UNIVERSIDADE DE CAMPINAS

FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

FERNANDO MARCOS WITTMANN

OPTIMIZATION APPLIED TO RESIDENTIAL NON-INTRUSIVE LOAD MONITORING

OTIMIZAÇÃO APLICADA AO MONITORAMENTO NÃO INTRUSIVO DE CARGAS ELÉTRICAS RESIDENCIAIS

Campinas

2017

FERNANDO MARCOS WITTMANN

OPTIMIZATION APPLIED TO RESIDENTIAL NON-INTRUSIVE LOAD MONITORING

OTIMIZAÇÃO APLICADA AO MONITORAMENTO NÃO INTRUSIVO DE CARGAS ELÉTRICAS RESIDENCIAIS

Dissertation presented to the School of Electrical and Computer Engineering (FEEC) of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Electrical Engineering, in the area of Electrical Energy.

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação (FEEC), da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica, na área de Energia Elétrica.

Orientador: Marcos Julio Rider Flores

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO DEFENDIDA PELO ALUNO FERNANDO MARCOS WITTMANN, E ORIENTADA PELO PROF. DR. MARCOS JULIO RIDER FLORES.

____

CAMPINAS

2017

Agência(s) de fomento e nº(s) de processo(s): FAEPEX, 1594/15

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Luciana Pietrosanto Milla - CRB 8/8129

W786o

Wittmann, Fernando Marcos, 1992-
Optimization applied to residential non-intrusive load monitoring / Fernando Marcos Wittmann. – Campinas, SP : [s.n.], 2017.

Orientador: Marcos Julio Rider Flores.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Programação linear inteira mista. 2. Otimização combinatória. 3. Energia elétrica - Medição. 4. Energia elétrica - Consumo. I. Rider Flores, Marcos Julio, 1975-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Otimização aplicada ao monitoramento não intrusivo de cargas elétricas residenciais
Palavras-chave em inglês:
Mixed integer linear programming
Combinatorial optimization
Electrical energy - Measurement
Electrical energy - Consumption
Área de concentração: Energia Elétrica
Titulação: Mestre em Engenharia Elétrica
Banca examinadora:
Marcos Julio Rider Flores [Orientador]
Luis Carlos Pereira da Silva
Roberto Cayetano Lotero
Data de defesa: 15-09-2017
Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - DISSERTAÇÃO DE MESTRADO

Candidato: Fernando Marcos Wittmann RA: 180577
Data da Defesa: 15 de setembro de 2017

Título da Dissertação: "OTIMIZAÇÃO APLICADA AO MONITORAMENTO NÃO INTRUSIVO DE CARGAS ELÉTRICAS RESIDENCIAIS".

Prof. Dr. Marcos Julio Rider Flores (Presidente, FEEC/UNICAMP)
Prof. Dr. Roberto Cayetano Lotero (CECE/UNIOESTE)
Prof. Dr. Luiz Carlos Pereira da Silva (FEEC/UNICAMP)

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no processo de vida acadêmica do aluno.

À meus pais, Ivo Amandio Wittmann e Maria Salete Wittmann.

Agradecimentos

À FAEPEX pelo apoio financeiro.

Ao meu orientador Marcos Rider pela oportunidade, paciência, apoio e disponibilidade. Sua abordagem como orientador tornou minha experiência de mestrado motivante e agradável.

Aos colegas e amigos do LE19 e LE23, em especial ao Camilo Lopez pelas importantes contribuições e ao Hader pelas dúvidas pontuais esclarecidas.

Aos meus amigos, de dentro e de fora da faculdade, em especial aqueles que conviviam comigo no dia-dia. Dentre eles, cito meus amigos da república Nelson e o Antônio, que me acompanham desde meus primeiros dias em Campinas, também Carla, Carol e Cristiano por tornarem o meu último semestre mais divertido. Também cito Alison, Joel, Jaime, Saullo, Ricardo J. e Ivi.

À Catarina Perez, por oferecer incentivo e encorajamento na reta final de meu mestrado.

Por último mas não menos importante, à minha família, por sempre me oferecer apoio, sustentação e exemplo.

"Study hard what interests you the most in the most undisciplined, irreverent and original manner possible."

