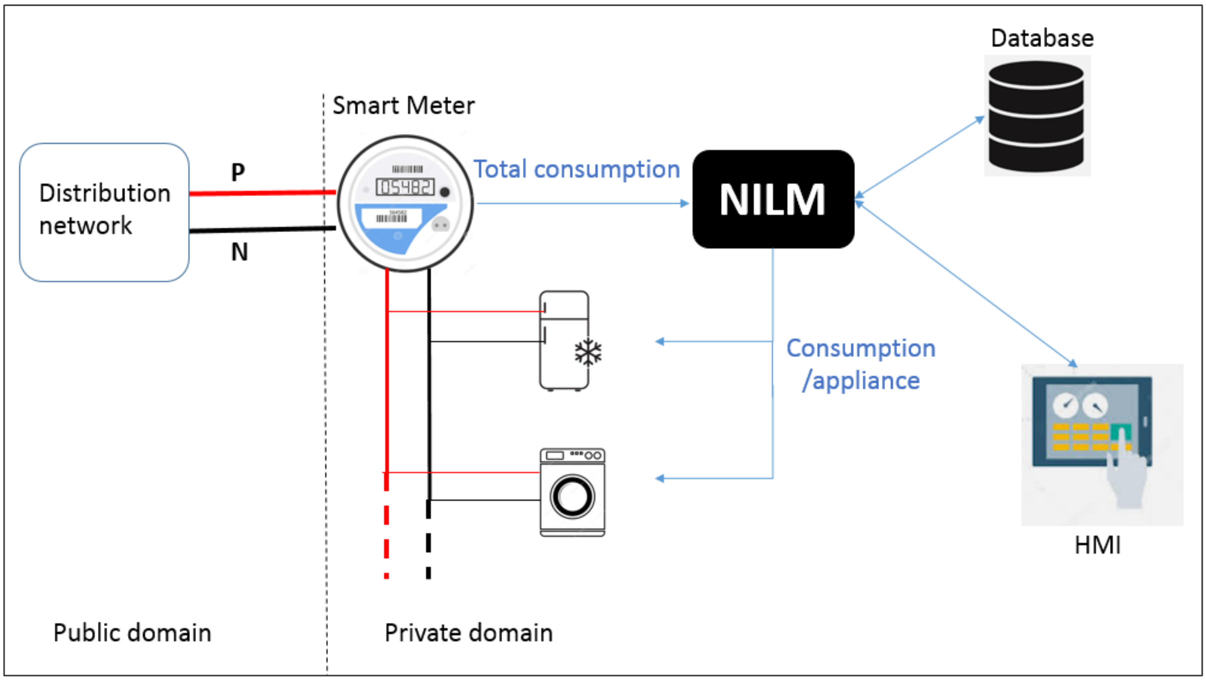


Figure 1.1: Figure showing the working principle of a NILM system.

1.2 NILM working principle

Each appliance or electrical load, in general, has specific internal hardware that results in unique power draws and consumption patterns. These patterns are called

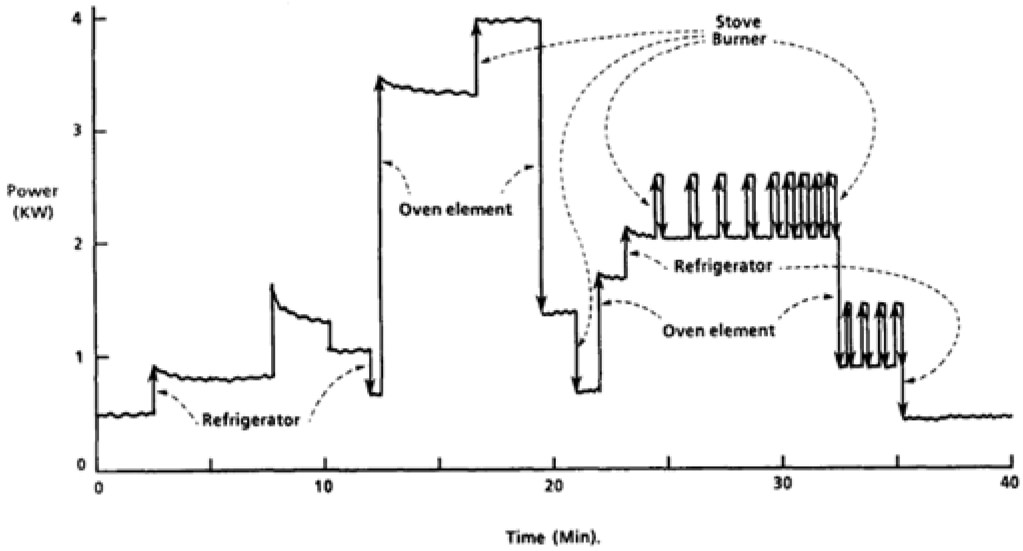


Figure 1.2: Consumption patterns of different loads in the aggregate signal [1].

"electrical signatures" [1] and are the core concept of NILM. NILM methods identify appliances by detecting these patterns in the aggregate signal of a household. The detection of these patterns is accomplished using various techniques from different fields like signal processing and machine learning to name a few. Usually, signal processing methods are used to preprocess the aggregate and ground-truth signals before constructing a learning model capable of identifying each targeted appliance from the aggregate signal of a given household. Figure 1.2 shows an example of how different electrical loads present in a household contribute to the aggregate power signal. Figure 1.1 illustrates the working principle of NILM system.

1.3 NILM methods categorization

George Hart introduced NILM in 1992 [1]. In his work, Hart showed that we could identify appliances by using their electrical load signatures. Since then, different approaches have been proposed to improve NILM performance. In [17], the authors give a detailed review of NILM methods. These approaches can be grouped according to different criteria, depending on whether they are event-based or not, depending on the statistical learning paradigm used or depending on the granularity of the data

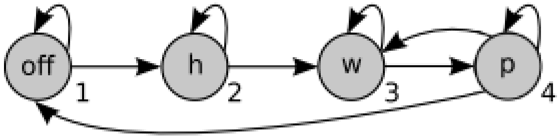


Figure 1.3: Representation of an appliance's consumption states using a Markov chain [2].

used for the development of the NILM approach.

1.3.1 Event-based/eventless

NILM methods can be grouped into event-based and eventless approaches [18]. Event-based methods use a discriminative learning model to approach the NILM problem. Contrarily, eventless methods use a generative approach. Event-based methods extract features from relevant events in the aggregate signal, like operational state changes of appliances [19] and power ON/OFF transients [20]. Then, event-based methods train a classifier to identify each targetted appliance/load using the previously extracted features. Eventless methods, on the other hand, assign each aggregated power sample to one target. The target may be a device or a combination of different devices. For that, Eventless methods use probabilistic and Bayesian methods like Hidden Markov Models (HMMs). The HMMs model the power signal of each appliance from ground truth data. For the inference step, eventless methods use an optimization method. The authors in [21] use the Viterbi algorithm to infer each appliance's state. Figure 1.3 shows how Markov chains are used to model appliances.

1.3.2 Learning paradigm

In addition to the previous categorization, one can make a distinction on the learning paradigms. Supervised learning consists of learning a prediction function by using so-called labeled or annotated data. There are two types of supervised learning. Regression, which consists in learning to predict quantitative variables, and classification, which consists in learning to predict qualitative variables. Supervised learning is the most commonly used approach as different open datasets have become available during the last decade. In [22] existing NILM datasets are reviewed. Supervised methods use labeled data to train and test a classifier. Some commonly used supervised techniques in the literature are K-Nearest Neighbours (k-NN) [23], Support Vector Machines (SVM) [24], Deep Neural Networks [25].

Unsupervised learning, unlike supervised learning, attempts to learn a prediction function with unlabeled data. This type of approach is also used as a preprocessing tool in some cases. Unsupervised learning is generally used for solving problems of data partitioning, estimation of data density distributions, or dimensionality reduction. In NILM, unsupervised learning methods don't require a training phase but instead use clustering methods [26] and Hidden Markov Models (HMMs) to model each appliance's state [27]. In addition to supervised and unsupervised methods, the literature has explored the semi-supervised approach. For instance, the authors in [28] designed general appliance models by training data from a set of houses. In addition, they used aggregate data from other habitations to infer the energy usage of each appliance.

1.3.3 Data granularity

Data granularity refers to the frequency with which data acquisition is performed. Primarily, two types of data are necessary for the development of a supervised NILM solution. The measurement data of the aggregate signal magnitudes (power, current, etc.), and the ground truth data corresponding to the measurements of the consumption magnitudes of each device targeted by the NILM solution. This data is considered to be a high sample rate data if it is obtained with a sample rate greater than 1Hz. The granularity of the data directly influences the choice of attributes to be used for the design of the learning model. Indeed, in high-frequency or high-resolution data (high-frequency data), more information can be extracted, due to the large number of events contained in the signals. The disadvantages associated with this category of data are the cost of storage necessary to store the large number of data generated, and the relatively high computing power to process this data.

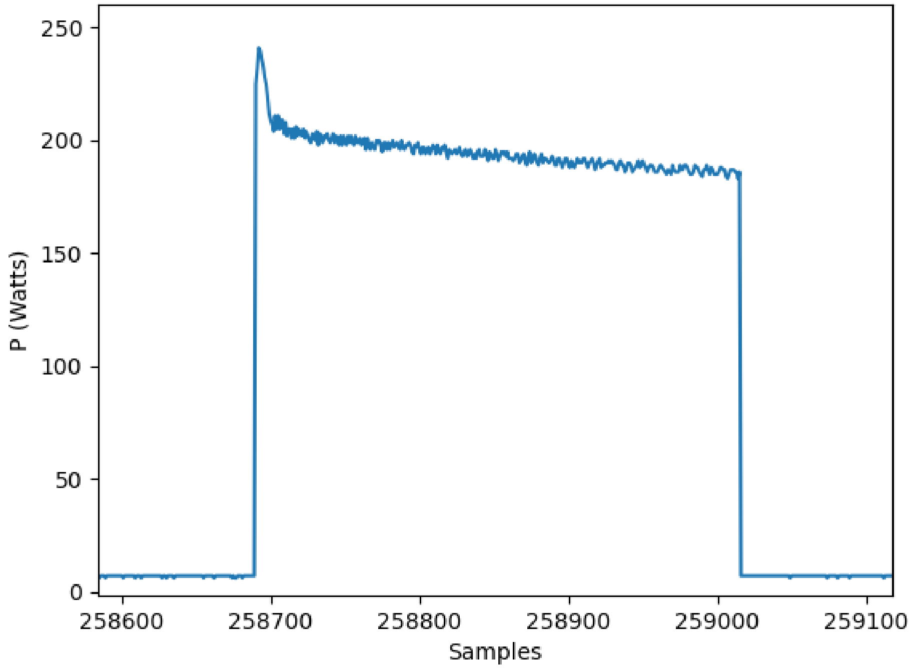


Figure 1.4: Figure showing the operating cycle of a refrigerator from the REDD dataset [3].

1.4 Load types

Load types define according to the consumption patterns of different electrical loads. We find four types of loads. Namely, On-Off, finite state machines (FSM), continuously variable, and permanent consumer loads [29].

1.4.1 On-Off loads

On-Off or two-states loads have only two consumption states, the state of operation (On) and state of non-operation (Off) of the load. Light bulbs and water pumps are examples of the On-Off load type. Figure 1.4 shows an example of the operating cycle of a refrigerator.

1.4.2 Finite state machine (FSM) loads

Finite state machines (FSM) are loads that have a finite number of operating states. Each state corresponds to the power consumption of the individual components that constitute a given appliance, like the water pump, and heating element of a washing

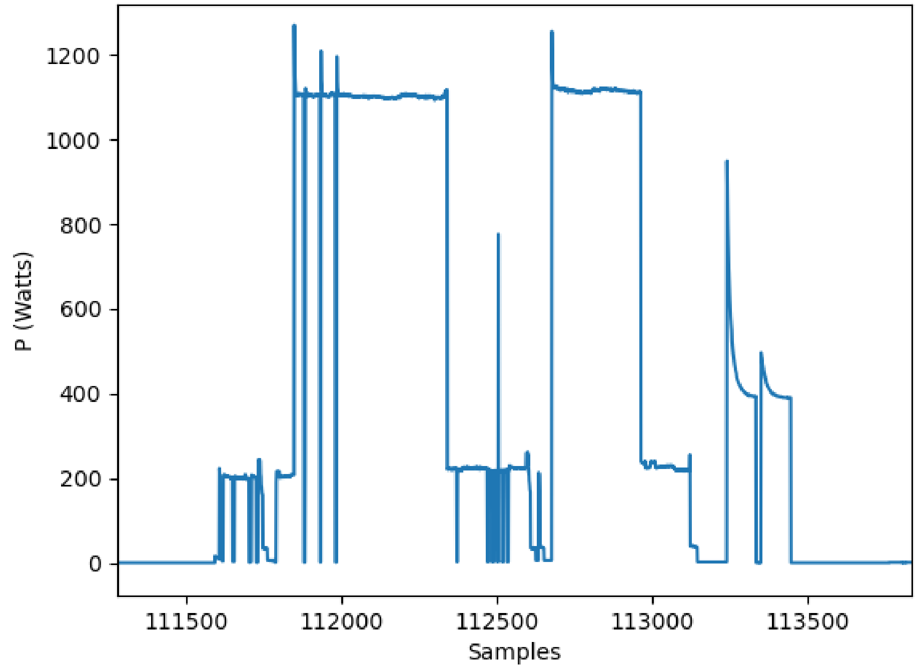


Figure 1.5: Figure showing the operating cycle of a dishwasher from the REDD dataset [3].

machine or a dishwasher. Figure 1.5 shows an example of the operating cycle of a dishwasher.

1.4.3 Continuously variable loads

Continuously variable loads, in contrast to finite state machines, have unpredictable operating states. For instance, a computer's power draw depends on the microprocessor's operating load that continuously varies according to the user's activity. Figure 1.6 shows an example of the operation of a laptop. Another example of this type of load is the printer. A printer's power consumption patterns depend on parameters like the content (text and images) to print on a given sheet and the format of the sheet used.

1.4.4 Permanent consumer loads

Permanent consumers are loads that are always On with approximately constant power draw. Examples of devices in this category include hard-wired smoke alarms and internet routers (modems). Figure 1.7 shows an example of the power draw of

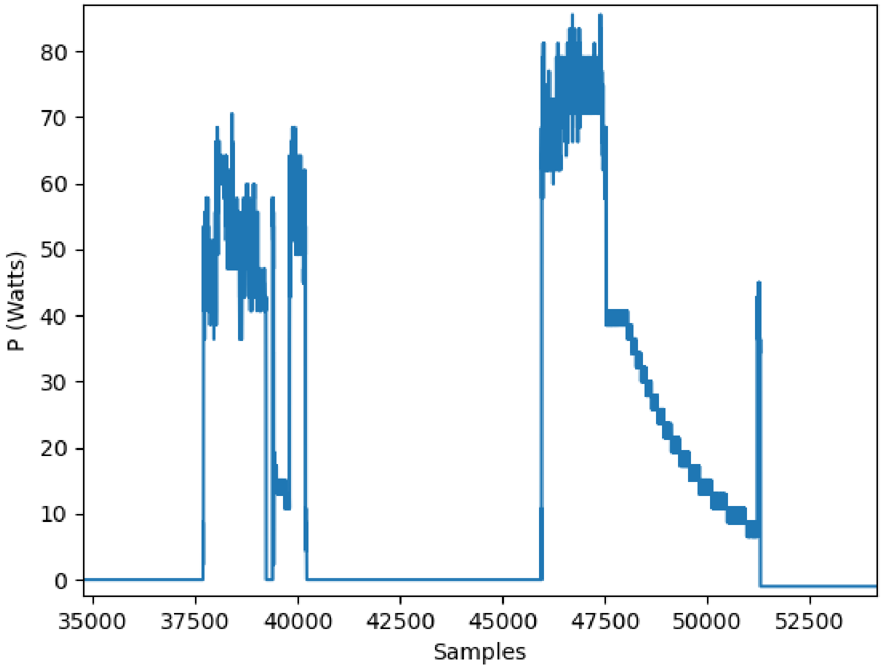


Figure 1.6: Figure showing the operating cycle of a laptop from the ECO dataset [4].

smoke alarms.

1.5 Features

Features or signatures are properties found in the signal we want to identify. In load disaggregation, the goal is to find distinctive features that best represent each targeted appliance/load. In NILM literature, we find three types of features, namely, steady-state features, transient features, and non-traditional features [30] [31].

1.5.1 Steady-state features

These types of features are extracted from the steady-state portion of the signal. The kind of extracted features depends on the granularity of data. For instance, the variations in Real Power (P), and Reactive Power (Q) are commonly used in low-frequency-based NILM methods. In high-frequency-based NILM methods, current harmonics are used for their superior discriminative capabilities, but at the expense of higher implementation costs. [32].

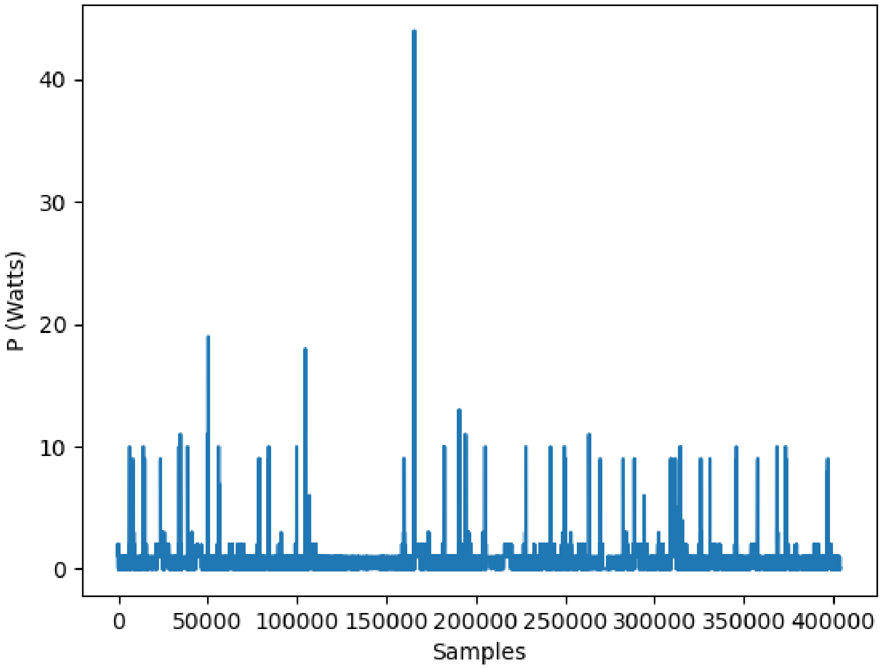


Figure 1.7: Figure showing the power draw of smoke alarms from the REDD dataset [3].

1.5.2 Transient features

Transient features are only extracted from high-frequency data because transients require a high sampling frequency to be measured. High-frequency data provides more information (details) about loads' signatures, which allows a more accurate model construction. Transient features are a better characterization of a given load in comparison to steady-state features. Steady-state features suffer from the overlap between different loads, which makes them less efficient than transient features. The drawbacks of this type of feature are the high implementation costs needed for storage and processing. Examples of transient features are the transient's duration, spectral envelopes, and current sikes values.

1.5.3 Non-traditional features

To improve the disaggregation performance, NILM researchers have explored different features, such as the use of ON-Duration Distribution and OFF-Duration Shape in [27], or the use of the time of the day in [33].

Table 1.1: "Table showing some NILM datasets and theiir attributes."

Name	**Origine**	**Coverage time**	**N° of houses**	**Ground truth resolution**	**Aggregate resolution**
REDD [3]	USA	3-19 days	6	3 s	1 s & 15 KHz
AMPds [34]	Canada	1 year	1	1 min	1 min
UK-DALE [35]	UK	3-17 months	4	6 s	1-6 s & 16 KHz
ECO [4]	Switzerland	8 months	6	1 s	1 s

1.6 Datasets

To develop NILM methods, we need aggregate measurements and ground truth measurements data. Ground truth data are measurements obtained by measuring each targeted load's consumption. Several acquisition systems are required to get these data, as each targeted load/appliance demands a dedicated acquisition system. During the last decade, a considerable number of datasets have been published. Their role is to facilitate research in the field of NILM by providing researchers with real consumption data acquired in different homes. Among these datasets, some are high frequency, while others are low frequency. In addition, some are more suitable for a certain type of NILM approach, such as event-driven approaches for example. Table 1.1 shows the characteristics of some of the NILM datasets. A review of existing NILM datasets is found in [22].

1.7 NILM performance evaluation metrics

Metrics are measures of the performance of an algorithm or method in general. Thus, a performance metric should reflect the behavior of an algorithm under different conditions. The metric choice depends mainly on the nature of the problem at hand and the desired end goal. Usually, a learning model development uses training, validation, and testing sets. The performance of the algorithm is computed using each set to evaluate the generalization capability of the algorithm. In NILM performance evaluation two sets are commonly used, referred to as training and testing sets. Also, the choice of metrics varies considerably, making the comparison of different NILM methods challenging. A fair comparison between different NILM approaches, even with the usage of the same evaluation metric, is nearly impossible due to incomplete or missing problem definitions [36]. Event-based methods use two metrics to evaluate both the event detection and energy estimation steps. Eventless NILM methods, on the other hand, are only interested in the energy estimation error because no event detection step is involved. An in-depth review of NILM performance metrics is found in [22]. Figure 1.8 shows the steps of the development of a NILM learning model. We observe that model development is an iterative process that aims to find the hyperparameters that result in the best performance of the learning model. First, initial parameter values are fed to the learning algorithm. Second, the algorithm is trained then tested using training and testing sets. Here, training and testing sets refer to data after preprocessing and feature selection rather than the raw data. Thrid, training, and testing performance are computed using the same metric(s). Finally, the obtained results are evaluated to decide if the model needs further improvements. Evaluation of NLIM methods and used metrics have been treated in several articles, like in [37] and citepereira2018performance. We kindly invite the reader towards these references for additional information about this subject.