Richard Feynman

ABSTRACT

This work presents a non-intrusive load monitoring (NILM) method based on mixed-integer linear programming (MILP). NILM are methods for disaggregating measurements from energy meters into information regarding operating appliances. Such information, such as the power consumption and operating state, are valuable for promoting energy savings and predictive maintenance. The proposed technique expands the classical model based on combinatorial optimization (CO). The new formulation handles the problem of ambiguity of similar loads, present in the classical model. Linear constraints are used to efficiently represent load signatures. Additionally, a window-based strategy is proposed to enhance the computational performance of the proposed NILM algorithm. The disaggregation can be made using only active power measurements at low sampling rate, which is already available in commercial smart meters. Other features can be added to the model, if available, such as the reactive power. The performance of the algorithm is evaluated using two test cases from the public dataset AMPds. The sampling rate from the test case is of one sample per minute. Results demonstrate the ability of the proposed method to accurately identify and disaggregate individual energy signatures in a computationally efficient way.

Keywords: load disaggregation, load signature, mixed-integer linear programming, non-intrusive load monitoring, combinatorial optimization.

RESUMO

Este trabalho apresenta um método de monitoramento não intrusivo (Non-Intrusive Load Monitoring - NILM) baseado em programação linear inteira mista (Mixed-Integer Linear Programming - MILP). NILM são métodos para desagregar leituras de medidores de energia em informações a respeito dos aparelhos eletrodomésticos em operação. Tais informações, como consumo e estado de operação, são valiosas para promover a eficiência energética e manutenção preventiva. A técnica NILM proposta neste trabalho expande o modelo clássico fundamentado em otimização combinatória (Combinatorial Optimization - CO). A nova formulação lida com o problema de ambiguidade de cargas similares, presente no modelo clássico. Restrições lineares são utilizadas para representar eficientemente as assinaturas de carga. Além disso, uma estratégia baseada em janelas temporais é proposta para melhorar o desempenho computacional. A desagregação de cargas pode ser feita utilizando apenas medidas de potência ativa em uma baixa taxa de amostragem, disponível em medidores inteligentes comerciais. A técnica também permite a utilização de outros tipos de medidas, se disponíveis, como a potência reativa. O desempenho do algoritmo é validado utilizando dois casos de teste a partir da base de dados pública AMPds. A taxa de amostragem do caso de teste é de uma amostra por minuto. Os resultados demonstram a habilidade do método proposto para identificar e desagregar com precisão as assinaturas de energia individuais de forma computacionalmente eficiente.

Palavras-chave: desagregação de carga, assinatura de carga, programação linear inteira mista, monitoramento não intrusivo, otimização combinatória.

List of Figures

List of Tables

ACRONYMS

AMPds Almanac of Minutely Power dataset

CO Combinatorial Optimization

EAP Error in Assigned Power

ETE Error in Total Energy

IP Integer Programming

LP Linear Programming

MILP Mixed-Integer Linear Programming

MS Multiple Switching

NILM Non-Intrusive Load Monitoring

NILMTK Non-Intrusive Load Monitoring Toolkit

NLP Non-linear Programming

QP Quadratic Programming

LIST OF SYMBOLS

Sets

D Set of appliances

S Set of indexes associated to power states

T Set of discrete time periods

$\tilde{T}$ Subset of discrete time periods within a window, where $\tilde{T} \subseteq T$

Indexes

i Index of each power state, where $i \in S$

j Index of each appliance, where $j \in D$

t Index of each time period, where $t \in T$

Parameters

D_i Appliance associated with the state i

G_i Number of time periods in which the state i must be activated

m Number of time periods in a window

MD_i Minimum number of time periods in which the state i must be ON

n Total number of power states in S

NP_i Number of time periods in which i has been activated in the previous window

P_i Active power state assigned to the index i

$P(t)$ Active power true measurement at time t

$\text{prev}_{\boldsymbol{i}}$ Previous state of i (0 if none)

Q_i Reactive power state assigned to the index i

$Q(t)$ Reactive power true measurement at time t

T_0 Initial time period of each new window

T_f Last time period of each new window in $\tilde{T}$

$T_{\textbf{end}}$ Last time period of T

X_i Final status of the state i within the current window. $X_i = 1$ if the state i is ON at the end of the window; otherwise, $X_i = 0$.

$y_j(t)$ True power measurement of a given appliance j at time t.

$\hat{y}_j(t)$ Predicted power measurement of a given appliance j at time t.

Variables

$dw_i(t)$ Binary variable that indicates if state $i \in S$ was turned off at time t

$up_i(t)$ Binary variable that indicates if state $i \in S$ was turned on at time t

$x_i(t)$ Binary variable that represents the status of state i at time t

$\delta_P(t)$ Difference between the measured value $P(t)$ and the combination of loads at time t

$\delta_Q(t)$ Difference between the measured value $Q(t)$ and the combination of loads at time t

Summary

Chapter 1

Introduction

1.1 Motivation

The number of smart meters in the world is expected to increase up to 780 Million by 2020 [1]. Nowadays, there are more than 50 Million installed in the US [2]. However, unlike phone bills in which calls are individually identified, the energy bill shows only the total price which is a limited information. Providing energy usage feedback to homeowners is a helpful way for promoting the energy efficiency [3]. Detailed appliance information on energy consumption is still invisible to the general population which implies in a source of waste. Without feedback, it is impossible for people to learn effectively about their energy usage patterns, necessary for energy savings.

As shown in the spectrum of the Figure 1.1, the energy feedback can be either indirect or direct [4]:

- The indirect feedback is provided after the consumption occurs. It may range from a standard electricity bill to daily reports. This is the most common scenario for customers of power utilities nowadays.

- The direct feedback is provided in real time with either aggregated or disaggregated energy information. As a difference, the aggregated information provides the whole-building consumption information while disaggregated information provides appliance level consumption information.

Disaggregated feedback is in the highest informational level of the Figure 1.1. It is a valuable resource for homeowners, commercial buildings, and power utilities. For homeowners

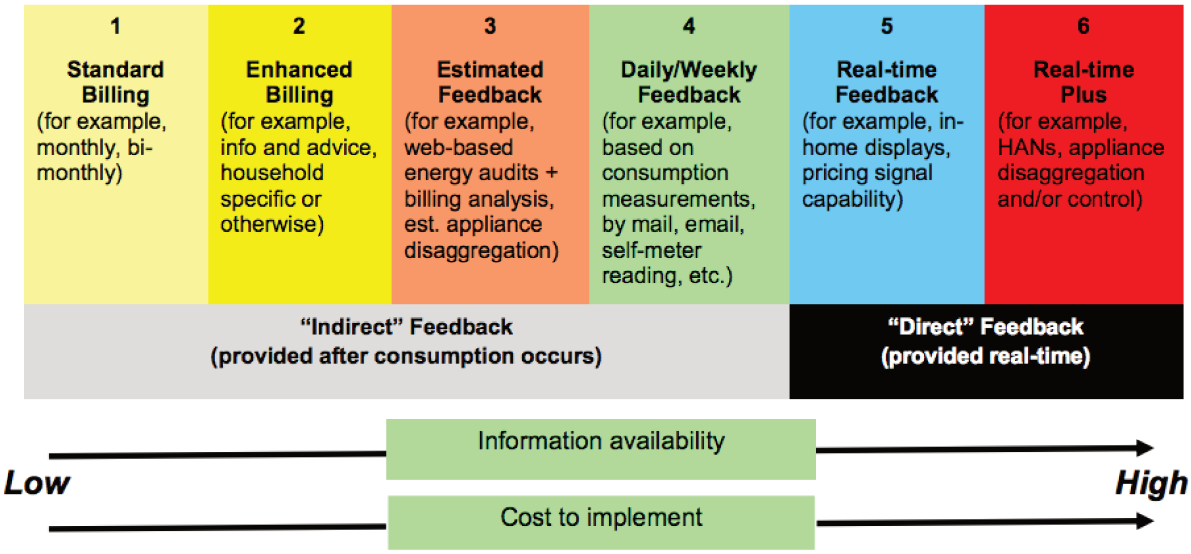


Figure 1.1: Spectrum with the different types of energy feedback [4]

and commercial buildings, disaggregation helps them making decisions about their consumption habits and appliances to save energy [5]. In addition, it can be helpful to detect malfunctions, inefficient equipment or for scheduling predictive maintenance [6]. In regards to power utilities, disaggregation helps them to understand their customers and provide them with better-customized services [7]. However, disaggregated feedback is a very difficult task to be accomplished since it implies into finding a way to monitor all the appliances of a building. In fact, the disaggregation of loads can be accomplished either with intrusion or not intrusively:

- The intrusive approach connects one measurement device to each power outlet. This process might lead to high installation costs since it is necessary an infrastructure with one measurement device for each outlet and also a network infrastructure to connect all of them. This solution might also lead to privacy concerns, for example, if the power utility wants to provide an intrusive approach to their customers, their employees should get inside their customer's houses.

- On the non-intrusive approach the energy consumption of the major appliances is estimated using only a single meter installed in the consumer's energy input panel. The measurements acquired by the energy meter are processed in order to provide details about the operating devices.

The advantage of non-intrusive identification is the reduced costs of hardware, maintenance, and privacy. However, non-intrusive identification is still a software challenge due to the complexity of extracting a set of individual load measurements from the whole house power measurement. This challenge is closely related to the cocktail party problem where there are multiple sound sources recorded by a microphone and we want to extract just one of these sources [8]. In this context, non-intrusive load monitoring (NILM), also known as load disaggregation, is a research field which seeks to break down a whole-house power signal into individual devices.

1.2 Description of the Problem

The keyword NILM was introduced in 1985 by George W. Hart in a technical report [9] and later published in 1992 by the same author [10]. The paper proposes a full methodology, from types of loads, load signatures, algorithm and physical implementation. The algorithm is based on pattern recognition, which is described in the next chapter.

Before introducing the pattern recognition method, Hart describes the NILM problem as a combinatorial optimization (CO) problem. However, the author discourages its usage. In the CO formulation, the disaggregation is obtained by combining the multiple possible states that minimize the full measurement for each time instant. For each time instant, we seek to minimize the error between the combination of power states and the aggregated measurement. Here, it is assumed that all the possible operating states are previously known. More details about this formulation are also presented in the next chapter. Hart points three main issues to discourage the usage of this formulation:

- The problem has a high computational cost and it increases with the addition of more states of devices or measurements.
- **Fundamental Problem**: The complete set of operating states are never known. If the model is used in the presence of unknown appliances, it would attempt to describe their behavior as a combination of other known appliances.
- **Multiple Switching (MS)**: A small change in the measurement might be translated into a big change of the combination of loads.

As shown in the next section, despite various NILM approaches proposed in the literature, very few have attempted in dealing with the CO formulation. While those difficulties are still challenging, there is still room and potential for expanding the formulation in ways to handle the previous problems. Techniques based on mathematical programming are still emerging. This work is especially concerned in formulating the NILM problem as a Mixed-Integer Linear Programming (MILP) problem in order to handle the MS.