1.8 Conclusion

In this chapter, we introduced the general problem of load disaggregation. We described the working principle of NILM methods and presented the different approaches found in the literature. We also discussed the datasets used in NILM research and explained the NILM design and evaluation process.

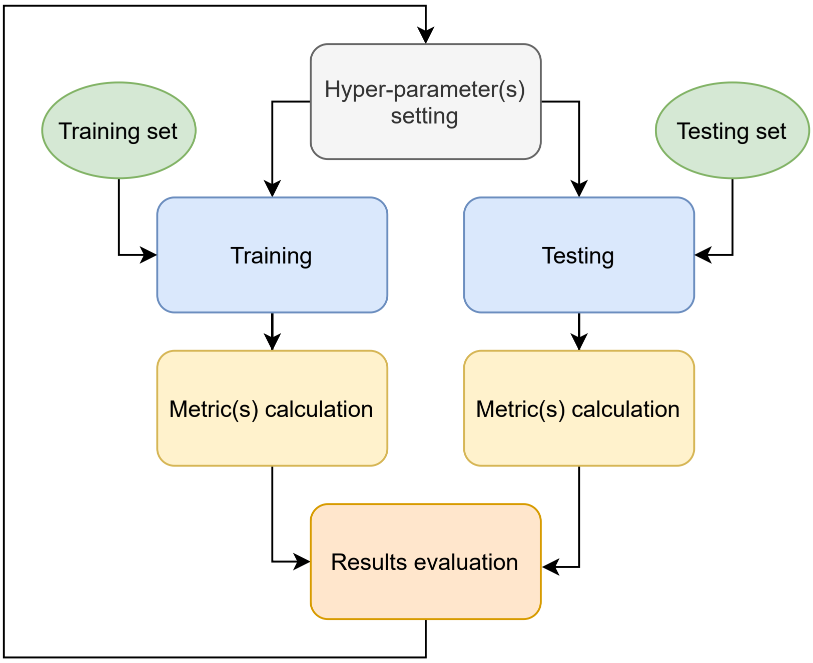


Figure 1.8: Block diagram showing the learning model development process for NILM.

CHAPTER 2

REVIEW OF NILM METHODS

Contents

2.1 Introduction 28

2.2 Hart's method 28

2.3 Hidden Markov Models based methods 29

2.4 Sparse Coding-based methods 32

2.5 Deep Learning-based methods 33

2.6 Other methods 35

2.7 Conclusion 35

2.1 Introduction

Non-Intrusive Load Monitoring has been around for almost thirty years. All along this period, a plethora of methods has been proposed in the literature. In the last decade, most of the proposed approaches rely on probabilistic models, machine learning and deep learning methods. In this chapter, we review the approaches that have contributed the most to the state-of-the-art.

2.2 Hart's method

Hart's load disaggregation method laid the foundations of NILM research. Other existing methods in the literature seek to improve the disaggregation performance by exploring different approaches. Hart, in his work [1], described two different versions of his proposed NILM algorithm. The first being the Manual-Setup-NILM (MS-NILM) which is a supervised learning implementation of his NILM algorithm, and the Automatic-Setup-NILM (AS-NILM), which is an unsupervised learning implementation of his NILM algorithm. As seen in figure 2.1. In step A (see figure 2.1), the average active and reactive power and RMS voltage are measured over 1-second intervals. In step B, the measured average active and reactive power are normalized to counter-react the voltage variations due to electrical grid operation. In step C, the time/index and the size of each operational state change are extracted from the aggregate power signals. In the case of the AS-NILM, the extracted step changes values are clustered in step D. The clustering algorithm locates the power changes in a two-dimensional feature space of active and reactive power. In step E, positive and negative clusters of similar magnitudes are paired and the remaining events and clusters are matched to existing or new clusters using a best-likelihood algorithm. Figure 2.2 shows the obtained clusters. In step F, the obtained clusters from the aggregate signal are associated with known operating state values obtained from the ground truth of each targeted load. In step G different statistics are computed to desctibe the energy consumption of each load in the household and when each load was active. This information is in turn used in step H to manually label each targeted load. This

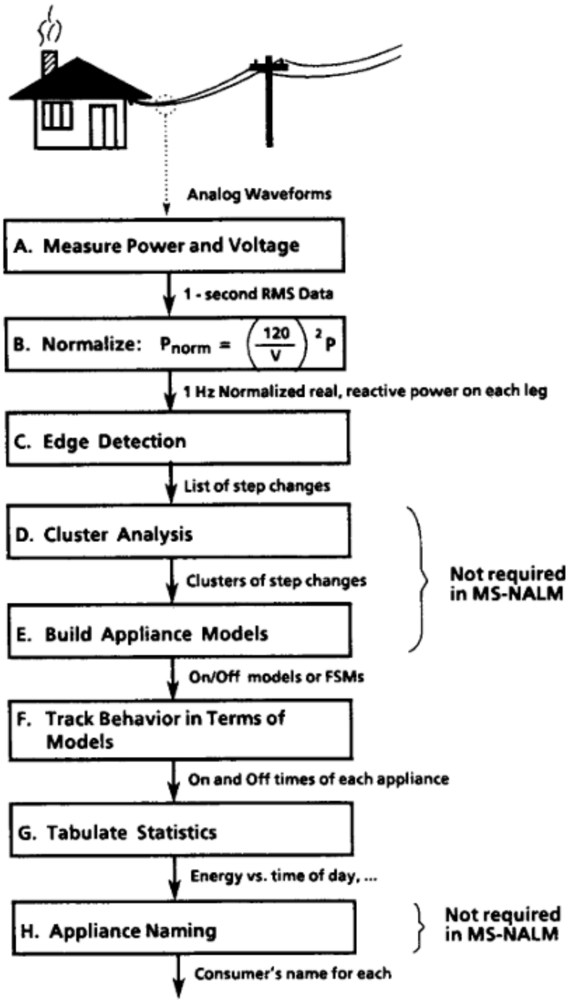

Analog Waveforms
A. Measure Power and Voltage
1 - second RMS Data
B. Normalize: $P_{norm} = \left(\frac{120}{V}\right)^2 P$
1 Hz Normalized real, reactive power on each leg
C. Edge Detection
List of step changes
D. Cluster Analysis
Clusters of step changes
E. Build Appliance Models
Not required in MS-NALM
On/Off models or FSMs
F. Track Behavior in Terms of Models
On and Off times of each appliance
G. Tabulate Statistics
Energy vs. time of day, ...
H. Appliance Naming
Not required in MS-NALM
Consumer's name for each

Figure 2.1: Bloc diagram showing the different steps of Hart's NILM algorithm [1].

last step as well as steps D and E are not required in the MS-NILM, as this step are executed in a supervised learning manner.

2.3 Hidden Markov Models based methods

As discussed in chapter 1, NILM methods categorize into event-based and eventless. For eventless approaches, the Factorial Hidden Markov Model (FHMM) [38], has been applied to the problem of load disaggregation in [27] as a probabilistic model for the aggregate signal. Factorial HMMs allow modeling multiple independent hidden state sequences. In the context of NILM, this is equivalent to modeling each targeted load

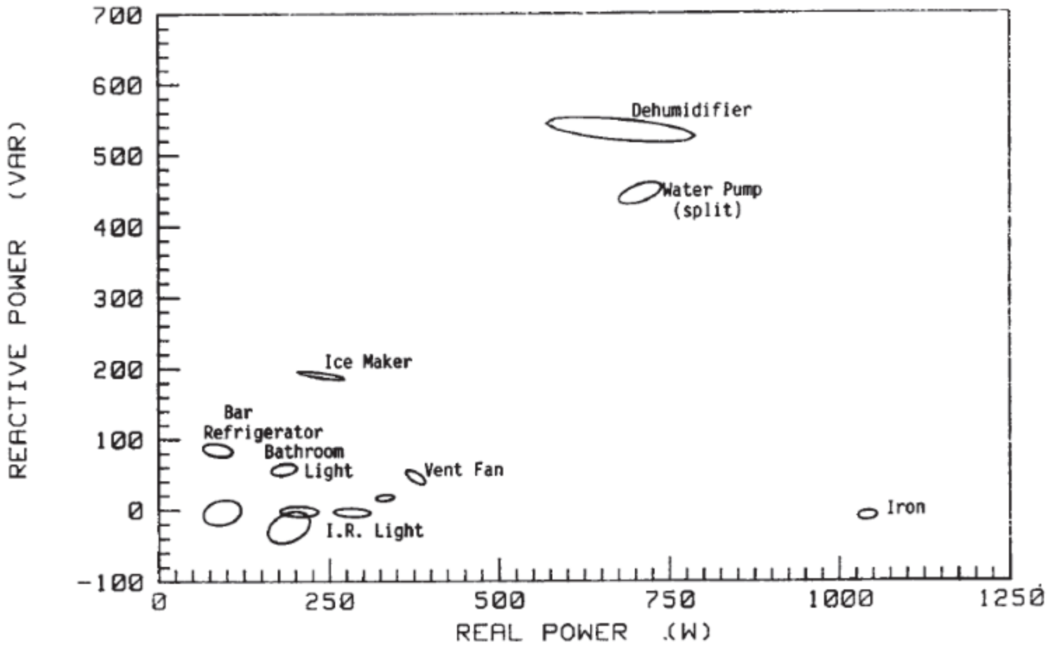


Figure 2.2: Data clusters of each load in the complex power space (P-Q) [1].

by a unique Hidden Markov Model (HMM). In HMMs, patterns are thought of as a product of sources that act statistically. The goal is to model the sources. A state vector is used to model the underlying behavior of a data source. The output of these states is modeled through an emission probability distribution. Moreover, power signals can be characterized as composed of stationary stochastic processes and a typical HMM is known for modeling the combination of stationary stochastic processes [5]. For instance, in [5], each targeted load is modeled using an individual HMM. The individual HMMs are then merged into a single HMM that can describe the combined power (i.e. the aggregate power). The states of each HMM are used to model the steady states (i.e. operational states) of each targeted load. The state transition probabilities of the HMM are used to model the transitions between the processes (i.e. operational state changes). As shown in figure 2.3. Each state of the combined load HMM model representing the aggregate power is modeled as a combination of the individual operating states of each load at a given instant. For inference, given a combined load profile the Viterbi algorithm is used to decode the combined load profile into individual load states. The authors in [27] used the FHMM and other of its variants to investigate the effectiveness of several unsupervised disaggregation methods on low-frequency power measurements collected in real homes. The authors found that the Conditional Factorial Hidden Semi-Markov Model (CFHSMM), which integrates additional features related to when and how appliances are used, allows a better representation of the power use of individual appliances. They also found that CFHSMM outperformed the other unsupervised disaggregation methods. Another unsupervised method is proposed in [39]. The authors used the Additive Factorial Hidden Markov Model, which is a special case of the FHMM where the output is an additive function of the different hidden states. In addition, they encoded the difference signal of the aggregate signal (amount of the variation) by modifying the FHMM. The resulting algorithm, referred to as AFAMAP (Additive Factorial Approximate MAP), combines the two FHMM models (additive and difference FHMM) into a single joint problem. The motivation to use approximate inference instead of exact inference is because, as the number of devices to disaggregate grows, the evaluation of all the possible HMM evolutions that could have generated the aggregate output implies an increase in the computational complexity of the disaggregation process. Therefore, the authors proposed an algorithm called Additive Factorial Approximate MAP (AFAMAP) which can bypass the unreachable exact inference through the approximation of the Maximum A Posteriori Probability [40]. A variant of the previously described difference FHMM was used in [21] to develop generic appliance models that are used to disaggregate the energy of common high-energy consuming appliances in an unsupervised way. Each appliance is represented as a probabilistic graphical model (i.e. the difference FHMM). A training process is used to learn the parameters of each model from the aggregate power using the expectation maximization algorithm. For the disaggregation step, an extension of the Viterbi algorithm is used to extract the targeted appliance's power signal from the aggregate signal. The extracted signal is then subtracted from the aggregate power signal. The same process is applied for each modeled appliance.

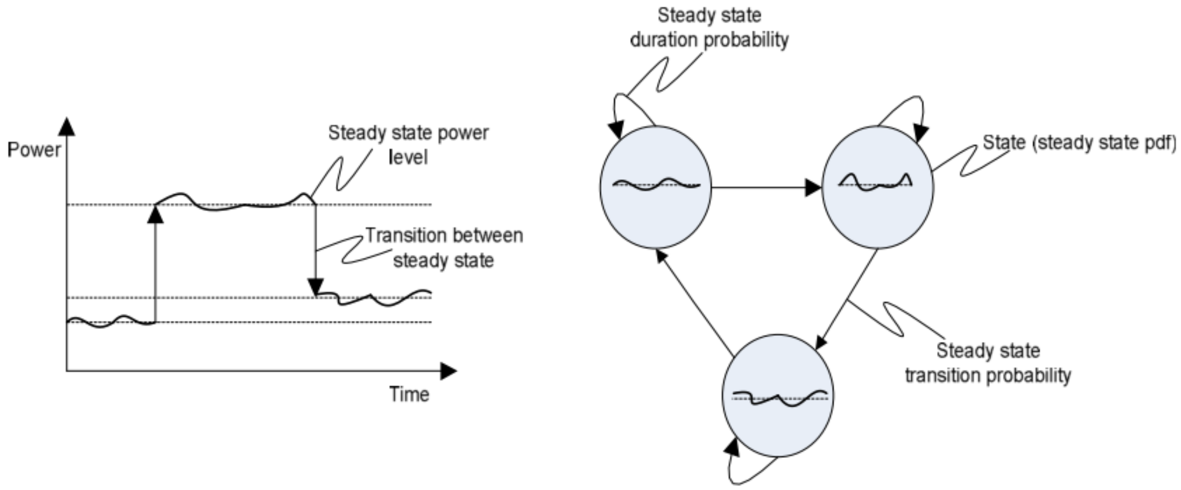

Power
Steady state power level
Transition between steady state
Time
Steady state duration probability
State (steady state pdf)
Steady state transition probability

Figure 2.3: Figure showing how an HMM is used to model the power consumption of a refregirator [5].

2.4 Sparse Coding-based methods

Sarse coding methods, also referred to as dictionary learning, model each load's signal as a sparse linear combination of the atoms of an unknown dictionary. A discriminative approach is then used to learn the sparse coefficients and the dictionary for each device. For instance, in [41], a sparse coding approach is used to disaggregate low-resolution, hourly aggregate data. The sparse coding algorithm is used to learn a model of each device's power consumption over a typical week, then these learned models are combined to predict the energy consumption of each load from the aggregate signal alone. In [42] the energy consumption of each device is modeled using a mixture of dynamical models corresponding to different operation modes of the device. Then, the signature consumption patterns are found by defining an appropriate dissimilarity between pairs of energy snippets and selecting representative snippets, which are referred to as "powerlets". In [43] and [7], instead of learning one level of a dictionary, the proposed method learns multiple layers of dictionaries for each device. These multi-level dictionaries are used as a basis for source separation during disaggregation.

2.5 Deep Learning-based methods

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection, and many other domains [44]. The advances in the hardware design of graphical processing units (GPU) and the creation of tensor processing units (TPU) contributed significantly to the rapid improvement of Deep Learning state of art methods. As most researchers in other fields, NILM researchers explored the benefits of applying Deep Learning. For instance, the use of Deep Learning methods was first proposed in [45]. The authors tested three architectures. The first is a long short-term memory (LSTM) which is a type of recurrent neural network (RNN), the second is a denoising autoencoder (dA), and the third architecture is a network that regresses the start time, end time, and average power demand of each appliance activation. The authors found that Deep Learning networks presented the advantage of generalization over unseen houses and that LSTMs worked better on two-state appliances than on multi-state appliances. In [46] an auto-encoder and LSTM architecture similar to the one used in [45] were presented. The proposed LSTM architecture contains more CNN layers to extract more representative features from the power signal. The authors reported poor performance for the auto-encoder and superior performance of their LSTM architecture compared to previous works. All these Deep Learning methods, as well as other similar ones, used a sequence-to-sequence mapping approach to model the load disaggregation problem. The "seq2point" method proposed in [6], introduced a sequence-to-point approach where each input window is mapped to a point. In other terms, the proposed network seeks to predict the middle point of any given input window. The authors reported state-of-the-art performance. In [47], the nature of the NILM problem is converted into an image classification problem. The authors considered voltage-current (VI) trajectories that were already treated in previous works but treated them as weighted pixelated VI images that can be used as inputs for a CNN. The proposed method was evaluated on different datasets than the ones used by previous state-of-the-art Deep

Learning methods, which makes a fair performance comparison difficult. VI trajectory images were also used in [48] with siamese neural networks for detecting previously unidentified appliances in an automated way. The siamese neural network is used to find a new, lower dimensional feature space where samples of the same appliance are clustered (i.e. near to each other). The DBSCAN clustering algorithm is then used to detect outlier samples that don't belong to any cluster. These samples are then labeled as 'unidentified'. In [49], an approach for hourly energy breakdown based on a tree-structured CNN model is presented. Each CNN learns to recognize a single targeted load, then a tree is used for iterative energy breakdown. At each iteration, a source is subtracted from the aggregate, similar to the approach used in [21]. The resulting signal is then used as input to recognize the designated appliance. A multi-label deep learning method was proposed in [50]. Multi-label classification was already treated in the literature, as we will see in the next section. A Generative Adversarial Network (GAN) was used in [51] to improve Kelly's denoising autoencoder [45] by integrating the generator of a GAN into Kelly's denoising autoencoder's disaggregation process to support a more accurate reproduction of appliance load sequences. In [52] and [53], Federated Deep Learning was used to tackle the data privacy problem of NILM systems. This problem intervein when data owners are asked to combine their local data to train a deep neural network model. This process often involves legal issues such as data privacy and security. The Federated Deep Learning framework allows each client (i.e. household's smart meter) to send model parameters instead of data. The parameters are then aggregated to a central cloud server to update the global deep learning model. An adaptation of the Bidirectional Encoder Representations from Transformers (BERT) [54] was proposed in [55]. The results show that the self-attention mechanism and bidirectional transformer model are effective for NILM tasks. An in-depth review of NILM Deep Learning-based methods is found in [56].

2.6 Other methods

In [57] Graph Signal Processing (GSP) was adapted for the NILM problem. GSP is a field where data is represented by a discrete signal indexed by a graph. The acquired signal samples correspond to the graph nodes with cleverly defined weighted graph edges. Then, classical signal processing concepts can be applied to these "graph signals" [58]. Multi-label classification was introduced in [59]. Multi-label classification allows the mapping of input data to multiple classes/labels which is advantageous in the case of NILM as we will see in chapter 4. In [60], Particle Filtering was introduced. The authors used particle-filtering (PF) with a factorial hidden Markov model (FHMM) for appliance state tracking. Conditional random fields (CRFs) were applied to NILIM in [61]. CRFs are similar to HMMs but with a different nature. CRFs are discriminative models, which maximize the conditional probability of observation and state sequences [62]. In [63], the Extreme Learning Machine (ELM) neural network was used. ELMs were introduced by Huang et al. in [64] for single-layer feed-forward networks (SLFNs) to overcome problems with the back-propagation algorithm.

2.7 Conclusion

In this chapter, we reviewed the major NILM approaches and their appropriate methods. As in most research fields, in NILM, the approaches' choice is influenced by the obtained results in other research domains like speech recognition, natural language processing (NLP), and image processing. Most of the methods we have seen in this chapter focus on improving the disaggregation performance using, in most cases, datasets that contain power measurements obtained from different households and complex disaggregation methods. Although the state-the-art disaggregation performance continues to improve, practical implementation considerations are still lacking, as very few works account for the practicalities of NILM deployment.

CHAPTER 3

LOAD DISAGGREGATION USING MULTICLASS CLASSIFICATION

Contents

3.1 Introduction **37**

3.2 Problem formulation **37**

3.3 Proposed method **37**

3.3.1 Method description 37

3.3.2 Data 38

3.3.3 Feature extraction 39

3.3.4 Feature selection 40

3.3.5 Training 42

3.3.6 Prediction 42

3.4 Results and discussion **43**

3.5 Conclusion **46**

3.1 Introduction

In this chapter, we present our approach to load disaggregation using multiclass classification. The goal of this work is to explore if a straightforward approach using multiclass classification can give good disaggregation performance results. Also, this work is our first attempt to load disaggregation. Therefore, it will serve as a starting point in our NILM study. The remaining of this chapter is organized as follows: in section 2, we formulate the load disaggregation problem, in section 3, the proposed method is described in detail. In section 4, results are presented and a discussion is made. Finally, a conclusion is given in section 5.

3.2 Problem formulation

Here we represent the aggregate power time series $S_{agg}(t)$ as the algebraic sum of each targeted load/appliance's time series $S_i(t)$ with $i = 1, .., N$, and N the number of targeted electrical loads, as shown in equation 3.1.

$$S_{agg}(t) = \sum_{i=1}^{N} S_i(t) + S_{noise}(t) \tag{3.1}$$

$S_{noise}(t)$ represents the noise time series which comprises measurement noise and power contributions of other non-targeted loads/appliances. The goal of NILM is to estimate the energy contributions of each targeted load, characterized by its time series $S_i(t)$. Figure 3.1 shows an example of the aggregate and ground truth power signals.

3.3 Proposed method

3.3.1 Method description

In this work, we use a mini-batch classification method to detect and estimate the energy use of household appliances in near real-time. The mini-batches are obtained by applying a moving window on power measurements sampled at a frequency of 1

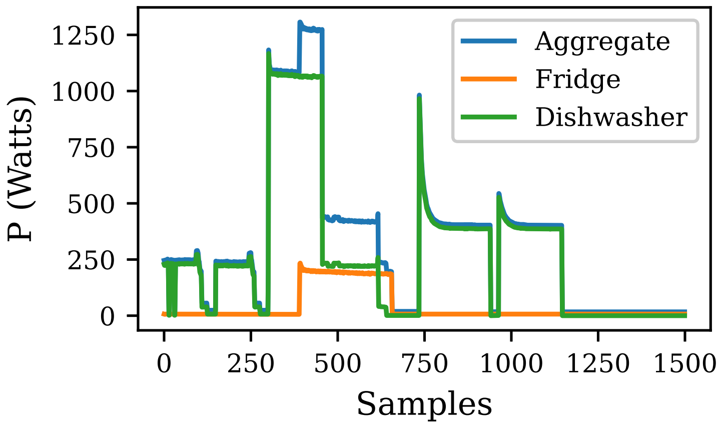


Figure 3.1: Figure showing an example of the aggregate power signal when only the fridge and dishwasher are used.

Hz. The choice of used features is made to reduce the complexity of our classification model. In this section, we discuss the data and features used to construct our model and then describe the mini-batch classification method.

3.3.2 Data

We used the REDD dataset to test our methods. The Reference Energy Disaggregation Data Set (REDD) [3] is an open dataset released by MIT for NILM research. The authors collected data from six houses located in Massachusetts, USA. The dataset contains high-frequency and low-frequency measurements. The high-frequency measurements represent the mains' current and voltage waveforms. The low-frequency measurements comprise power measurements for the mains and individual circuits.

For this work we considered Household 1 which contains active power measurements over a period of 8 days. We used 80% of the signal for training and 20% for testing. To compare our results with other methods, we considered 4 appliances (fridge, washing machine, dishwasher, and microwave). Figure 6.2 shows the ative power signal of each considered appliance during one day for household 1 and the resulting aggregate signal. To be more specific, we report in Table 3.1 the number of operating cycles used for training and prediction.

Table 3.1: Number of operating cycles used for training and prediction

	N° of operating cycles for training	N° of operating cycles for prediction
Fridge	513	153
Washing m.	55	31
Dish w.	102	33
Microwave	421	113

3.3.3 Feature extraction

A low frequency near real-time NILM should be able to identify household appliances and estimate their energy usage by processing as little data as possible. Therefore, the challenge is to find low computational cost and discriminative features, which can produce acceptable mini-batch classification performance.

For each appliance, active power signal S_i^k $(i = 1, 2, ..., n)$ $(k = 1, 2, ..., K)$ with n the total number of samples in the signal "S" and K the total number of appliances. A window W of size "L" is used to compute various window statistics. The window size value $L = 100$ was determined empirically. Windows with smaller size increase the identification error. On the other hand, a larger L value affects negatively the energy disaggregation. We choose window statistics, which are: mean, standard deviation (std), maximum (max), minimum (min), median (med) and variance (var). The obtained vectors are V_{mean}, V_{std}, V_{min}, V_{max}, V_{med}, V_{var}:

$$V_{parameter,p} = parameter(W_p) \tag{3.2}$$

$$W_p =< S_{L(p-1)+1} ... S_{L.p} > \tag{3.3}$$

$$parameter = \{mean, std, min, max, med, var\} \tag{3.4}$$

$$P = \frac{n}{L} \tag{3.5}$$

With W_p ($p = 1, 2, ..., P$) the p_{th} window calculation and P the total number of window calculations. A feature matrix X (see equation 3.6) is created with the obtained vectors V_{mean}, V_{std}, V_{min}, V_{max}, V_{med}, V_{var}. The target vector y (see equation 3.7) contains the class label k_p for each training instance X_p . The data matrix X with the target vector y give the training matrix M as shown in equation 3.8.

$$\mathrm{X} = [V_{mean}, V_{std}, V_{min}, V_{max}, V_{med}, V_{var}] \tag{3.6}$$

$$y =< k_1 ... k_p > \tag{3.7}$$

$$M = [\mathrm{X}, y] \tag{3.8}$$

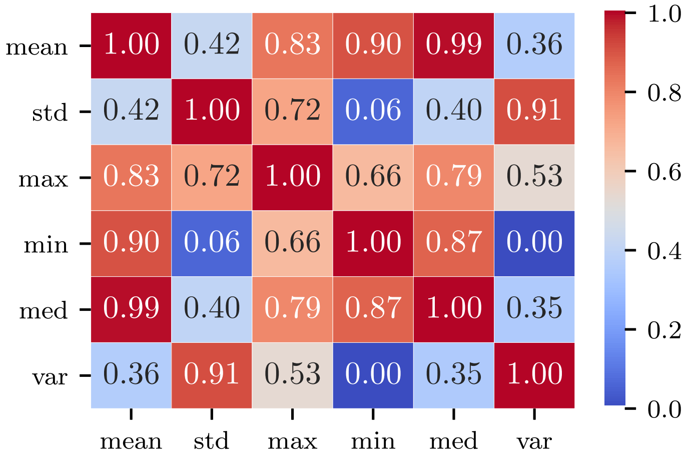
3.3.4 Feature selection

To choose the best discriminating features, we used a feature selection method based on mutual information estimation [65], we then select the 3 most informative features. The mutual information measures the dependency between two variables and is equal to zero if the two variables are independent, higher values mean higher dependency. In our case, the mutual information (MI) is computed between each feature and the target variable y. Table 3.2 shows that the standard deviation (std), mean and maximum (max) values of the aggregate power in each window are the best discriminative features.

Table 3.2: Results of the mutual information estimation for each feature $V_{parameter,i}$ with the target variabe y

Feature	std	mean	max	min	med	var
MI score	1.2	1.26	1.32	0.48	1.04	1.19

To further reduce the complexity of our model, we used a correlation test. The correlation is calculated between each pair of features, among standard deviation (std), mean and maximum (max) values of the aggregate power. Then, the resulting correlation matrix is displayed as a heat map, as shown in Figure 3.2. The higher correlation values are colored with warmer colors. We select the less correlated pair of features to increase the classification performance. As shown in Figure 3.2 the less correlated pair of features is std and mean values of the aggregate power in each window, which have a correlation coefficient of 0.42.



	mean	std	max	min	med	var
mean	1.00	0.42	0.83	0.90	0.99	0.36
std	0.42	1.00	0.72	0.06	0.40	0.91
max	0.83	0.72	1.00	0.66	0.79	0.53
min	0.90	0.06	0.66	1.00	0.87	0.00
med	0.99	0.40	0.79	0.87	1.00	0.35
var	0.36	0.91	0.53	0.00	0.35	1.00

Figure 3.2: Heatmap for std, mean, min, max, med, var.

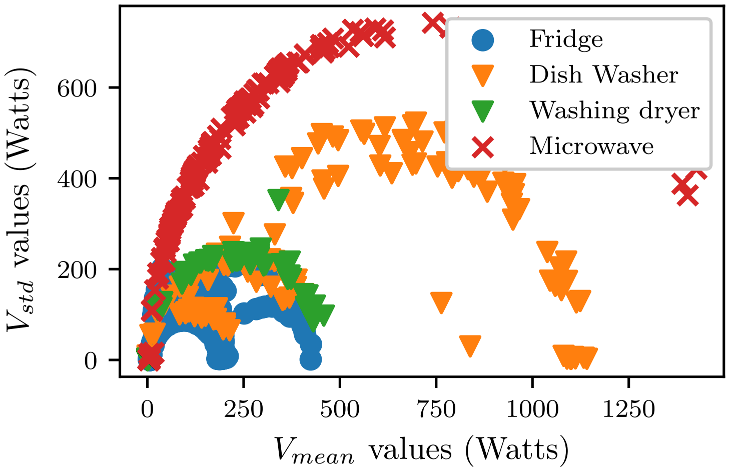


Figure 3.3: Scatter plot of mean power versus std of power for each appliance.

3.3.5 Training

After feature selection, by removing the discarded features, the matrix X becomes:

$$X = [V_{mean}, V_{std}] \tag{3.9}$$

A KNN classifier [66] is trained with the matrix M. Figure 3.3 shows a scatter plot of the classes to be learned by the KNN classifier.

3.3.6 Prediction

For prediction, a synthetic aggregate power signal S_{agg} is created by summing appliance-level power signals S_i^k, as shown in equation 3.10. With $i = 1, 2..., n$ and n the number of samples, k is the number of appliances (class labels). We use a synthetic aggregate signal due to the problem of lack of synchronization between the total load (i.e. aggregate) and each appliance's ground truth data due to the different sample rates [67]. Furthermore, to avoid using complex event detection algorithms, which need a lot of computing power, we use a constant threshold of (Thr). We choose a threshold value of the $Thr = 10$VA as we find that this value avoids false events due to noise.

$$S_{agg(i)} = \sum_{k=1}^{K} S_i^k \tag{3.10}$$

The features are then extracted using the windowing method described in subsection 3.3.3. The obtained feature matrix X_{agg} is fed to the KNN for prediction. The prediction vector $\hat{y}$ returned by the KNN algorithm gives the predicted appliance class label $\hat{k}_p$ for each value $X_{agg(p)}$ (see equation 3.11) of the aggregate feature matrix X_{agg}.

$$X_{agg(p)} = [V_{mean(p)}, V_{std(p)}] \tag{3.11}$$

$$\hat{y} = < \hat{k}_1 ... \hat{k}_p > \tag{3.12}$$

The prediction process can be summarized as follows: each L seconds, a features value $X_{agg(p)}$ is computed and fed to the KNN model. A class label $\hat{k}_p$ is then returned by the KNN, which corresponds to the current active appliance. Therefore, each L seconds, the system returns a feedback vector F_p containing the detected appliance $\hat{k}_p$ and its power consumption $V_{mean(p)}$:

$$F = [\hat{k}_p, V_{mean(p)}] \tag{3.13}$$

3.4 Results and discussion

In this section, the evaluation metrics for the proposed approach are defined and the obtained results are discussed.

The field of NILM lacks standard (or commonly adopted) metrics for the evaluation of the algorithms, making fair comparison difficult [15]. To evaluate the results of our approach, we choose the three most common performance evaluation metrics used in the literature [15], which are accuracy (Acc), $F1 - score$ and the Total Energy Correctly Assigned ($TECA$).

$$Acc = \frac{TP + TN}{TP + FN + TN + FP} \tag{3.14}$$

$$F1 - score = \frac{2 \times Pr \times R}{Pr + R} \tag{3.15}$$

$$Pr = \frac{TP}{TP + FP} \tag{3.16}$$

$$R = \frac{TP}{TP + FN} \tag{3.17}$$

The true positive parameter TP represents the number of samples that have been correctly classified or, more precisely, the power quantity correctly assigned to that device. The false-positive parameter FP represents the number of samples that have been incorrectly classified or, more precisely, the power quantity incorrectly assigned to that device. The false-negative parameter FN represents the number of samples that should be but have not been classified or, more precisely, the power quantity that should have been assigned to that device but has been assigned to another or has

Table 3.3: Performance comparison between our method and the Seq2point method [6].

Metrics	Methods	Microwave	Fridge	Dish w.	Washing m.	Overall
MAE (%)	seq2point	28.199	28.104	20.048	18.423	23.693
	our method	4.138	5.540	11.565	2.964	6.052
SAE (%)	seq2point	0.059	0.180	0.567	0.277	0.270
	our method	0.125	0.091	0.263	0.001	0.120

not been assigned at all. The precision parameter (Pr) measures the portion of power samples that have been correctly classified among the power samples assigned to a given device. The recall parameter (R) measures what power portion of a given device is correctly classified in general, also considering the samples that would belong to that device but have been wrongly assigned to another or not assigned at all.

Therefore, the accuracy Acc measures how well each appliance is detected and the F-score combines the results obtained through the precision and recall analysis.

$$TECA = 1 - \frac{\sum_{i=1}^{n}\sum_{k=1}^{K} \mid \hat{S}_i^{(k)} - S_i^{(k)} \mid}{2 \times \sum_{i=1}^{n} S_{agg}(i)} \tag{3.18}$$

The total energy correctly assigned (TECA) measures the portion of power correctly attributed to all devices. $\hat{S}_i^{(k)}$ is the separated appliance signal, $S_i^{(k)}$ is the original appliance signal, $S_{agg}(i)$ is the observed aggregate signal, K is the total number of appliance signals and n is the number of samples.

Table 3.3 shows a comparison between the proposed method and a state of the art method (seq2point) [6] which uses a deep learning approach. We use the same metrics used by the author of the seq2point method to be able to compare the performance with our method. These are: the mean absolute error (MAE) (see equation 6.6) and the normalised signal aggregate error (SAE) (see equation 3.20).

$$MAE = \frac{1}{n}\sum_{i=1}^{n} \mid \hat{S}_i - S_i \mid \tag{3.19}$$

$$SAE = \frac{\mid \hat{E} - E \mid}{E} \tag{3.20}$$

The mean absolute error (MAE) gives the error in power estimation for a given appliance at each time step. The normalised signal aggregate error (SAE) gives the error in energy estimation for a given appliance. With $E = \sum_{i=1}^{n} S_i$ being the ground truth energy and $\hat{E} = \sum_{i=1}^{n} \hat{S}_i$ being the estimated energy.

From Table 3.3, we see that the proposed method shows better overall performance. The mean absolute error (MAE) is lower for each appliance.The normalized signal aggregate error (SAE) is lower for each appliance except the microwave. In addition, our method has a lower time complexity for prediction, therefore lower computational cost. The traditional KNN algorithm uses a linear search method to find the K nearest neighbors. Therefore, the time complexity is $O(nd)$, where n is the size of the training dataset and d is the dimensionality, as the complexity, in this case, is proportional to the size of the training dataset [68]. To reduce the time complexity of our model's prediction, we use a KD Tree algorithm [69] to find the K nearest neighbors. This reduces the complexity to $O(log(n))$ [69]. On the other hand, the seq2point method which is based on a deep convolutional neural network, has a complexity of at least $O(n^2)$ [70].

Table 3.4: Results of the performance evaluation

	F1-score	Accuracy	$E(\%)$	$(\hat{E})(\%)$
Microwave	0.867	0.981	17.170	15.008
Fridge	0.562	0.463	51.422	46.737
Dish w.	0.667	0.957	25.970	32.811
Washing m.	0.562	0.981	5.435	5.441
TECA	0.931			

Table 3.4 summarizes the performance of the proposed method. The average $F-score$ shows that our classification model has a good overall performance. The microwave is better discriminated by the classifier than the other appliances, as depicted by its corresponding $F1-score$. This is due to its high apparent power consumption and consumption patterns which are not common to other appliances. This results

in very different values of standard deviation. On the other hand, the other appliances (fridge, washing machine and, dishwasher) share some consumption patterns which are difficult to discriminate. This translates to overlapped clusters in the feature space as shown in Figure 3.3. Also, 93% of the total energy is correctly attributed as indicated by the TECA value.

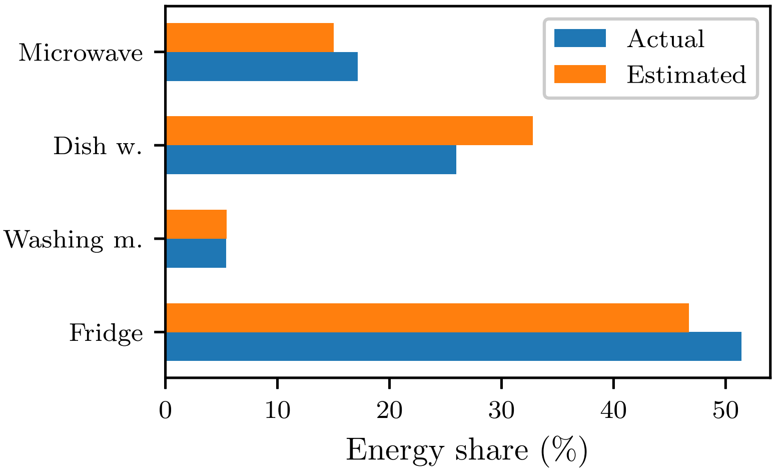


Figure 3.4: Comparison between predicted and ground truth energy breakdowns.

Figure 3.4 shows a comparison between the actual energy breakdown (E) and the estimated energy breakdown after disaggregation ($\hat{E}$). We observe that the microwave and washing machine have their energy better estimated than the dishwasher and the fridge. In fact, the fridge and dishwasher have similar consumption patterns from 0 to $500V.A$ which makes their detection challenging.

3.5 Conclusion

In this chapter, we presented our approach to load disaggregation using multiclass classification. We used a sliding window to extract mini-batches of data samples which we then used to compute features. We used ground truth data to train the KNN classifier and the aggregate power data to test the classification performance of the KNN algorithm. The obtained results show good overall disaggregation performance. However, this approach has limitations when it comes to the disaggregation of multiple loads working simultaneously. In this case, the presented approach will tend to attribute the power to the most power-consuming load. In simple scenarios, where high-power-consuming loads are not used at the same time, this limitation will not considerably affect the disaggregation performance. For a practical application of NILM, the disaggregation method should take into account the case of concurrent loads that are used in the same time interval. In the next chapter, we will present a disaggregation method that tackles this limitation.

CHAPTER 4

MULTI-LABEL CLASSIFICATION FOR LOAD DISAGGREGATION

This chapter is based on a journal article co-authored by Mourad Adnane and Mourad Haddadi [Sahrane et al., Electric Power Systems Research, 2021 [71]]

Contents

4.1 Introduction . . . 49

4.2 Data . . . 49

4.3 Disaggregation method . . . 49

4.3.1 Filtering . . . 50

4.3.2 Discretization . . . 51

4.3.3 Event detection . . . 52

4.3.4 Features extraction . . . 53

4.3.5 Classification . . . 53

4.3.6 Data labeling . . . 54

4.4 Performance evaluation . . . 55

4.5 Results and discussion . . . 56

4.5.1 Preprocessing results . . . 56

4.5.2 Event detection results . . . 57

4.5.3 Labeling results . . . 57

4.5.4 Disaggregation performance . . . 57

4.6 Conclusion . . . 61

4.1 Introduction

In the previous chapter, we presented our first learning model for load disaggregation based on multiclass classification. As discussed previously, this method gives good results when there is little overlap between loads' signatures but performs poorly as the degree of overlap increases. To overcome this limitation, we experimented with another type of classification that can predict multiple labels for each data instance; this type of classification is called multi-label classification. In this chapter, we will describe our approach to load disaggregation using multi-label classification.

4.2 Data

We used the REDD datasetto test our methods. In this work, we used the low-frequency circuit-level data from all six houses. For each habitation, we computed the synthetic aggregate signal by summing individual ground-truth power signals. We adopted this approach to create different load targeting scenarios, as shown in the following sections. The proposed disaggregation method was trained on 60% of data and tested on 40% of data for each house.

4.3 Disaggregation method

To study the effect of non-targeted loads on NILM performance, we develop a disaggregation method based on multi-label classification. We choose the multi-label approach because it allows inferring a group of labels instead of one unique label, which is more convenient to detect loads working simultaneously. Figure 4.1 shows the different steps of the proposed disaggregation method. We first pre-process the aggregate signal with a filtering step followed by a discretization step. Then, we use an event detection method to detect operational state changes in the aggregate signal to extract appropriate features. We use a labeling method to label the ground truth data that we use to train and test a classifier in the classification step.

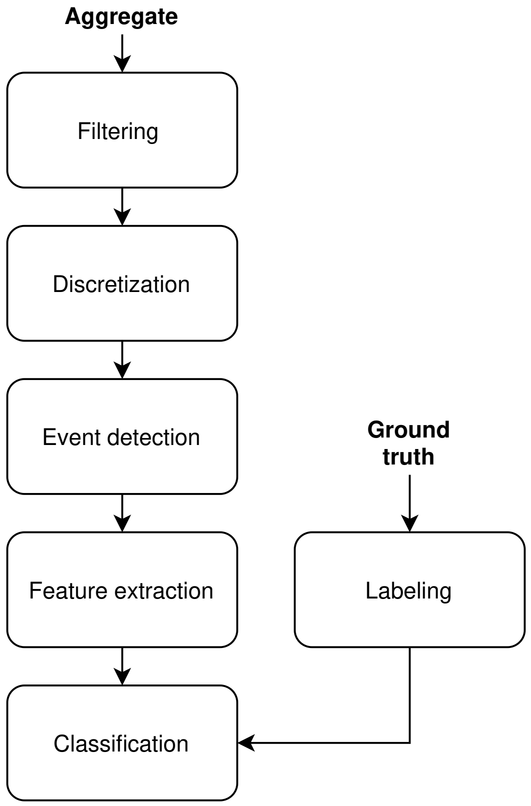


Figure 4.1: Block diagram of the proposed NILM approach.

4.3.1 Filtering

We use a median filter as a noise reduction method in our pre-processing step. We use this step to remove high-power transients that can result in brief noisy steady states after the discretization step. The median filter works by replacing each sample of the signal with the median of its neighboring samples. We apply the median filter to the vector S_{agg} which contains O samples of active power measurements. A median filter $med()$ characterized by a window W of size $2L+1$ is applied to S_{agg} to obtain the filtered aggregate vector $S_{agg,f}$ as shown below:

$$S_{agg,f}[i] = med([S_{agg(i-L)}, ..., S_{agg(i+L)}]) \tag{4.1}$$

With $i = 0, ..., O-1$. Because the i^{th} sample has to be centered in i, the first L and last L samples are not considered in the filtering process. Thus, the size of the resulting filtered vector $S_{agg,f}$ equals $O-2L$.

4.3.2 Discretization

Discretization works by quantizing attributes. The quantization process maps values from a large set of values into a smaller one. Different discretization methods exist in the literature. The authors in [72] give a review of existing methods. In our case, we apply discretization to the aggregate power signal's amplitude to filter false event alarms. In this work, we use the *equal-width binning* method [72] that works by discretizing the continuous attribute by creating a specified number of bins n_{bins}. We obtain the bins' width w_{bins} by dividing the range of the variable by the number of bins, as shown in equation 4.2. We calculate the bin's edges e by applying equation 4.3. Each value of the transformed feature $S_{agg,d}$ is equal to the average of the two bin edges, as shown in equation 4.4.

$$w_{bins} = \frac{S_{agg,f}^{max} - S_{agg,f}^{min}}{n_{bins}} \tag{4.2}$$

$$e[k] = S_{agg,f}^{min} + (k \times w_{bins}) \tag{4.3}$$

with $k = 0, 1, ..., n_{bins}$

$$S_{agg,d}[k] = \frac{e[k] + e[k+1]}{2} \tag{4.4}$$

with $k = 0, 1, ..., n_{bins} - 1$

We use the discretization step to reduce the variability of the filtered aggregate vector $S_{agg,f}$. Reducing the variability can be seen as absorbing small power variations. Otherwise, these power variations would necessitate an adaptive event detection method like the one used in [73]. This type of approach can be challenging to implement because it is difficult finding the appropriate parameter values. We find it more convenient, in our case, to filter the unwanted events with discretization before the event detection step. Figure 4.2 shows the discretization of a filtered aggregate segment using two different number of bins n_{bins}. When using $nbins = 5$, we observe missing events and inaccurate power values.

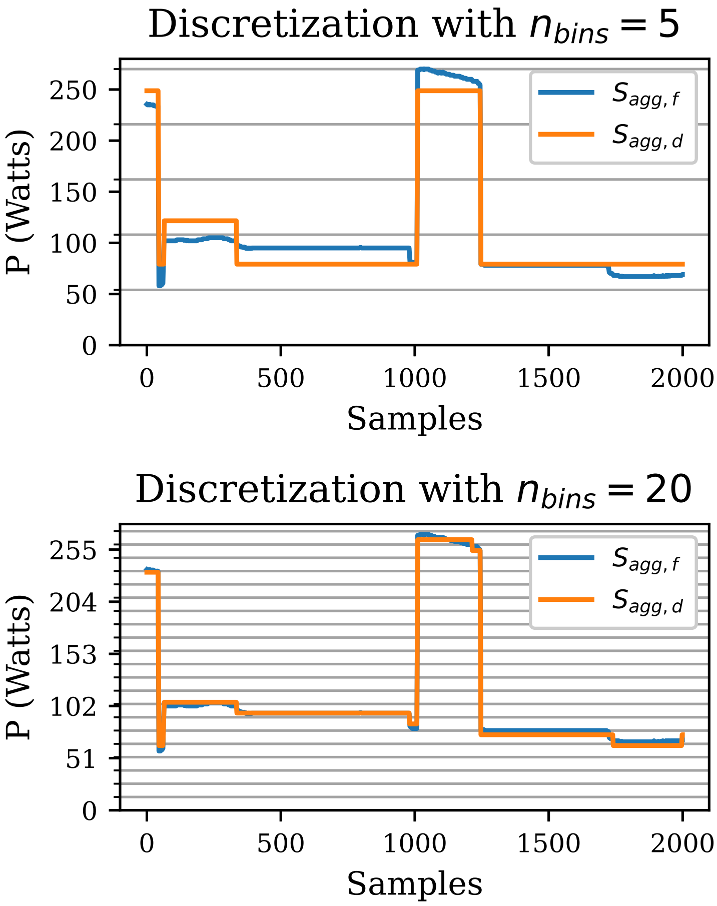


Figure 4.2: Discretization of the filtered aggregate signal $S_{agg,f}$ produces the signal $S_{agg,d}$. The horizontal lines represent the bins' edges e. The spaces between the horizontal lines represent the bins' width w.

4.3.3 Event detection

As mentioned in section 4.3, an event detection step is needed to detect when loads change their power consumption state. Our event detection method consists of differentiating the discretized aggregate vector $S_{agg,d}$ to obtain an event detection vector E, as shown in equation 4.5.

$$E[j] = S_{agg,d}[k+1] - S_{agg,d}[k] \tag{4.5}$$

The change point indices are the indices k, where the detection vector E is not equal to zero, as shown below:

$$C[g] = k \wedge E[k] \neq 0 \tag{4.6}$$

with $g = 1, 2, ..., G$, such as G is the number of detected events.

4.3.4 Features extraction

The features considered in this work are the steady-state aggregate power P_{agg}, the operational state change ΔP_{agg} and the operational state duration T. The feature extraction process is described in equations 4.7, 4.8, and 4.9.

$$P_{agg}[g] = S_{agg,d}[C[g]] \tag{4.7}$$

$$\Delta P_{agg}[g] = |P_{agg}[g+1] - P_{agg}[g]| \tag{4.8}$$

$$T[g] = C[g+1] - C[g] \tag{4.9}$$

The steady-state aggregate power P_{agg} represents the discretized aggregate value $S_{agg,d}$ when we detect a change at the index $C[g]$. The operational state change ΔP_{agg} is the power difference between two consecutive states $P_{agg}[g]$ and $P_{agg}[g+1]$. $T[g]$ represents the duration of each operational state $P_{agg,d}[g]$. We calculate it by the change point indices $C[g]$ and $C[g+1]$ by differentiating them.

4.3.5 Classification

For classification, we use the classifier chain model (CC) [74]. The CC model is one of the problem transformation approaches to multi-label classification. In this type of approach, we transform the multi-label problem into one or more single-label problems. This transformation allows employing single-label classifiers to make single-label classifications that are then transformed back into multi-label representations. While the single-label or multiclass classifier associates an instance x with a single label y from the set of labels Y, the multi-label classifier instead associates a subset of labels $S_y \subseteq Y$.

Our model uses steady-state features [1]. The multi-label classification approach is more adapted to our problem because more than one load can contribute to a given operational state (steady-state).

The classifier chain (CC) makes use of $|Y|$ binary classifiers with $|.|$ representing the cardinality of the set of labels Y . We associate each binary classifier C_p with a single label $y_p \in Y$. The $0/1$ label association of previous classifiers extends the feature space of each classifier to create a linked chain of classifiers. This chaining method allows the CC model to consider label correlations and therefore avoids predicting implausible label associations.

We use the random forest classifier (RF) [75] as the base single-label classifier C_p. In [76], the authors also used the RF algorithm in the context of a multi-label classification for NILM. The RF is an ensemble learning method used for classification and regression modeling. The RF algorithm builds a set of trees on random samples of the learning data. In each step of the algorithm, a learning sample is drawn randomly from the learning data. An individual tree is then grown on each sample's element. Combining the predictions of all the trees gives superior prediction accuracy comparing to single classification or regression trees [77].

4.3.6 Data labeling

The used dataset, REDD, gives as ground truth the active power data of each appliance. However, the dataset doesn't provide the indices where each load is turned ON or OFF. We apply a labeling step on each ground truth power signal to obtain binary ground truth labels for each targeted electrical load.

To label the ground truth data, we use an adaptive threshold due to background noise changes over time which may cause labeling errors. The resulting labels matrix A has a dimension of $(O - 2L) \times N$ with N the number of targeted loads, O the size of the aggregate vector S_{agg}. Each element of the matrix takes a value of $0/1$ to represent the absence or presence of the corresponding load in the aggregate signal S_{agg}.

Also, we use an adaptive threshold Thr to account for low power consumption steady states that can not be detected using a constant thresholding method. Labeling the absence/presence of a given load is equivalent to estimating the value of the OFF state of the targeted electrical load. The OFF state value varies across loads. Therefore, an adaptive estimation method is necessary. The OFF state is the most common in the ground truth signal of the targeted loads. It is, therefore, possible to use the mode of the ground truth data to estimate the threshold Thr as expressed in equation 4.10.

$$Thr(n) = mode(med(X_g^{(n)})) + b \tag{4.10}$$

Where n stands for the target number with $n = 1, 2, ..., N$ (N is the total number of targets) and X_g the ground truth signal, which is first smoothed using a median filter to reduce noise and remove spikes that can cause false events. The constant b accounts for the small fluctuations due to measurement noise.

In the REDD dataset (the one we use in this work) and in most NILM datasets, a dominant OFF state characterizes the ground-truth signals. This fact justifies the choice of our labeling method, where we use the mode of ground truth data to estimate the threshold. However, in the case where there is no dominant OFF state, we could use a clustering algorithm in which each cluster represents a consumption state of the appliance we want to label.

4.4 Performance evaluation

We define the detection performance metric as the average of the micro and macro F1 scores, as shown below:

$$F1(TP, FP, FN) = \frac{2.TP}{2.TP + FP + FN} \tag{4.11}$$

$$F1_{Micro} = F1(\sum_{n=1}^{N} TP_n, \sum_{n=1}^{N} FP_n, \sum_{n=1}^{N} FN_n) \tag{4.12}$$

$$F1_{Macro} = \frac{1}{N} \sum_{n=1}^{N} F1(TP_n, FP_n, FN_n) \tag{4.13}$$

$$Pr = \frac{F1_{Micro} + F1_{Macro}}{2} \tag{4.14}$$

The micro F1, $F1_{Micro}$ and macro F1, $F1_{Macro}$ are label-based averaging methods for calculating the $F1$ score across labels [7]. The $F1_{Micro}$ gives an overall performance indication by computing the $F1$ score globally. The $F1_{Macro}$ computes the average of the $F1$ scores of each label n or targeted load in the context of NILM. The $F1$ score varies between "0" and "1". Where "1" means perfect discriminative performance and "0" no discriminative capability of the classifier. True positives (TP_n) account for the number of times the classification algorithm correctly predicts the presence of the label n. False positives (FP_n) account for the number of times the classification algorithm wrongly predicts the presence of the label n. False negatives (FN_n) account for the number of times the classification algorithm wrongly predicts the absence of the label n.

4.5 Results and discussion

4.5.1 Preprocessing results

Figure 4.3 shows the results of the preprocessing step. We obtained the filtered signal using a median filter with a window size $W = 100$ samples. We found that smaller W values result in more noise, and larger values do not give better results. We used the median filter to reduce the noise in the aggregate power signal. The discretization gives constant steady states necessary to extract correct state durations during the features extraction step. We obtained these results using the KBinsDiscretizer algorithm [78] with a number of bins $n\_bins = 100$, an *ordinal* encoding, and a $uniform$ strategy. n_bins is the number of bins generated by the algorithm, the encoding method determines how the bin's identifier is generated, and the strategy defines the method for obtaining the widths of each bin [79]. We noticed that the number of events in the discretized signal is proportional to the number of bins used. Thus, increasing the number of bins *n_bins* increases the number of false alarms while smaller values resulted in missed events.

4.5.2 Event detection results

Figure 4.4 shows the detection signal obtained as defined in section 4.3. The preprocessing step ensures that the changes detected are not due to noisy fluctuations of the aggregate signal. Therefore, there is no need to use a threshold to filter noisy changes.

4.5.3 Labeling results

Figure 4.5 shows a comparison between the constant threshold labeling method and our method described in section 4.3. We can see that the fixed threshold method doesn't detect parts of the electrical heater ground truth's signal that are lower than the threshold, while our proposed approach successfully detects them. Therefore, using a constant decision threshold value results in mislabeling of the ground truth loads. Because some loads may have consumption states that are lower than a defined constant threshold. Using mislabeling data results in poor learning capabilities for the system. For this test, we used a fixed threshold $Thr = 50W$, and for the adaptive thresholding method, a value for the constant b equal to $5W$. We used the value $b = 5W$ because the noise level during the OFF states does not exceed $5W$ for each of the ground truth signals.

4.5.4 Disaggregation performance

In table 4.1, we give a performance comparison between our method and the multi-label consistent deep dictionary learning (MLCDDL) method [7]. The authors of this method did not mention whether their results were obtained by integrating non-targeted loads or not. We reported our results in the absence and presence of non-targeted loads to evaluate our method's performance in both cases. The comparison shows that our disaggregation method presents better results in both cases

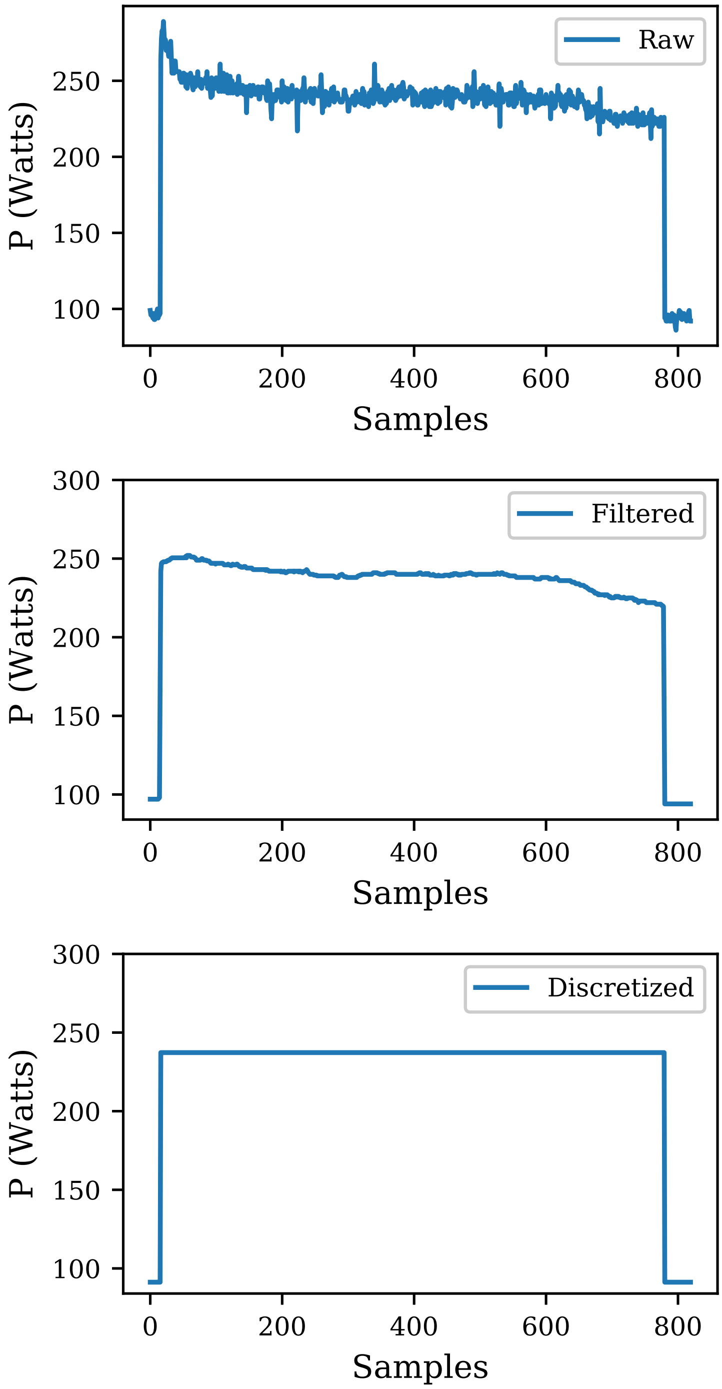


Figure 4.3: Effects of median filtering and discretization on the raw aggregate signal. The raw aggregate power signal is first filtered than discretized.

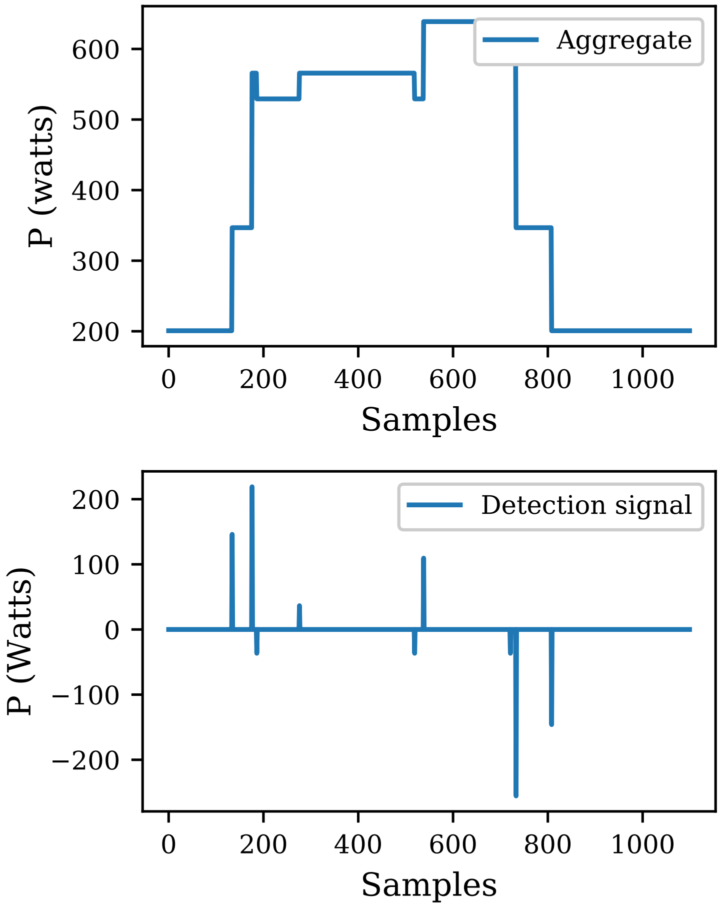


Figure 4.4: Detection signal used for event detection, obtained by differentiating the filtered and discretized aggregate signal.

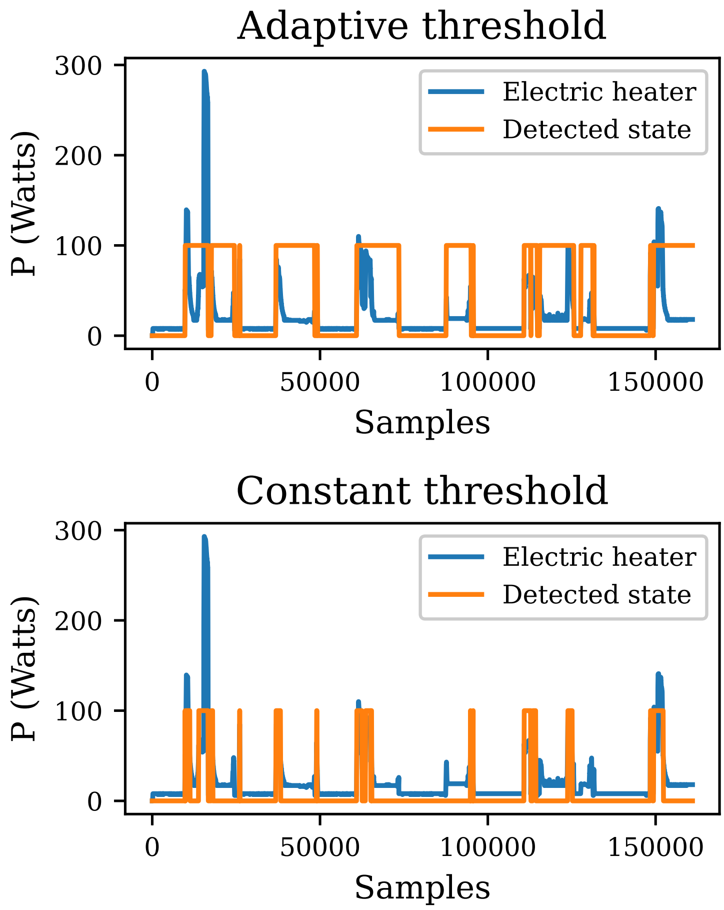


Figure 4.5: Comparison between the constant and adaptive thresholding methods. The adaptive labeling method detects (high state) all the operating cycles while the constant method detects only parts of the signal which are greater than the constant threshold $Thr = 50W$.

Table 4.1: Performance comparison between our method and the MLCDDL method [7].

Load	**Our method**		**MLCDDL**
	F1-score (with non-targets) (%)	**F1-score (without non-targets) (%)**	**F1-score (%)**
Dishwasher	42.96	**67.16**	**56.97**
Lighting	**84.34**	**93.69**	69.07
Washing.M	**65.36**	**93.74**	56.48

(absence and presence of non-targets) for the washing machine and lighting. For the dishwasher, our disaggregation method gives better results only when we exclude non-targeted loads.

4.6 Conclusion

In this chapter, we presented our multi-label approach to NILM. Our proposed NILM method uses the classifier chain algorithm and the random forest classifier (RF) as the base single-label classifier. Using multi-label classification allows the disaggregation of loads that work at the same time. We introduced our preprocessing approach using discretization and our ground-truth labeling method. We found that our adaptive labeling method gives better results than the constant threshold method.

CHAPTER 5

A CLUSTERING-BASED EVENT DETECTION METHOD FOR NILM

This chapter is based on a conference article co-authored by Mourad Adnane and Mourad Haddadi [Sahrane et al., 6th International Symposium on New and Renewable Energy, IEEE, 2021 [80]]

Contents

5.1 Introduction 63
5.2 Background 63
5.3 Data 65
5.4 Methods 65
5.4.1 Filtering 66
5.4.2 Data clustering 66
5.4.3 Events extraction 68
5.4.4 Evaluation metrics 68
5.5 Results and discussion 69
5.6 Conclusion 73

5.1 Introduction

Event detection in the context of NILM refers to the process of detecting operational state changes (or transitions) in power consumption. The event detection step is then followed by a feature extraction step as shown in figure 5.1. The type of features extracted varies depending on the granularity of data (high or low-frequency) and the NILM method requirements. In this chapter, we present our event detection method based on clustering aggregate data samples. The remainder of this chapter is structured as follows. Section 5.2 discusses recent event detection methods. Section 5.3 describes the data used in this work. Section 5.4 presents our event detection method. Section 5.5 discusses our obtained results. Section 5.6 concludes this paper.

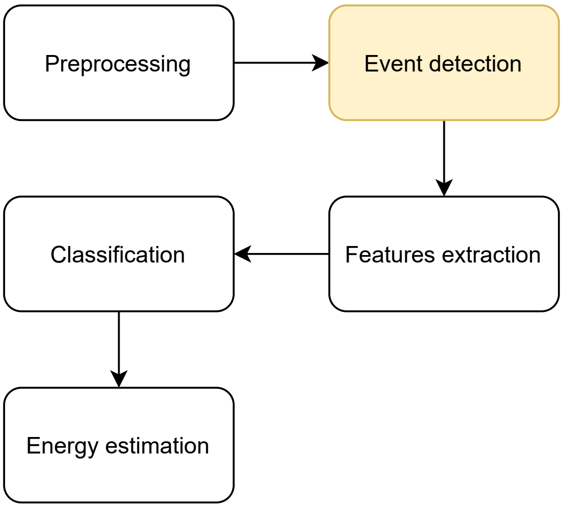


Figure 5.1: Block diagram showing the steps of an event-based NILM.

5.2 Background

Different event detection approaches exist in the NILM literature. These methods are classified into three categories in [73], namely those based on: expert heuristics, probabilistic models, and matched filters. Expert Heuristics are rule-based methods that expect the aggregate signal to vary between a predetermined range of values with a specified tolerance [1]. In [81], the "state change detection" rule compares the difference series of the aggregate signal to predetermined ranges to detect "ON" and "OFF" events. Probabilistic models are statistical approaches used to detect abrupt changes in time series. In statistics and signal processing literature, these techniques are called *change detection* or *change-point detection*. Several of these methods are used in the NILM literature, like the Generalized Likelihood Ratio (GLR) in [82] and Goodness of Fit (GoF) statistic in [83]. Matched filters work by correlating a mask which is the known signal of the event to be detected, with the unknown aggregate signal. These masks are specific events (like ON/OFF events) extracted from the ground truth of each targeted load. For instance, in [84], a multi-resolution matched filter is used to detect events in the spectral envelope derived for the aggregate power signal. Recently, hybrid event detection methods have merged as a new category. This type of method uses a combination of the three categories described above. For instance, in [85], an event detection method based on the standard deviation of the signal's envelope is proposed. The algorithem tracks the variation of the statndard deviation of the current signal envelope using a sliding window. A threshold is used to reject the events detected during steady states and retain only events which are responsible for new steady states. In [86] the event detection problem is treated as a segmentation problem where the method tries to segment stationary and non-stationary intervals of the aggregate signal. The method uses the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [87] to separate stationary and non-stationary segments. In [88], the authors combines power difference of two consecutive samples and standard deviation of a pre-defined number of samples (i.e., the width of moving window) for their event detector. The proposed algorithm works by comparing the standard deviation value computed on each window with the power difference of each two consecutive samples. Then, a threshold value is used to accept or reject the event under test. The method proposed in [89] combines a base algorithm based on moving average change with a time limit and two auxiliary algorithms based on derivative analysis and filtering analysis. The base algorithm is used to detect the events in the aggregate signal. The auxiliary algorithms are used to improve the detection performance by removing noisy detections of the base algorithm.

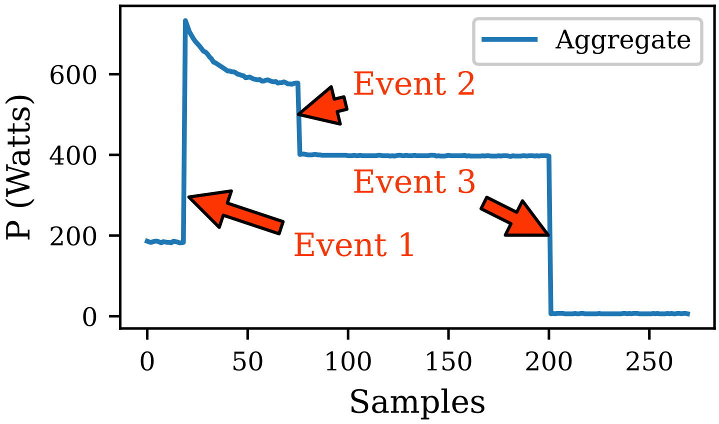


Figure 5.2: Example of operational state changes events in the aggregate power signal of a household.

5.3 Data

We used REDD dataset to test our methods. We use the ground truth data of the oven, fridge, dishwasher, washing machine, microwave, and bathroom outlet. We then construct a synthetic aggregate data by summing all the ground truth signals.

5.4 Methods

Our approach to event detection identifies specific change-points amongst the aggregate signal's samples. We are only interested in identifying change-points that lead to another consumption state of a given appliance/load. Small variations in the aggregate signal are not considered change-points and must therefore be ignored by our method. Figure 5.2 shows an example of operational state changes events in the aggregate power signal of a household. In the remainder of this work, we use the term "event" and "change-point" interchangeably for readability concerns. Here we describe the steps we followed to design our method as shown in figure 5.3. First, we apply a median filter on the aggregate power signal, then use a data clustering step to segment the aggregate power signal. In the events extraction step, we extract the indices and power values of each event. Finally, we evaluate our method using the ground truth events with an appropriate evaluation metric.

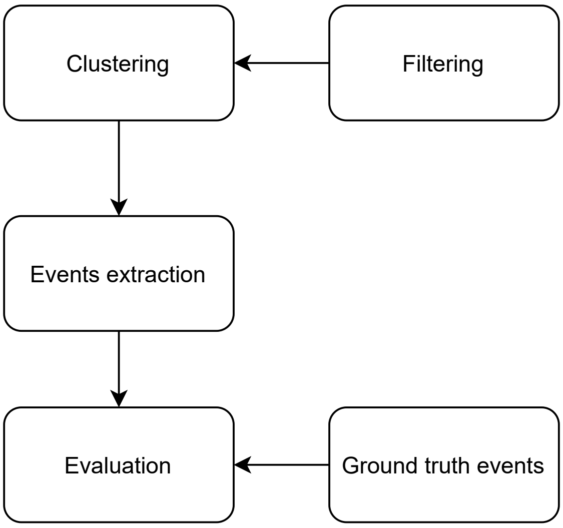


Figure 5.3: Block diagram showing the steps of our event detection method.

5.4.1 Filtering

We apply a median filter on the aggregate signal to attenuate high transient spikes and reduce noise in the operational state portion of the signal. The median filter has a transfer function $med()$ and window size W. The obtained filtered aggregate signal $S_{agg,f}$ is expressed in equation 5.1.

$$S_{agg,f} = med(S_{agg}) \tag{5.1}$$

With S_{agg} representing the raw aggregate signal.

5.4.2 Data clustering

Data clustering is an unsupervised learning method that aims to find patterns in unlabeled data. This method groups similar data instances together into groups or clusters and different data instances into different groups [90]. We apply data clustering to the filtered aggregate signal $S_{agg,f}$ to obtain different segments that we use in the events extraction step of our method. Figure 5.4 shows the segmented filtered aggregate signal after the clustering step.

Formally, the clustering structure is represented as follow:

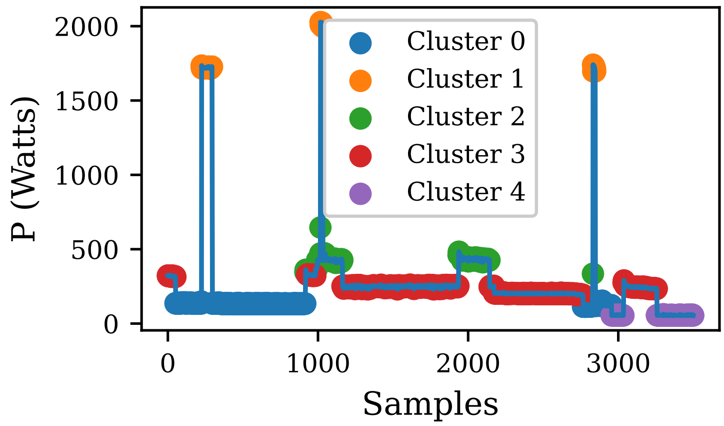


Figure 5.4: Example of the resulting segmentation of the aggregate signal into different steady power consumption states using kmeans clustering algorithm. Each cluster contains samples from the aggregate signal that have similar power values.

$$C = C_1, ..., C_K \tag{5.2}$$

with C a set of subsets of $S_{agg,f}$, such that:

$$S_{agg,f} = \bigcup_{k=1}^{K} C_k \tag{5.3}$$

and:

$$C_k \cap C_l = \emptyset \tag{5.4}$$

for $k \neq l$.

We test our method using three different clustering algorithms, namely, the K-means [91], Mean Shift [92], and Mini-batch K-means [93].We chose these algorithms because they all have only one parameter to tune to find the best possible performance. Also, algorithms with a small number of parameters are usually more computationally efficient. Therefore, more adapted for a practical NILM application. Here, we give a brief description of each considered algorithm.

The K-means algorithm divides the samples into $n_{clusters}$ disjoint clusters C. $n_{clusters}$ needs to be specified as a parameter. Each cluster is described by the mean of samples in the cluster. The K-means algorithm constructs the clusters by minimizing the sum of squared differences between each sample and the cluster's mean (centroid). The

Mean Shift algorithm forms data clusters by shifting a window also called Kernel in the direction of increasing data density. The *bandwidth* is the size parameter of the window across which the mean is computed. The Mini-batch K-means is a modification of the K-means algorithm to reduce computation costs. Instead of treating all the data in one batch, the Mini-batch K-means uses mini-batches which are subsets of the data, randomly sampled in each training iteration. This approach reduces the amount of computation needed to converge to a local solution.

5.4.3 Events extraction

In the events extraction step, we retrieve the indices and values of each event. We consider a candidate sample $S_{agg,f}(t)$ as being an even/change-point $e(t)$ only if it is followed by a sample $S_{agg,f}(t+1)$ that is from a different cluster as shown in the following equations:

$$S_{agg,f}(t) = e(t) \tag{5.5}$$

if and only if:

$$S_{agg,f}(t) \in C_k \tag{5.6}$$

and:

$$S_{agg,f}(t+1) \in C_l \tag{5.7}$$

for $k \neq l$.

5.4.4 Evaluation metrics

Currently, there is no standardized framework for NILM event detection evaluation. Therefore, different performance metrics are employed in the literature to evaluate event detectors. For instance, in [89] the authors used the TPR (true-positive rate), FPR (false-positive rate), and FNR (false-negative rate). In [88] the authors measured TP (true detection), FP (false detection), FN (misdetection) and also computed the precision and recall. In this work, we use the F1-score (equation 5.8) to evaluate the performance of our method. In addition, we use the TPR (true-positive rate) (equation

5.9), FPR (false-positive rate) (equation 5.10), and FNR (false-negative rate) (equation 5.11) to analyze the performance of our method in different scenarios.

$$F1 - score = \frac{2.TP}{2.TP + FN + FP} \tag{5.8}$$

$$TPR = \frac{TP}{E_{gt}} \tag{5.9}$$

$$FPR = \frac{FP}{E_{gt}} \tag{5.10}$$

$$FNR = \frac{FN}{E_{gt}} \tag{5.11}$$

TP, FP, and FN refer respectively to the total number of true positives, false positives, and false negatives. E_{gt} refers to the number of ground truth events that are present in the aggregate power signal. In the context of event detection, true positives (TP) represent the number of true-event detections, meaning that the detected event is present in the ground truth events. False positives (FP) represent the number of false detections of events that do not exist in the ground truth events. False negatives (FN) represent the number of missed events that should have been detected but were not detected.

5.5 Results and discussion

Figure 5.5 shows the effect of the median filter on the aggregate power signal. We observe a significant reduction of the transient spike at the starting of the fridge. In addition, the noise reduction during the operational phase of the fridge's consumption cycle is also observed. To obtain these results, we used a median filter window size $W = 15$ samples. We want to reduce noise and high transients to improve the performance of our method. Noise and high transients result in noisy detections of false events.

Figure 5.6 shows the result of our event detection method. The red markers represent the detected events (change points) in the aggregate power signal. The events in

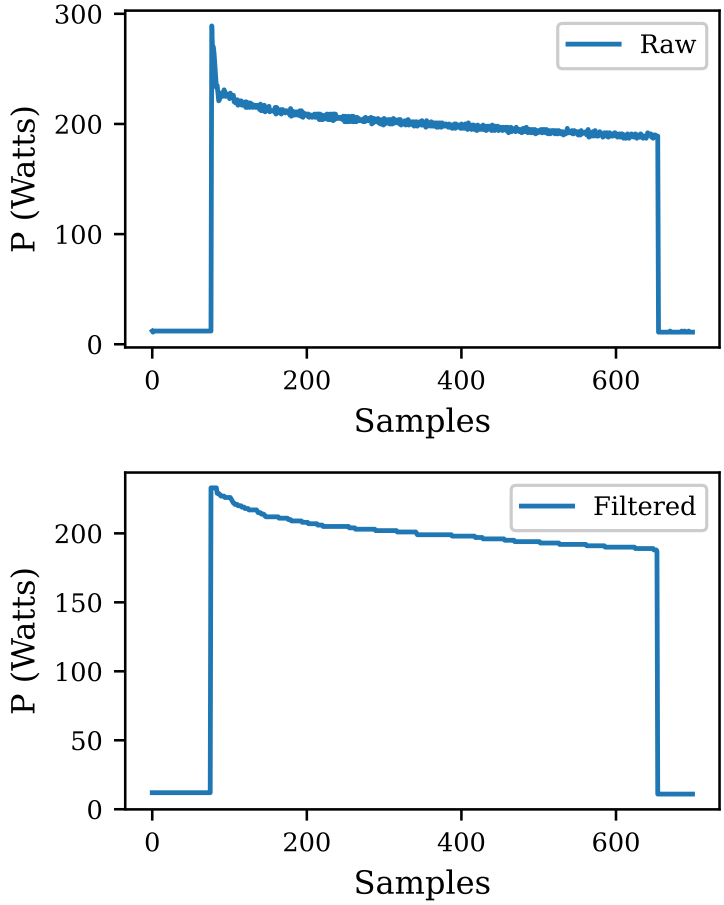


Figure 5.5: Fridge consumption cycle in the aggregate signal before and after applying the median filter.

our context are rapid power transitions in the aggregate power signal like ON/OFF events or operational state changes. We can see that each change point is at the beginning of its corresponding event. We obtained these results using our method with the K-means algorithm and a number of clusters $n_{clusters} = 12$ cluster.

We tested our method using three different clustering algorithms which are K-means, Mean Shift, and Mini-batch K-means. Each algorithm has its strengths and weaknesses, as we will see in the remaining of this section. Table 5.1 presents the performance results of our method when using each of the three clustering algorithms. Figure 5.7 shows the performance results of our method when using the K-means clustering algorithm. We see that increasing the number of clusters improves the detection of true events. This is observed with the increase of the true-positive rate

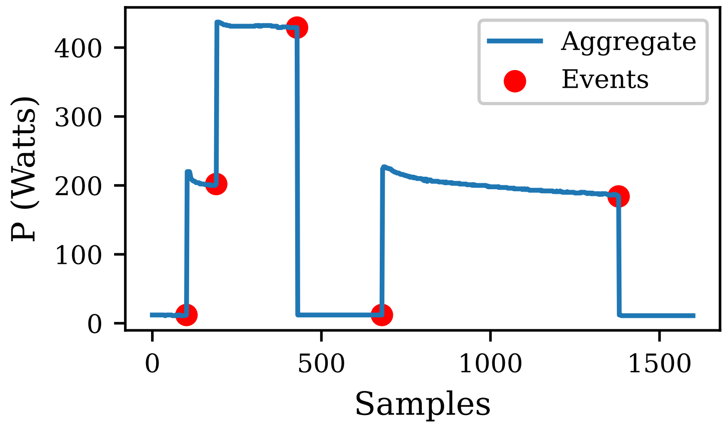


Figure 5.6: Figure showing the obtained result using our event-detection method. Each red marker represents a detected event.

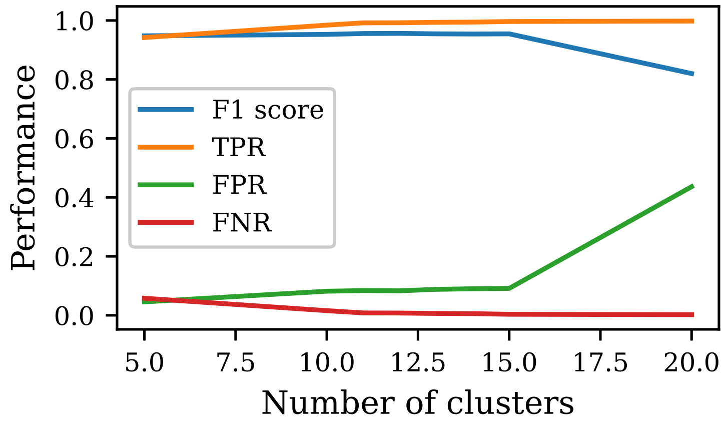


Figure 5.7: Performance results of our method using the Kmeans algorithm.

(TPR) and the decrease of the false-negative rate (FNR). However, the number of noisy events also increases as indicated by the rise of the false-positive rate (FPR). To achieve high discriminate performance between events and non-events., we want to maximize the true-positive rate (TPR) and minimize the false-negative rate (FNR) and false-positive rate (FPR). We obtained the best performance value of $F1 - score = 95.62\%$ using $n_{clusters} = 12$. We obtained similar results to that depicted in Figure 5.7 when we tested our method with the Mini-batch K-means algorithm. However, a lower number of clusters ($n_{clusters} = 7$) was needed to obtain the optimal performance value $F1 - score = 94.95\%$.

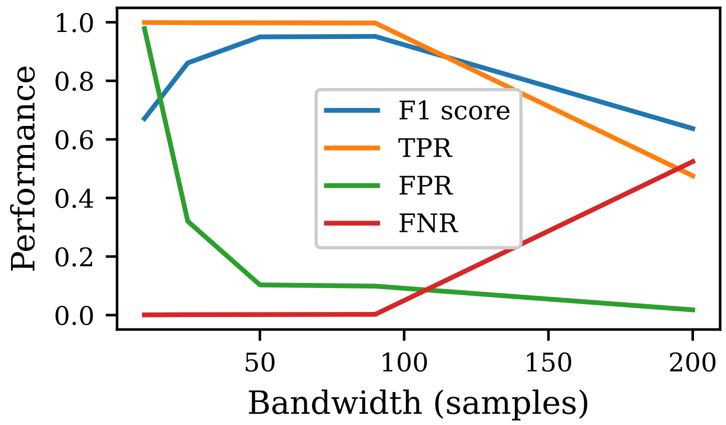


Figure 5.8: Performance results of our method using the Mean Shift algorithm.

Table 5.1: Performance results for our method using K-means, Mean Shift, and Mini-batch K-means clustering algorithms.

Used algorithm	**F1-score (%)**	**Execution time (s)**	**Memory usage (MiB)**
K-means	**95.62**	12.29	297.2
Mean Shift	95.16	213.6	301.2
Mini-batch K-means	94.95	**1.39**	**294.3**

Figure 5.8 shows the performance results we obtained when we used our method with the Mean Shift clustering algorithm. We see that the performance increases for bandwidth values lower than 50 samples. Increasing the bandwidth above 100 samples reduces performance (F1 score). Here the decrease in performance is due to the rise of missed events (FNR) and the diminution of the rate of detected true events (TPR). We noticed that increasing the bandwidth lowers the number of detected clusters which is an expected result. Because when we increase the bandwidth, a higher number of samples of the aggregate signal is required to form a data cluster. When we decrease the bandwidth, a lower number of samples of the aggregate signal is required to form a data cluster, which results in a higher number of data clusters. The best performance value $F1 - score = 95.16\%$ was obtained using a $bandwidth = 90$ samples.

In order to incorporate NILM into smart meters, NILM algorithms should be computationally efficient because smart meters have limited computation resources. For this reason, we computed the time and memory usage of our method when using each clustering algorithm. The obtained results are summarized in Table 5.1. We report our memory usage results in mebibyte (MiB) which is equal to 2^{20} bytes. As shown in Table 5.1 our method consumes approximately the same amount of memory when using each of the three algorithms. However, the execution time of our method varies significantly depending on the clustering algorithm used. The Mini-batch K-means clustering algorithm is the most time-efficient. Our method's execution time using the Mini-batch K-means is approximately 154 times faster than when using the Mean Shift and 9 times faster when compared to the execution time using the K-means. The results presented in this work were obtained using an Intel i7 2.8GHz computer with 8GB RAM of memory.

5.6 Conclusion

In this chapter, we presented our event detection method based on data clustering. We used data clustering to segment the aggregate power signal. In our proposed method, we defined an event as each sample that is followed by another data sample from a different data cluster. We tested our method with three different clustering algorithms, namely, K-means, Mean Shift, and Mini-batch K-means. Our method presented high-performance detection results for all three clustering algorithms. Concerning computation efficiency, all three algorithms used the same amount of memory but, the Mini-batch K-means was considerably more time-efficient. Thus, using our method with the Mini-batch K-means algorithm is preferred if we consider computation costs. Also, our results showed that K-means and Mini-batch K-means were more robust to false negatives (missed event) than the Mean Shift which in contrast, showed better robustness to false positives (false detections) when compared to the K-means and Mini-batch K-means. This last result motivated us to combine, in the future, different clustering algorithms to further improve the performance of our method.

CHAPTER 6

NEAR REAL-TIME LOW-FREQUENCY LOAD DISAGGREGATION

This chapter is based on a journal article accepted for publication, co-authored by Mourad Haddadi [Sahrane et al., ENP Engineering Science Journal, 2021]

Contents

6.1 Introduction 75

6.2 Proposed method 77

6.2.1 Multi-label Classification 77

6.2.2 Multi-output Regression 78

6.2.3 Data 79

6.3 Results and discussion 79

6.4 Conclusion 83

6.1 Introduction

Most of the existing disaggregation approaches are offline methods [94], meaning that they use the entire dataset or day measurements before inferring the consumption of each appliance. This translates into a very low frequency of feedback that does not allow the consumer to take actions in real or near-real time. Real-time or near real-time disaggregation information is needed for the consumer in order to reduce his consumption for more than 9.2% [12] [13]. Zeifman [95] proposed six requirements for a load disaggregation system to be practical with the existing smart meter technology:

1. A sampling rate of 1 Hz: most smart meters use a 1 Hz sampling rate. The sampling frequency affects the feature extraction process and hence the NILM should be designed to work with 1 Hz data.

2. Accuracy: for an acceptable user experience the system should have a minimum accuracy of 80-90%.

3. Easy configuration: minimum training or no training (i.e., unsupervised) and capability to adapt to new appliances and discard old ones.

4. Near real-time feedback: the system is able to give feedback on the energy use of each appliance in a minimum time interval.

5. Robustness: the ability to detect a large number of appliances (e.g., more than 20 devices).

6. Multi-type appliance recognition: some types of appliances are trickier to detect than others, light dimmers which do not have a finite sate of consumption are more difficult to identify than multi-sate appliances like dishwashers. A practical NILM should be able to detect all types of appliances.

These requirements are extensively used among the NILM community and are used as a reference to evaluate load disaggregation methods and there is still no complete solution that satisfies all the six requirements. The fourth requirement (i.e., near real-time capability) is not largely addressed in the NILM literature and this is what motivated us to work on this issue. Another parameter that is not taken into account in the existing solutions is the deployment cost. To deploy NILM on smart meters, the algorithms should require as little memory and computation resources as possible.

In [96] an unsupervised near real-time solution is proposed. This solution is based on the use of low-frequency features (i.e., reactive and active power) as well as high-frequency features (i.e., transients). A clustering algorithm and a manual labeling procedure are used to construct an appliance signature database. The advantage of this solution is that it is unsupervised. However, some features like transients cannot be obtained with existing smart meters.

In [97] a practical implementation of a spectral decomposition-based real-time NILM solution is proposed. The authors use active power and voltage measurements obtained at a frequency of 1 Hz. This method shows good results but has a high implementation cost due to the complexity of the used method.

In [98], the author describes a NILM system able to perform disaggregation on a low-cost embedded processor in real-time using low-frequency sampling data. The method uses a super-state hidden Markov model and a Viterbi algorithm variant which preserves dependencies between loads. This approach is not scalable to a large number of appliances.

In [99], a particle-based distribution truncation method is proposed. This solution uses 1 Hz measurements and has the ability to run in real-time. This approach presents good performance but has a high implementation cost. In fact, the authors implemented their solution on an Intel Core i7-2600 with 8GB of random access memory.

In [8], a method based on particle filtering is proposed. This method uses 1 Hz measurements and is capable of running in real-time. For the implementation, it is reported that the algorithm can work in real-world applications on low-cost hardware such as a Raspberry Pi.

The remaining of this chapter is organized as follows: in section 6.2, the proposed method is described in detail. In section 6.3, results are presented and a discussion is

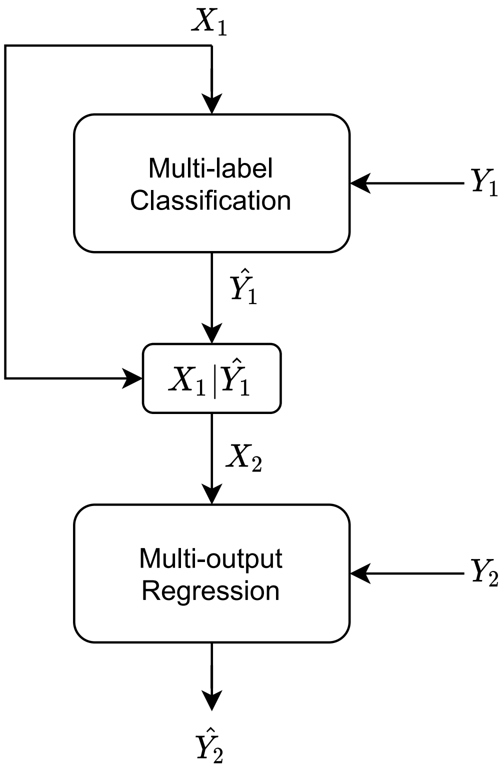


Figure 6.1: Block diagram of the proposed method.

made. Finally, a conclusion is given in section 6.4.

6.2 Proposed method

Our disaggregation method combines a multi-label classification algorithm with a multi-output regression algorithm as shown in Figure 6.1. The classification step serves to predict the state of each load (ON/OFF), and the regression method returns the power consumption of each load in a near real-time fashion. More specifically, for each single aggregate power measurement, our method predicts the corresponding disaggregated power values for each load.

6.2.1 Multi-label Classification

We choose a multi-label classification approach because it is more appropriate for the load disaggregation problem. In general, multiple loads can be operating concurrently in a household, which makes their identification challenging. In a classification context, each load is represented by a unique class/label. A multi-label classification approach allows the association of multiple labels to one data instance thus, permitting to account for cases where more than one load is operating. Formally, given a set of labels Y, each data instance x is associated with a subset $l \subset L$, with L the power set of Y. Two types of multi-label classification methods exist, namely, problem transformation methods and algorithm adaptation methods [100]. Problem transformation methods transform the multi-label classification problem into multiple binary classification problems. Algorithm adaptations modify an existing multi-class algorithm to support multi-label classification. In this work, we use a random forest classifier algorithm implementation adapted to support multi-label classification. The random forest algorithm [75] is an ensemble method that grows multiple decision trees on various sub-samples of the dataset and then averages the predictions to improve the predictive accuracy and control over-fitting. We use the multi-label classification to map each aggregate power sample $x_{1i} \in X_1$ to a label subset l, with X_1 representing a vector of N active power measurements. The ground truth label subset corresponding to each x_{1i} is found in the labels matrix $Y_1 = [y_{1i}, ..., y_{1N}]^T$ with y_{1i} an M-binary vector containing the ON/OFF state of the M loads. $\hat{Y}_1$ represents the predicted states of the M loads given X_1 as input to the classifier, as shown in Figure 6.1.

6.2.2 Multi-output Regression

The goal of multi-output regression is to predict more than two numerical values given an input instance. As for multi-label classification, we find problem transformation methods and algorithm adaptation methods for solving the multi-output problem. An in-depth review of multi-output regression approaches is found in [101]. In this work, we use a problem transformation approach that consists of performing a separate regression for each target. Treating each target load independently is possible because the power consumptions of each load are mutually independent. The feature matrix X_2 is built using the predictions $\hat{Y}_1$ and the aggregate power values X_1 as shown in Figure 6.1. The ground truth power trace of each load $P_j = [p_1, ..., p_N]^T$ is contained in the matrix $Y_2 = [P_1, ..., P_M]$. To find the coefficients of our model, we use Ridge regression [102]. Unlike linear regression, which estimates the model's coefficients by minimizing the residual sum of squares between the observed targets in the data and the targets predicted by linear approximation, Ridge regression minimizes a penalized residual sum of squares. We choose to use Ridge regression because, as mentioned in [102], when using multiple independent variables, and if these variables are not perfectly uncorrelated, the residual sum of squares method has a high probability of giving unsatisfactory results. In our case, the aggregate power samples X_1 and the predicted states of each load $\hat{Y}_1$ are more or less correlated depending on the average consumption of each load. Because switching a load ON/OFF translates into a high/low state which increases/decreases the aggregate power.

6.2.3 Data

We used The REDD dataset [3]. We considered Household 1 which contains active power of ground truth and aggregate data measured over a period of 8 days. We used 80% of the signal for training and 20% for testing. To compare our results with [8], we targeted the same appliances which are, fridge, oven, washing dryer, dishwasher, kitchen outlet, and microwave.

6.3 Results and discussion

The field of NILM lacks standard (or commonly adopted) metrics for the evaluation of the algorithms, making fair comparison difficult [15]. To evaluate the results of our approach, we use the F1-score (6.2) and the relative energy error (6.7). To compare our results with [8], we use the accuracy Acc (6.1) and the normalized mean square error $NRMSE$ (6.5).

$$Acc = \frac{TP + TN}{TP + FN + TN + FP} \tag{6.1}$$

$$\text{F1-score} = \frac{2 \times Pr \times R}{Pr + R} \tag{6.2}$$

$$Pr = \frac{TP}{TP + FP} \tag{6.3}$$

$$R = \frac{TP}{TP + FN} \tag{6.4}$$

The true positive parameter TP represents the number of samples that have been correctly classified or, more precisely, the power quantity correctly assigned to that device. The false-positive parameter FP represents the number of samples that have been incorrectly classified or, more precisely, the power quantity incorrectly assigned to that device. The false-negative parameter FN represents the number of samples that should be but have not been classified or, more precisely, the power quantity that should have been assigned to that device but has been assigned to another or has not been assigned at all. The precision parameter (Pr) measures the portion of power samples that have been correctly classified among the power samples assigned to a given device. The recall parameter (R) measures what power portion of a given device is correctly classified in general, also considering the samples that would belong to that device but have been wrongly assigned to another or not assigned at all. Therefore, the accuracy Acc measures how well each appliance is detected and the F1-score combines the results obtained through the precision and recall analysis.

$$NRMSE = \frac{RMSE}{\overline{X}_1} \tag{6.5}$$

$$RMSE = \sqrt{\frac{\sum\limits_{i=1}^{N} (\hat{x}_{1i} - x_{1i})^2}{N}} \tag{6.6}$$

$$\text{Energy-Error}_j = \frac{\mid \hat{E}_j - E_j \mid}{E_j} \tag{6.7}$$

With $\hat{E}_j$, the estimated energy consumption for the j^{th} load and E_j is the actual energy consumed by the load.

Table. 6.1 shows the evaluation results of our method. We obtain the best classification performance for the fridge with an F1-score = 96.95%. This is because the

Table 6.1: Performance results for each load.

Load	**F1-score (%)**	**Energy error (%)**
Fridge	96.95	3.93
Oven	84.71	0.17
Dishwasher	77.38	3.24
Kitchen Outlet	85.62	0.39
Washing Dryer	78.34	1.26
Microwave	89.87	0.33

Table 6.2: Overall performance results of the proposed NILM method.

Macro-F1 (%)	**Micro-F1 (%)**	**Average Error (%)**
85.48	91.88	1.55

Table 6.3: Performance comparison between our method and the PALDi method [8].

Load	**Accuracy (%)**	
	Our method	PALDi
Fridge	98.4	78.86
Oven	99.91	99.09
Dishwasher	98.57	77.12
Kitchen Outlet	96.72	98.32
Washing Dryer	99.63	99.53
Microwave	99.76	88.33
Total	98.83	90.21
NRMSE	0.65	2.96

refrigerator has a less complexe load signature in comparison to other loads. Also, the refrigerator has the highest number of working cycles, thus, allowing the classifier to learn to detect it in different loads combinations scenarios. The worst classification performance is obtained for the dishwasher with an F1-score = 77.38%. We found that the "wash and drain" cycle of the dishwasher consumes almost the same power as the refrigerator and, because we only use the power as a feature, the classifier can't discriminate between them. In this case, the classifier will often predict the refrigerator as it is the most populated class compared to the dishwasher. The proposed near real-time method can detect loads that work simultaneously as shown in Figure 6.2. We found in our data thirty-nine different load combinations and up to four loads working simultaneously.

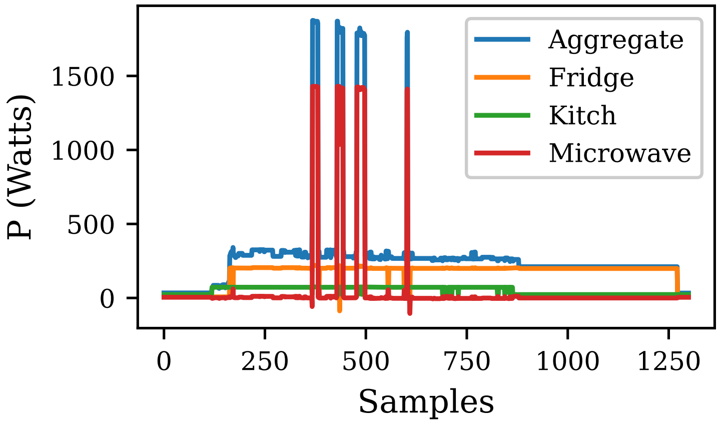


Figure 6.2: Figure showing the disaggregation result obtained using our method. The power signals of the fridge, kitchen outlet, and microwave are disaggregated from the aggregate power signal.

Concerning the energy estimation performance, we found that, in our case, good classification performance doesn't always result in good power/energy estimation performance. For instance, the refrigerator which is the most accurately classified load has the highest energy estimation error as shown in Table 6.1. High power spikes which occur when the refrigerator's compressor starts working can attain up to ten times its average power consumption. These values are difficult to predict because they don't have consistent measurement values in the dataset due to the low temporal resolution of the data. Figure 6.3 shows a bar plot of the estimated energy and the actual consumed energy. Table 6.2 gives the overall performance of the proposed method. Table 6.3 shows the comparison of the results of our method with the PALDi method [8]. We observe that the accuracy is higher for all loads except for the kitchen outlet. We also obtained a lower energy estimation error as measured with the NRMSE.

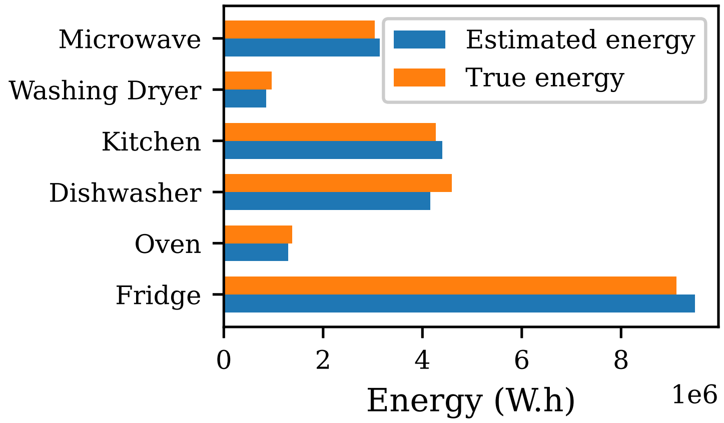


Figure 6.3: Comparison between the estimated energy by our method for each load, and the corresponding actual energy consumption.

6.4 Conclusion

In this chapter, we presented a near real-time load disaggregation method based on multi-label classification and multi-output regression. We used a multi-label classifier to predict the ON/OFF state of each load from the aggregate active power signal and a multi-output regression to estimate the power consumption of each load. The obtained results showed that our method disaggregates loads' energy consumption with low relative energy error. Using only the active power as a feature doesn't allow to differentiate between loads that consume the same power. A compromise exists between NILM feedback frequency and disaggregation performance. Increasing NILM feedback frequency translates into decreasing the amount of available data for NILM prediction, thus, reducing the discriminative capability of extracted features. Also, using high-frequency data may be more adapted for the near real-time NILM problem but at the expense of higher implementation costs.In the future, we will work on the hardware implementation of our method and test it on several households.

CHAPTER 7

EFFECT OF NON-TARGETED LOADS ON NILM PERFORMANCE

This chapter is based on a journal article co-authored by Mourad Adnane and Mourad Haddadi [Sahrane et al., Electric Power Systems Research, 2021 [71]]

Contents

7.1 Introduction **86**

7.2 Background **86**

7.3 Problem formulation **88**

7.4 Methods **89**

7.4.1 Computation of Pearson’s correlation 91

7.4.2 Statistical significance test 91

7.5 Results and discussion **93**

7.6 Conclusion **98**

7.1 Introduction

The devices we do not target for the disaggregation process should be considered and reported. Those devices still exist in real-world scenarios, but researchers view them as noise. Actually, in the field of Non-Intrusive Load Monitoring, the term noise" refers to all the events in the aggregate signal that do not originate from the targeted electrical loads. It includes missing readings, measurement errors, Gaussian noise generated from the sensor, and the non-targeted loads. In [37], the authors defined the noise as being the amount of power remaining in the observed aggregate power reading once we subtract the disaggregated appliance power readings (in-ground truth). Therefore, most researchers exclude the non-targeted loads from their studies and do not label them in the training phase. Also, authors rarely provide detailed information about the testing protocol they use. It makes it difficult to assess if the obtained results still apply in real-world scenarios. Method testing on denoised data (excluding non-targeted loads) produces better results that can be misleading as this testing protocol doesn't reflect a real-world scenario [103], [37]. In [103], the authors designed an experiment and tested different state-of-the-art algorithms on denoised data (excluding non-targeted loads) and data containing noise (data that includes non-targeted loads). The results showed superior disaggregation performance on the denoised aggregate data for all the tested methods. Unfortunately, the literature does not fully address the problem of non-targeted loads. In addition, no existing work models the unknown consumption that results from unkown electrical loads [49]. This fact motivated this work. Our goal here is to explore how those unlabeled, non-targeted loads may affect the disaggregation performance of NILM. Besides, we show if one can predict this effect. We summarize those efforts in the following sections.

7.2 Background

The research efforts invested by the research community during these three decades resulted in several improvements. But there are still many challenges that need to be addressed to make NILM more adapted for real-world usage and more robust.

One of such challenges is to improve the disaggregation performance in presence of non-targeted loads. In fact, given the working environment that comprises unknown appliances/loads, the NILM system must be capable of robust performance [104], [39].

As mentioned in [105], state of the art methods are still susceptible to measurement noise and non-targeted loads, and there is an ongoing effort to tackle this problem. In [106], the authors use power consumption information obtained from the user's manual of appliances to build a sparse switching event recovering (SSER) model based on the sparsity of appliances' switching events. A robust version of the method (RSSER) is also proposed and is reported to be more efficient in the presence of noise. The RSSER model is developed using additional constraints and is solved using a parallel local optimization algorithm.

Denoising autoencoders (dAE) are used in the context of NILM [107], [108], [45]. An autoencoder (AE) is a neural network that tries to reproduce the input in its output. AEs first encode the input data to a compact vector representation in the code layer. Then, we can obtain the result (network's output) by decoding it. A denoising autoencoder attempts to reconstruct a clean target from noisy input [109]. dAEs are typically trained by artificially corrupting a signal before it goes into the network's input data and using the clean signal as the network's target [45]. In the context of NILM, the corrupted signal represents the aggregate signal containing noise. Therefore, we can reconstruct the aggregate signal with only the targeted appliances using the dAE.

For instance, in [107], two scenarios are considered, a noised scenario where the aggregated signal comprises measurement noise and the contributions of unknown appliances and a denoised scenario where the aggregated power is the sum of the power profiles of the disaggregated loads. The tests were performed on three datasets, UK-DALE [35], AMPds [34], and REDD [3]. The results show that the generalization property of the dAE allows more robust performance if compared with the Additive Factorial Approximate MAP method (AFAMAP) [39].

In [60], the authors used particle filters (PF). They tested the proposed method on different scenarios where the number of loads varies between 9 and 18. The authors report that disaggregation performance decreases as the number of loads increases. They also found that the more loads running simultaneously, the more complex the disaggregation problem becomes.

7.3 Problem formulation

In the remaining text of the present work, we use the words "target(s)" and "non-target(s)" interchangeably with "targeted load(s)" and "non-targeted load(s)" for clarity and readability concerns. We consider a house h with N targets and M non-targeted loads. Let $S \in \mathrm{R}^{((N+M)*O)}$ be the matrix that contains the power signal of each load present in the house, such as each power signal contains O time samples:

$$S = [s_1, s_2, ..., s_N, s_{N+1}, ..., s_{N+M}]^T \tag{7.1}$$

s_1 to s_N represent power signals of the targeted loads and s_{N+1} to s_{N+M} represent power signals of the non-targeted loads. The aggregate power signal S_{agg} can be expressed as:

$$S_{agg} = S_{targets} + S_{non-targets} \tag{7.2}$$

with:

$$S_{targets} = \sum_{n=1}^{N} s_n \tag{7.3}$$

and:

$$S_{non-targets} = \sum_{n=N+1}^{N+M} s_n \tag{7.4}$$

The goal of a NILM is to estimate each s_n in $S_{targets}$ given the aggregate signal S_{agg}. As we can see in equation 7.2 and Figure 7.1, the aggregate signal S_{agg} is the summation of both the targeted and non-targeted loads' signals $S_{targets}$ and $S_{non-targets}$. Therefore it is crucial to study the effect of non-targeted loads on performance to design robust disaggregation methods.

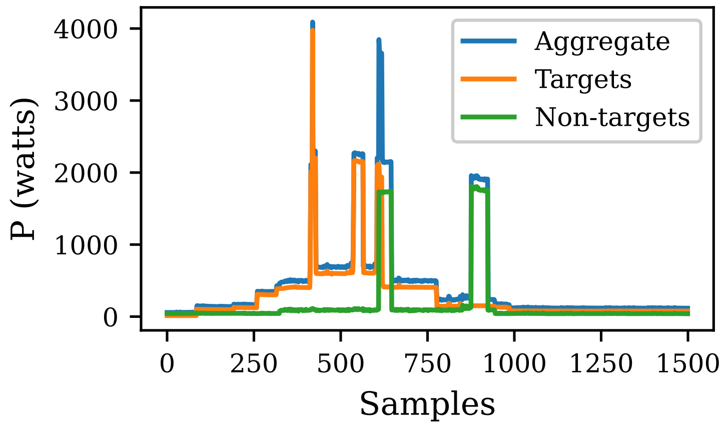


Figure 7.1: Active power of the targets, non-targets, and aggregate signals.

7.4 Methods

To study the effect of non-targeted loads on NILM performance, we design two experiments: experiment 1 and experiment 2, which represent two different scenarios where different load targeting strategies are employed. We represent the targeting process as a partition of the $N + M$ appliances/loads present in house h by modifying the values of N and M . For instance, household 1 has $N1 = 7$ and $M1 = 2$ in experiment 1 (see Table 7.1), and it has $N2 = 5$ and $M2 = 4$ in experiment 2 (see Table 7.2). Figure 7.2 shows a block diagram of the two experiments. Furthermore, we compute the disaggregation performance impact I of non-targeted loads on NILM performance by subtracting the obtained performance value Pr (equation 4.14) including non-targeted loads from the performance value Pr_0 excluding non-targeted loads as shown in equation 7.5. We used our multi-label disaggregation method which we described in chapter 4.

$$I = Pr_0 - Pr \tag{7.5}$$

To explain the impact of non-targeted loads on NILM performance, we consider the probability density functions (pdf) of the data distributions of the targets' signal $f_1(p)$ and the non-targeted loads' signal $f_2(p)$ with p a random variable representing the power consumption. We hypothesize that the more power values shared by the targets and non-targets, the more significant the impact I on disaggregation performance. Mathematically, this translates into obtaining a relation between the impact I and the degree of similarity or overlap of $f_1(p)$ and $f_2(p)$. We use the overlapping coefficient (OVL) [110] to find the overlap between $f_1(p)$ and $f_2(p)$ as shown below:

$$OVL(f_1, f_2) = \int_p min(f_1(p), f_2(p))dp \tag{7.6}$$

The overlapping coefficient (OVL) computes the surface between the x-axis and the minimum (min) between $f_1(p)$ and $f_2(p)$, as shown in figure 7.3. The OVL was previously used in [36] to measure the disaggregation complexity by computing the similarity of the power draws of the different appliances. The model used in this work requires parameters that are difficult to obtain in practice. These are the number of each appliance's consumption states and the distributions (or an approximation) of each aggregated consumption state.

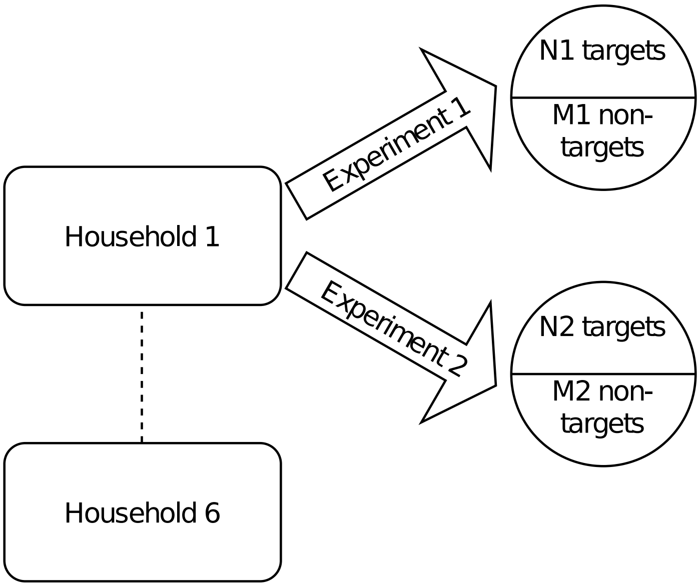


Figure 7.2: Block diagram showing the design of experiment 1 and experiment 2.

To test our hypothesis, we first compute Pearson's correlation coefficient r [111] (equation 7.7) on our sample to find if a linear correlation exists between the amount of overlap we computed with the overlapping index (OVL) and the performance impact I. Then, we evaluate the statistical significance of the correlation r we calculated.

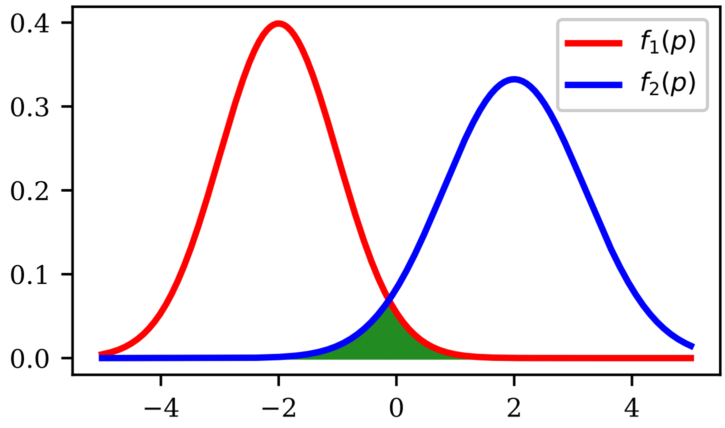


Figure 7.3: In green, the overlap between two probability density functions, $f_1(p)$ and $f_2(p)$.

7.4.1 Computation of Pearson's correlation

The Pearson correlation coefficient r (equation 7.7) measures the strength of linear association between two variables. And it can take values from 0 to ±1, with 0 meaning no correlation and ±1 meaning a perfect positive/negative linear correlation between the two variables.

$$r = \frac{cov(S_{targets}, S_{non-targetedloads})}{s(S_{targets}) \times s(S_{non-targetedloads})} \tag{7.7}$$

with $cov(.)$, being the covariance and $s()$, the sample's standard deviation.

7.4.2 Statistical significance test

The purpose of the statistical significance test is to explore if the correlation r found in our sample can still apply to the population. Figure 7.4 illustrates the steps followed in the statistical significance test. The population, in our case, represents all the households on which we want to apply a NILM method. We first define the *null* hypothesis H_0 and the *alternative* hypothesis H_a, as shown below:

- H_0: there is no significant linear correlation between the OVL and the performance impact in the population.

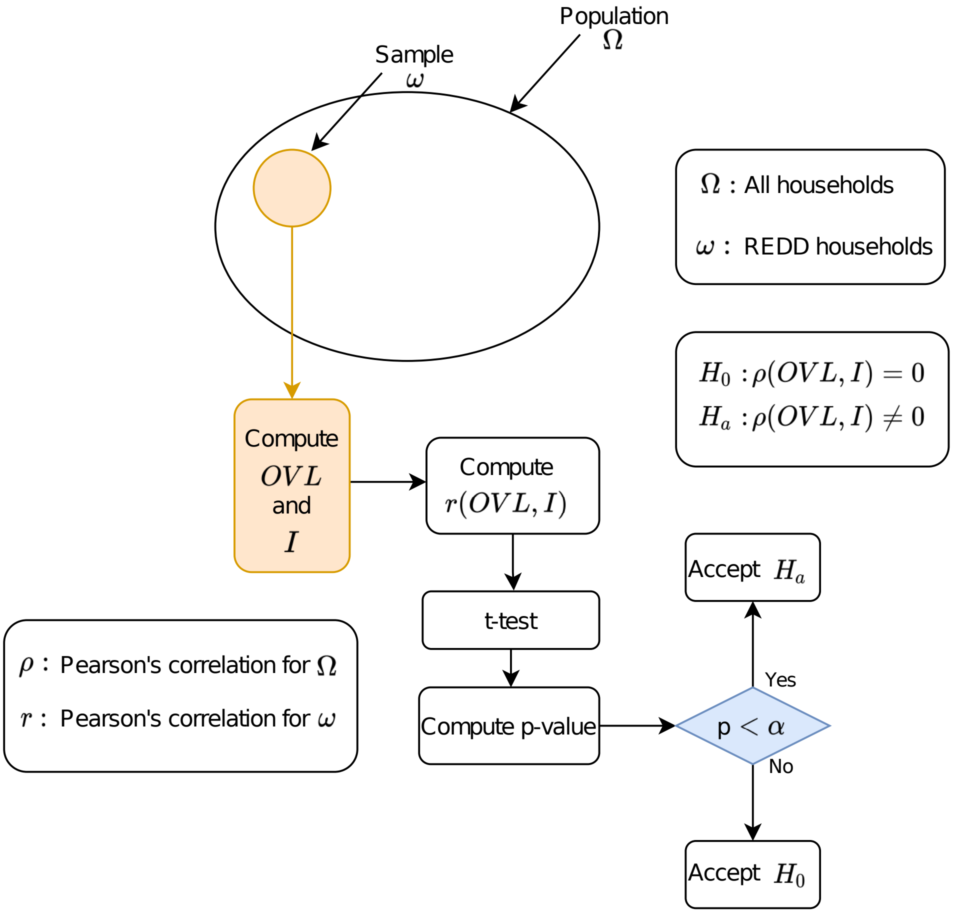

Population
Ω
Sample
ω
Ω : All households
ω : REDD households
$H_0 : \rho(OVL, I) = 0$
$H_a : \rho(OVL, I) \neq 0$
Compute OVL and I
Compute $r(OVL, I)$
t-test
Compute p-value
$p < \alpha$
Yes
No
Accept H_a
Accept H_0
ρ : Pearson's correlation for Ω
r : Pearson's correlation for ω

Figure 7.4: Figure showing the steps followed in the statistical significance test.

- H_a: there is a significant linear correlation between the OVL and the performance impact in the population.

The *alternative* hypothesis H_a is our initial hypothesis previously formulated in this section. We use the Greek letter ρ to differentiate between the *population* correlation coefficient and the *sample* correlation coefficient r. If the test confirms the *null* hypothesis H_0, this would mean that the linear correlation r found in our sample is not present in the entire population ($\rho = 0$). In our context, this would mean that for any given household, the performance impact I is not related to the amount of overlap between the power draws of the targeted and the non-targeted electrical loads computed with the OVL.

We use a *t-test* (equation 7.8) as our test statistic for correlation. The test statistic allows us to quantify the difference between our *null* hypothesis H_0 ($\rho = 0$) and our sample correlation value r to test our assumption of population correlation ($\rho \neq 0$), which corresponds to our hypothesis H_a. We use the *t-test* [112] because we have a small sample size n_s ($n_s < 30$) and because the *t-distribution* is zero-centered ($\mu = 0$), which is more convenient for a correlation significance testing than a *z-distribution*. After quantifying our sample correlation r by computing the *t-test*, we calculate a *p-value*. A *p-value* estimates the likelihood of our assumption of correlation given the *null* hypothesis H_0 ($\rho = 0$). In other words, it estimates the probability of having r given the *null* hypothesis H_0 ($\rho = 0$). We use a significance level α to decide if this probability (*p-value*) is significant enough for rejecting the *null* hypothesis in favor of our hypothesis H_a.

$$t = \frac{r \times \sqrt{n_s - 2}}{\sqrt{1 - r^2}} \tag{7.8}$$

with n_s representing the sample size.

7.5 Results and discussion

To study the effect of non-targeted loads on disaggregation performance, we realized two experiments to assess the impact of the non-targeted loads on the disaggregation performance. For each experiment, we apply a target loads selection strategy. In the first experiment, we maximize the number of target loads to minimize the non-targeted load's number. In the second experiment, we target the same type of loads across all houses. This last approach is often adopted in the literature. The targeted loads are fridge, washing machine, dishwasher, microwave, and lighting. These loads' power measurements are not present in every household dataset file. For example, in household 6, only the fridge's and lighting's power consumptions are measured. To inspect the impact of the non-targeted loads on performance we trained and tested our model without the non-targets. Then, we integrated the non-targets to train and test our disaggregation model. Table 7.1 and Table 7.2 depict the targeted and non-targeted loads for experiment 1 and 2. The results of the two experiments are found respectively in Table 7.3 and Table 7.4. By comparing the average number

Table 7.1: Targets and non-targets for experiment 1.

Households	**Targets**	**Non-targets**
Household 1	’oven’ ’fridge’ ’dish.W’ ’lights’ ’washing.M’ ’mic’ ’stove’	’bath gfi’ ’kitch’
Household 2	’lights’ ’dish.W’ ’stove’ ’mic’ ’fridge’	’disposal’ ’kitch’
Household 3	’lighting’ ’dish.W’ ’washing.M’ ’electronics’ ’fridge’ ’furnace’ ’mic’	’unknown outlets’ ’bathroom gfi’ ’disposal’ ’kitch’ ’smoke alarms’
Household 4	’lighting’ ’dish.W’ ’washing.M’ ’furnace’ ’stove’	’bathroom gfi’ ’kitch’ ’miscellaneous’ ’smoke alarms’ ’unknown outlets’
Household 5	’lighting’ ’dish.w’ ’mic’ ’furnace’ ’electric heat’ ’fridge’ ’electronics’	’bathroom gfi’ ’disposal’ ’kitch’ ’unknown outlets’
Household 6	’stove’ ’electronics’ ’ fridge’ ’electric heat’ ’lighting’ ’air cond’	’bathroom gfi’ ’kitch’ ’unknown outlets’

of non-targeted loads for each experiment, we observe that applying the second target selection criteria increases the average number of non-targets, unlike the first criteria. The average number of non-targets increases from 3.5 loads using the first criteria to 5.83 using the second criteria.

We can find the impact of non-targeted loads on performance I (equation 7.5) by computing the difference between the performance with and without non-targeted loads. In the first experiment, there are, on average, 6.16 targets and 3.5 non-targets loads and, the average performance’s impact is 3.78%. In the second experiment, there are, on average, 3.83 targets and 5.83 non-targets, and the average performance impact is 12.36%.

These results show that, in our case, reducing the average number of non-targeted loads (by targeting more loads) reduces the average performance impact. However, we found that the number of non-targeted loads is not always a good predictor of the performance impact. Other parameters like the complexity of the non-targeted loads’ consumption patterns can also influence the disaggregation performance impact. In fact, for a given house (say house N° 5), the obtained results showed that the more

Table 7.2: Targets and non-targets for experiment 2.

Households	**Targets**	**Non-targets**
Household 1	′fridge′ ′dish.W′ ′lights′ ′washing.M′ ′mic′	′bath gfi′ ′kitch′ ′ stove′ ′oven
Household 2	′lights′ ′dish.W′ ′mic′ ′fridge′	′disposal′ ′kitch′ ′stove′
Household 3	′lighting′ ′dish.W′ ′washing.M′ ′fridge′ ′mic′	′electronics′ ′furnace ′ ′smoke alarms′ ′bathroom gfi′ ′disposal′ ′kitch′ ′unknown outlets′
Household 4	′lighting′ ′dish.W′ ′washing.M′	′bathroom gfi′ ′kitch′ ′miscellaneous′ ′smoke alarms′ ′unknown outlets′ ′furnace′ ′stove′
Household 5	′lighting′ ′dish.w′ ′mic′ ′fridge′	′bathroom gfi′ ′disposal′ ′kitch′ ′uknwn outlets′ ′furnace′ ′electric heat′ ′electronics′
Household 6	′fridge′ ′lighting′	′bathroom gfi′ ′kitch′ ′electric heat′ ′stove′ ′electronics′ ′unknown outlets′ ′air cond′

loads targeted during the training process, the lesser the impact of non-targets on the testing performance of the NILM solution. For instance, targeting all the loads inside house N° 5 will give better results than when we do not target the Electric Heating. Similarly, targeting all loads minus Electric Heating would provide better results than targeting all loads minus Electric Heating and Furnace (Electric Heating and Furnace are non-targets in this case). When comparing different houses, this conclusion is not valid. It depends on the non-targeted loads and the targeted loads too. It depends on the amount of overlap between the distributions of power samples for each of the non-targets signals and the target's signal. It is why the number of non-targeted loads in a given house and the number of non-targeted loads in another house are not sufficient to know whether the impact will be more significant in any of the two habitations. Then, the number of non-targeted loads/appliances is not always a good predictor of

Table 7.3: Performance results for experiment 1.

Households	**Performance (%)**		**N° targets**	**N° non-targets**
	With non-targets	Without non-targets	**(N)**	**(M)**
Household 1	78.53	81.04	7	2
Household 2	71.27	77.89	5	2
Household 3	70.85	72.32	7	5
Household 4	65.53	68.66	5	5
Household 5	59.68	64.35	7	4
Household 6	79.43	83.71	6	3
Averages	**70.88**	**74.66**	**6.16**	**3.5**

the effect of non-targets on disaggregation performance. The quality of the loads and, more precisely, the amount of overlap between the distributions of power samples for each of the non-targets signal and the target's signal is a better predictor. Table 7.5 shows how each non-targeted load impacts the NILM performance in household 5. We obtained these results using a leave-one-out strategy which consists of removing and then restoring each non-targeted load to find its performance impact (I). As we can see, the non-targeted loads produce different performance impact values. It is thus inaccurate to infer the performance impact based only on the number of non-targeted loads.

To test our hypothesis (performance impact of non-targets depends on the amount of overlap between the distributions of power samples for each of the non-targets signal and the targets signal), we computed the overlapping coefficient (OVL) (equation 7.6) and the performance impact I (equation 7.5) to obtain our sample. As explained in section 7.4, our experiments produced two results for each of the six households. Therefore, the obtained sample size is $n_s = 12$. We then computed Pearson's correlation coefficient to check if a linear correlation exists between the overlapping index (OVL) and the performance impact I. We found a moderate positive correlation with the value $r = .676$. Figure 7.5 shows a scatter plot of the standardized values of our sample and a regression line that approximates the linear relation between the

Table 7.4: Performance results for experiment 2.

Households	**Performance (%)**		**N° targets**	**N° non-targets**
	With non-targets	without non-targets	**(N)**	**(M)**
Household 1	83.18	84.09	5	4
Household 2	78.77	83.94	4	3
Household 3	66.63	73.41	5	7
Household 4	64.63	89.94	3	7
Household 5	49.96	77.98	4	7
Household 6	91.12	99.08	2	7
Averages	**72.38**	**84.74**	**3.83**	**5.83**

Table 7.5: Table showing how each non-targeted load impacts the NILM performance in household 5.

Omitted load	**Impact (I) (%)**
Bathroom gfi	0.96
Disposal	0
Kitchen outlets	4.01
Unknown outlets	1.55
Furnace	5.01
Electric heating	5.24
Electronics	4.95

overlapping index (OVL) and the performance impact I.

In addition, a statistical significance test was conducted to determine if the obtained correlation r, using our sample of size $n_s = 12$ can still apply for any given household. We found a significant ($p < .05$) correlation with a p-value $p = .01583$. Table 7.6 depicts the values used for the statistical significance test.

Our method presents similar average performance results for the two experiments in the presence of non-targeted loads (70.88% for experiment 1 and 72.38% for experiment 2). One possible explanation is that the complexity of the loads' space is intrinsic to the households. It is therefore not affected by the targeting strategy. In

Table 7.6: Values used for the statistical significance test.

Parameters	n_s	r	t	α	p
Values	12	.676	2.9	.05	.01583

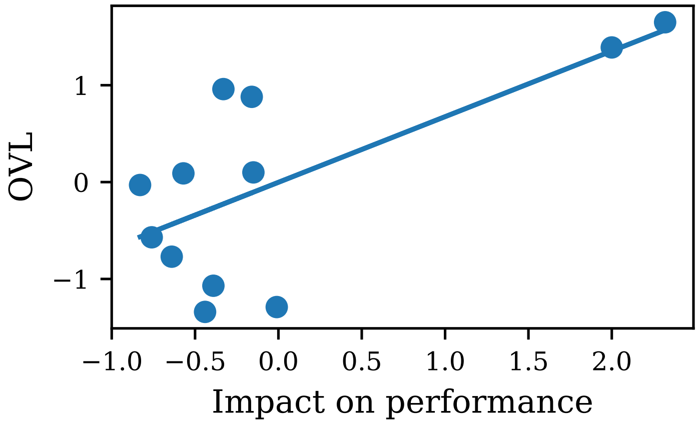


Figure 7.5: Scatter plot of the values (after standardization) used in the hypothesis test. The x-axis represents the performance impact I and the y-axis represents the overlapping index (OVL) computed for the non-targets' signal and the targets' signal. The blue line represents the fitted regression line.

this case, the performance would depend on the NILM method and the complexity of loads' space.

7.6 Conclusion

In this chapter we studied the effect of non-targeted loads on NILM performance using a statistical hypothesis-testing approach. Our results showed that targeting more loads in a given household can be an effective strategy to reduce the effect that non-targeted loads cause on the NILM performance. However, the number of non-targeted loads is not sufficient to explain how they affect the NILM performance. Besides, it doesn't permit to predict this effect in different scenarios (houses). We instead found that the overlap between the two respective distributions of the targets and non-targets was a better predictor of the effect of non-targets on NILM performance. We demonstrated this using a statistical significance test of the correlation we found between the overlapping index (OVL) and the performance impact (I). In the future, we will try to develop a non-targets-aware NILM method capable of adapting to different scenarios with minimal human intervention.

GENERAL CONCLUSION

The principal purpose of this thesis is to study and simulate the different steps of a Non-Intrusive Load Monitoring system using real-world power measurements. The single-label classification approach was our first attempt to model a NILM problem. The model was built with the assumption that often time, targeted loads are not used at the same time. Even though the method showed good results, it presents some limitations for more complex cases with high overlap rate between loads. The multi-label classification approach comes to tackle the limitation of the single label-classification by allowing the identification of multiple loads at a time. The proposed method was event-based and relied on a simple event detector to detect state power changes in the aggregate signal. The method showed good results when combined with a binning method as preprocessing step. However, this approach is prone to noise and suffers to detect events that are not sharp (gradual increase or decrease). We therefore wanted to experiment with another event-detection approach. The clustering-based event detection method we proposed tackles the aforementioned problems. Research showed that increasing the feedback frequency could result in additional savings. This motivated us to find how we could model a near-real time load disaggregation system capable of giving energy consumption feedback each 3 seconds. Our near real-time NILM approach combines a multi-label classification and a multi-output regression to predict the power usage of each targeted load in a near real-time fashion. To successfully deploy NILM solutions in a real-word scenario, many challenges are still to be overcome. One of these challenges is the robustness towards non-targeted loads. This motivated us to study the effect of non-targeted loads on the disaggregation performance to better understand how non-targets impacts the disaggregation performance and also how to make NILM methods more adapted for this problem. These works resulted in the following contributions:

1. Study of the effect of non-targeted loads on the NILM performance.
2. A new event detection approach based on data clustering.
3. A new data preprocessing method using discretization.
4. A new load disaggregation approach based on multi-label classification and multi-output regression.

Future Works

In the future, we plan to undertake the following works:

- Apply different deep learning architectures for NILM
- Design a measurement system and user interface for collecting ground-truth and aggregate signals.
- Collect data and construct a NILM dataset of Algerian households.
- Use the dataset to develop a NILM system.
- Hardware implementation and testing of the developed NILM system in a household.

BIBLIOGRAPHY

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

[2] L. Mauch, K. S. Barsim, and B. Yang, "How well can hmm model load signals," in *Proceeding of the 3rd international workshop on non-intrusive load monitoring (NILM 2016)*, no. 6, 2016.

[3] 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, pp. 59–62, 2011.

[4] C. Beckel, W. Kleiminger, R. Cicchetti, T. Staake, and S. Santini, "The eco data set and the performance of non-intrusive load monitoring algorithms," in *Proceedings of the 1st ACM conference on embedded systems for energy-efficient buildings*, pp. 80–89, 2014.

[5] T. Zia, D. Bruckner, and A. Zaidi, "A hidden markov model based procedure for identifying household electric loads," in *IECON 2011-37th Annual Conference of the IEEE Industrial Electronics Society*, pp. 3218–3223, IEEE, 2011.

[6] 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.

[7] V. Singhal, J. Maggu, and A. Majumdar, "Simultaneous detection of multiple appliances from smart-meter measurements via multi-label consistent deep dictionary learning and deep transform learning," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2969–2978, 2018.

[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] U. Berardi, "A cross-country comparison of the building energy consumptions and their trends," *Resources, Conservation and Recycling*, vol. 123, pp. 230–241, 2017.

[10] N. J. Nunes, L. Pereira, F. Quintal, and M. Berges, "Deploying and evaluating the effectiveness of energy eco-feedback through a low-cost nilm solution," in *Proceedings of the 6th International Conference on Persuasive Technology*, pp. 2–5, 2011.

[11] C. Fischer, "Feedback on household electricity consumption: a tool for saving energy?," *Energy efficiency*, vol. 1, no. 1, pp. 79–104, 2008.

[12] 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.

[13] K. Ehrhardt-Martinez, K. A. Donnelly, S. Laitner, *et al.*, "Advanced metering initiatives and residential feedback programs: a meta-review for household electricity-saving opportunities," American Council for an Energy-Efficient Economy Washington, DC, 2010.

[14] S. Ahmadi-Karvigh, B. Becerik-Gerber, and L. Soibelman, "A framework for allocating personalized appliance-level disaggregated electricity consumption to daily activities," *Energy and Buildings*, vol. 111, pp. 337–350, 2016.

[15] C. Nalmpantis and D. Vrakas, "Machine learning approaches for non-intrusive load monitoring: from qualitative to quantitative comparation," *Artificial Intelligence Review*, pp. 1–27, 2018.

[16] S. Darby *et al.*, "The effectiveness of feedback on energy consumption," *A Review for DEFRA of the Literature on Metering, Billing and direct Displays*, vol. 486, no. 2006, p. 26, 2006.

[17] A. Ruano, A. Hernandez, J. Ureña, M. Ruano, and J. Garcia, "Nilm techniques for intelligent home energy management and ambient assisted living: A review," *Energies*, vol. 12, no. 11, p. 2203, 2019.

[18] M. Bergés and Z. Kolter, "Non-intrusive load monitoring: A review of the state of the art," in *Proceedings of the International Workshop on Non-Intrusive Load Monitoring, Pittsburgh, PA, USA*, vol. 7, 2012.

[19] P. Xiao and S. Cheng, "Neural network for nilm based on operational state change classification," *arXiv preprint arXiv:1902.02675*, 2019.

[20] B. Liu, W. Luan, and Y. Yu, "Dynamic time warping based non-intrusive load transient identification," *Applied Energy*, vol. 195, pp. 634–645, 2017.

[21] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Non-intrusive load monitoring using prior models of general appliance types," 2012.

[22] 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, p. e1265, 2018.

[23] M. B. Figueiredo, A. De Almeida, and B. Ribeiro, "An experimental study on electrical signature identification of non-intrusive load monitoring (nilm) systems," in *International Conference on Adaptive and Natural Computing Algorithms*, pp. 31–40, Springer, 2011.

[24] H.-H. Chang, K.-L. Lian, Y.-C. Su, and W.-J. Lee, "Power-spectrum-based wavelet transform for nonintrusive demand monitoring and load identification," *IEEE Transactions on Industry Applications*, vol. 50, no. 3, pp. 2081–2089, 2013.

[25] M. Xia, K. Wang, X. Zhang, Y. Xu, *et al.*, "Non-intrusive load disaggregation based on deep dilated residual network," *Electric Power Systems Research*, vol. 170, pp. 277–285, 2019.

[26] J. Yu, Y. Gao, Y. Wu, D. Jiao, C. Su, and X. Wu, "Non-intrusive load disaggregation by linear classifier group considering multi-feature integration," *Applied Sciences*, vol. 9, no. 17, p. 3558, 2019.

[27] 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*, pp. 747–758, SIAM, 2011.

[28] B. Humala, A. S. U. Nambi, and V. R. Prasad, "Universalnilm: A semi-supervised energy disaggregation framework using general appliance models," in *Proceedings of the Ninth International Conference on Future Energy Systems*, pp. 223–229, 2018.

[29] 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.

[30] I. Abubakar, S. Khalid, M. Mustafa, H. Shareef, and M. Mustapha, "Application of load monitoring in appliances' energy management–a review," *Renewable and Sustainable Energy Reviews*, vol. 67, pp. 235–245, 2017.

[31] F. De la Prieta, Z. Vale, L. Antunes, T. Pinto, A. T. Campbell, V. Julián, A. J. Neves, and M. N. Moreno, *Trends in Cyber-Physical Multi-Agent Systems. The PAAMS Collection-15th International Conference, PAAMS 2017*. Springer, 2018.

[32] J. R. Herrero, A. L. Murciego, A. L. Barriuso, D. H. de La Iglesia, G. V. González, J. M. C. Rodríguez, and R. Carreira, "Non intrusive load monitoring (nilm): A state of the art," in *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pp. 125–138, Springer, 2017.

[33] S. Welikala, C. Dinesh, M. P. B. Ekanayake, R. I. Godaliyadda, and J. Ekanayake, "Incorporating appliance usage patterns for non-intrusive load monitoring and load forecasting," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 448–461, 2017.

[34] S. Makonin, F. Popowich, L. Bartram, B. Gill, and I. V. Bajić, "Ampds: A public dataset for load disaggregation and eco-feedback research," in *2013 IEEE Electrical Power & Energy Conference*, pp. 1–6, IEEE, 2013.

[35] 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. 1, pp. 1–14, 2015.

[36] D. Egarter, M. Pöchacker, and W. Elmenreich, "Complexity of power draws for load disaggregation," *arXiv preprint arXiv:1501.02954*, 2015.

[37] S. Makonin and F. Popowich, "Nonintrusive load monitoring (nilm) performance evaluation," *Energy Efficiency*, vol. 8, no. 4, pp. 809–814, 2015.

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

[39] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation," in *Artificial intelligence and statistics*, pp. 1472–1482, PMLR, 2012.

[40] 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.

[41] J. Kolter, S. Batra, and A. Ng, "Energy disaggregation via discriminative sparse coding," *Advances in neural information processing systems*, vol. 23, pp. 1153–1161, 2010.

[42] E. Elhamifar and S. Sastry, "Energy disaggregation via learning powerlets and sparse coding," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[43] 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, 2017.

[44] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[45] 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*, pp. 55–64, 2015.

[46] W. He and Y. Chai, "An empirical study on energy disaggregation via deep learning," *Advances in Intelligent Systems Research*, vol. 133, pp. 338–342, 2016.

[47] L. De Baets, J. Ruyssinck, C. Develder, T. Dhaene, and D. Deschrijver, "Appliance classification using vi trajectories and convolutional neural networks," *Energy and Buildings*, vol. 158, pp. 32–36, 2018.

[48] L. De Baets, C. Develder, T. Dhaene, and D. Deschrijver, "Detection of unidentified appliances in non-intrusive load monitoring using siamese neural networks," *International Journal of Electrical Power & Energy Systems*, vol. 104, pp. 645–653, 2019.

[49] Y. Jia, N. Batra, H. Wang, and K. Whitehouse, "A tree-structured neural network model for household energy breakdown," in *The World Wide Web Conference*, pp. 2872–2878, 2019.

[50] L. Massidda, M. Marrocu, and S. Manca, "Non-intrusive load disaggregation by convolutional neural network and multilabel classification," *Applied Sciences*, vol. 10, no. 4, p. 1454, 2020.

[51] K. Bao, K. Ibrahimov, M. Wagner, and H. Schmeck, "Enhancing neural non-intrusive load monitoring with generative adversarial networks," *Energy Informatics*, vol. 1, no. 1, p. 18, 2018.

[52] S. Dai, F. Meng, Q. Wang, and X. Chen, "Federatednilm: A distributed and privacy-preserving framework for non-intrusive load monitoring based on federated deep learning," *arXiv preprint arXiv:2108.03591*, 2021.

[53] Q. Li, J. Ye, W. Song, and Z. Tse, "Energy disaggregation with federated and transfer learning,"

[54] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[55] Z. Yue, C. R. Witzig, D. Jorde, and H.-A. Jacobsen, "Bert4nilm: A bidirectional transformer model for non-intrusive load monitoring," in *Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring*, pp. 89–93, 2020.

[56] P. Huber, A. Calatroni, A. Rumsch, and A. Paice, "Review on deep neural networks applied to low-frequency nilm," *Energies*, vol. 14, no. 9, p. 2390, 2021.

[57] 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)*, pp. 81–87, IEEE, 2014.

[58] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE transactions on signal processing*, vol. 61, no. 7, pp. 1644–1656, 2013.

[59] K. Basu, V. Debusschere, S. Bacha, U. Maulik, and S. Bondyopadhyay, "Nonintrusive load monitoring: A temporal multilabel classification approach," *IEEE Transactions on industrial informatics*, vol. 11, no. 1, pp. 262–270, 2014.

[60] 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, 2014.

[61] H. He, Z. Liu, R. Jiao, and G. Yan, "A novel nonintrusive load monitoring approach based on linear-chain conditional random fields," *Energies*, vol. 12, no. 9, p. 1797, 2019.

[62] P. Heracleous, P. Angkititrakul, N. Kitaoka, and K. Takeda, "Unsupervised energy disaggregation using conditional random fields," in *IEEE PES Innovative Smart Grid Technologies, Europe*, pp. 1–5, IEEE, 2014.

[63] V. M. Salerno and G. Rabbeni, "An extreme learning machine approach to effective energy disaggregation," *Electronics*, vol. 7, no. 10, p. 235, 2018.

[64] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2011.

[65] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Physical review E*, vol. 69, no. 6, p. 066138, 2004.

[66] P. Cunningham and S. J. Delany, "k-nearest neighbour classifiers," *Multiple Classifier Systems*, vol. 34, no. 8, pp. 1–17, 2007.

[67] J. Kim, T.-T.-H. Le, and H. Kim, "Nonintrusive load monitoring based on advanced deep learning and novel signature," *Computational intelligence and neuroscience*, vol. 2017, 2017.

[68] Z. Deng, X. Zhu, D. Cheng, M. Zong, and S. Zhang, "Efficient knn classification algorithm for big data," *Neurocomputing*, vol. 195, pp. 143–148, 2016.

[69] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[70] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5353–5360, 2015.

[71] S. Sahrane, M. Adnane, and M. Haddadi, "Multi-label load disaggregation in presence of non-targeted loads," *Electric Power Systems Research*, vol. 199, p. 107435, 2021.

[72] H. Liu, F. Hussain, C. L. Tan, and M. Dash, "Discretization: An enabling technique," *Data mining and knowledge discovery*, vol. 6, no. 4, pp. 393–423, 2002.

[73] K. D. Anderson, M. E. Berges, A. Ocneanu, D. Benitez, and J. M. Moura, "Event detection for non intrusive load monitoring," in *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society*, pp. 3312–3317, IEEE, 2012.

[74] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine learning*, vol. 85, no. 3, p. 333, 2011.

[75] T. K. Ho, "Random decision forests," in *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1, pp. 278–282, IEEE, 1995.

[76] X. Wu, Y. Gao, and D. Jiao, "Multi-label classification based on random forest algorithm for non-intrusive load monitoring system," *Processes*, vol. 7, no. 6, p. 337, 2019.

[77] C. Strobl, J. Malley, and G. Tutz, "An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests.," *Psychological methods*, vol. 14, no. 4, p. 323, 2009.

[78] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[79] R. W. Dias Pedro, A. Machado-Lima, and F. L. Nunes, "Towards an approach using grammars for automatic classification of masses in mammograms," *Computational Intelligence*, 2020.

[80] S. Sahrane, M. Adnane, and M. Haddadi, "A clustering event detection approach for non-intrusive load monitoring," in *2020 6th International Symposium on New and Renewable Energy (SIENR)*, pp. 1–7, IEEE, 2021.

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

[82] D. Luo, L. K. Norford, S. R. Shaw, and S. B. Leeb, "Monitoring hvac equipment electrical loads from a centralized location–methods and field test results/discussion," *ASHRAE Transactions*, vol. 108, p. 841, 2002.

[83] Y. Jin, E. Tebekaemi, M. Berges, and L. Soibelman, "A time-frequency approach for event detection in non-intrusive load monitoring," in *Signal Processing, Sensor Fusion, and Target Recognition XX*, vol. 8050, p. 80501U, International Society for Optics and Photonics, 2011.

[84] S. B. Leeb, S. R. Shaw, and J. L. Kirtley, "Transient event detection in spectral envelope estimates for nonintrusive load monitoring," *IEEE Transactions on Power Delivery*, vol. 10, no. 3, pp. 1200–1210, 1995.

[85] M. N. Meziane, P. Ravier, G. Lamarque, J.-C. Le Bunetel, and Y. Raingeaud, "High accuracy event detection for non-intrusive load monitoring," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2452–2456, IEEE, 2017.

[86] K. S. Barsim and B. Yang, "Sequential clustering-based event detection for non-intrusive load monitoring," *Computer Science & Information Technology*, vol. 6, no. 10.5121, 2016.

[87] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *Kdd*, vol. 96, pp. 226–231, 1996.

[88] A. U. Rehman, S. R. Tito, T. T. Lie, P. Nieuwoudt, N. Pandey, D. Ahmed, and B. Vallès, "Non-intrusive load monitoring: A computationally efficient hybrid event detection algorithm," in *2020 IEEE International Conference on Power and Energy (PECon)*, pp. 304–308, IEEE, 2020.

[89] M. Lu and Z. Li, "A hybrid event detection approach for non-intrusive load monitoring," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 528–540, 2019.

[90] L. Rokach and O. Maimon, "Clustering methods," in *Data mining and knowledge discovery handbook*, pp. 321–352, Springer, 2005.

[91] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[92] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[93] D. Sculley, "Web-scale k-means clustering," in *Proceedings of the 19th international conference on World wide web*, pp. 1177–1178, 2010.

[94] S. Barker, S. Kalra, D. Irwin, and P. Shenoy, "Nilm redux: The case for emphasizing applications over accuracy," in *NILM-2014 workshop*, Citeseer, 2014.

[95] M. Zeifman, "Disaggregation of home energy display data using probabilistic approach," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 1, pp. 23–31, 2012.

[96] T. Bernard and M. Marx, "Unsupervised learning algorithm using multiple electrical low and high frequency features for the task of load disaggregation," in *Proceedings of the 3rd International Workshop on NILM, Vancouver, BC, Canada*, pp. 14–15, 2016.

[97] S. Welikala, N. Thelasingha, M. Akram, P. B. Ekanayake, R. I. Godaliyadda, and J. B. Ekanayake, "Implementation of a robust real-time non-intrusive load monitoring solution," *Applied Energy*, vol. 238, pp. 1519–1529, 2019.

[98] S. W. Makonin, *Real-time embedded low-frequency load disaggregation*. PhD thesis, Applied Sciences: School of Computing Science, 2014.

[99] Y. F. Wong, T. Drummond, and Y. Şekercioğlu, "Real-time load disaggregation algorithm using particle-based distribution truncation with state occupancy model," *Electronics Letters*, vol. 50, no. 9, pp. 697–699, 2014.

[100] B. Buddhahai, W. Wongseree, and P. Rakkwamsuk, "A non-intrusive load monitoring system using multi-label classification approach," *Sustainable cities and society*, vol. 39, pp. 621–630, 2018.

[101] H. Borchani, G. Varando, C. Bielza, and P. Larranaga, "A survey on multi-output regression," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, no. 5, pp. 216–233, 2015.

[102] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[103] R. Kukunuri, N. Batra, A. Pandey, R. Malakar, R. Kumar, O. Krystalakos, M. Zhong, P. Meira, and O. Parson, "Nilmtk-contrib: Towards reproducible state-of-the-art energy disaggregation,"

[104] 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.

[105] 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, 2018.

[106] G. Tang, K. Wu, J. Lei, and J. Tang, "A simple and robust approach to energy disaggregation in the presence of outliers," *Sustainable Computing: Informatics and Systems*, vol. 9, pp. 8–19, 2016.

[107] R. Bonfigli, A. Felicetti, E. Principi, M. Fagiani, S. Squartini, and F. Piazza, "Denoising autoencoders for non-intrusive load monitoring: improvements and comparative evaluation," *Energy and Buildings*, vol. 158, pp. 1461–1474, 2018.

[108] F. C. C. Garcia, C. M. C. Creayla, and E. Q. B. Macabebe, "Development of an intelligent system for smart home energy disaggregation using stacked denoising autoencoders," *Procedia Computer Science*, vol. 105, no. C, pp. 248–255, 2017.

[109] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

[110] H. F. Inman and E. L. Bradley Jr, “The overlapping coefficient as a measure of agreement between probability distributions and point estimation of the overlap of two normal densities,” *Communications in Statistics-Theory and Methods*, vol. 18, no. 10, pp. 3851–3874, 1989.

[111] P. Sedgwick, “Pearson’s correlation coefficient,” *Bmj*, vol. 345, p. e4483, 2012.

[112] D. Kalpić, N. Hlupić, and M. Lovrić, “Student’s t-tests,” *International Encyclopedia of Statistical Science. Part 19/Lovrić, Miodrag (ur.).; Berlin: Springer, 2011.; 1559-1563; DOI: 10.1007/978-3-642-04898-2_641; p-ISBN 978-3-642-04897-5, eISBN 978-3-642-04898-2*, 2011.