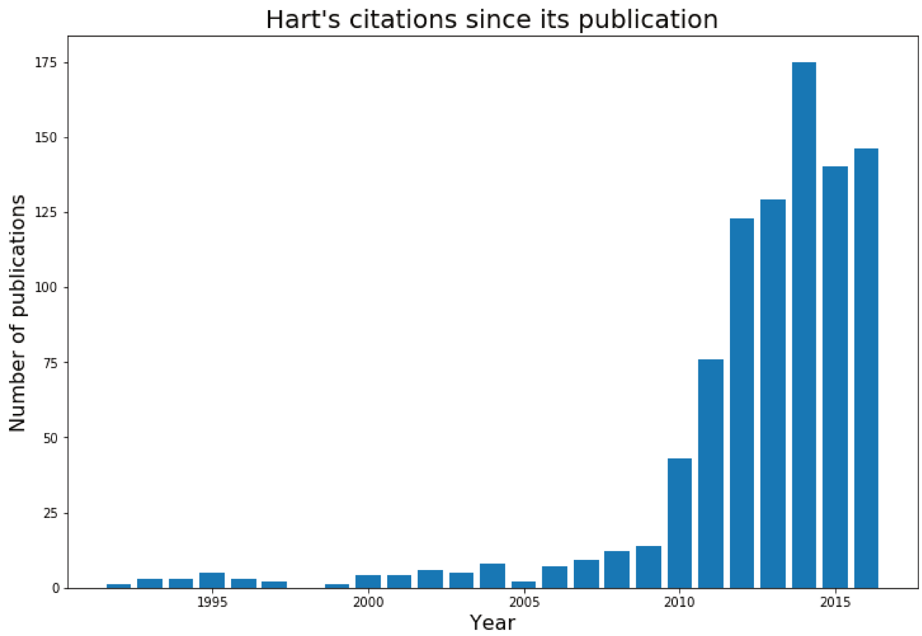


Figure 1.2: Number of publications with reference to Hart's original work since its publication

1.3 Previous Work

Figure 1.2 shows the number of publications in the NILM field per year over the last 25 years. Those NILM publications are inferred using publications citing the first NILM work published by Hart in 1992[1]. An exponential growth of NILM publications can be observed since 2010.

A further analysis of the publication's titles since 2010 did not reveal any trending strategy. Figure 1.3 shows the most frequent words in those titles. Some obvious keywords such as 'nonintrusive', 'load', 'monitoring', 'energy' and 'disaggregation' were removed. As main insights, the NILM research field seems to be mainly focused on residential applications and buildings. Therefore, little attention has been paid to industrial applications. In addition, it seems that privacy is increasingly a concern in the field.

There are many ways of categorizing the NILM approaches. For example, they can be categorized into either supervised or unsupervised approaches [12]:

- Supervised NILM uses measurements of each single appliance to build their models and then disaggregate. It is assumed that we previously have access to the measurements of each individual appliance that is part of the full house measurement.

- Unsupervised NILM does not require single measurements of each appliance for training. This approach allows general appliance models as input which are then tuned to each

[1]Based on [11]. Data acquired from Google Scholar. Code available in https://github.com/WittmannF/nilm-publications

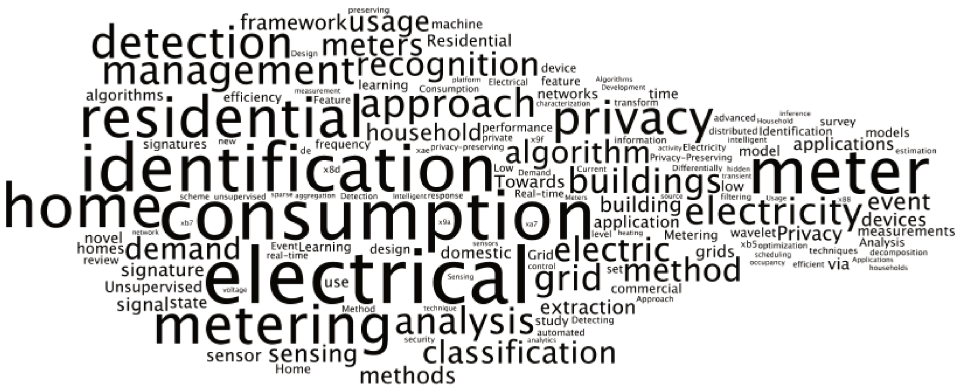


Figure 1.3: Trending words in the titles of NILM publications since 2010

specific house. The challenge here is to find the right parameter's values to be tuned.

Zeifman and Roth in [13] divide the types of algorithms in two categories: pattern recognition (one-to-one matching) and optimization (multiple matching). Zeifman's categorization could also be expanded into either event-based algorithms or eventless algorithms:

- The event-based matching is based on the detection and classification of events. Since the pattern recognition strategies are based on one-to-one matching, the efforts of these strategies are mainly focused on feature extraction. As an example of features to be extracted, we have the active and reactive power, harmonics, wavelet's signatures and fundamental frequency. These events are then classified by well-established classifiers, such as k-nearest neighbor [14, 15, 16], fuzzy sets [17, 18], decision trees [19, 20], support vector machines [21, 22], neural networks [23, 24] and deep learning [25, 26]. As noted by [13], the advantages of pattern recognition methods are that they provide better results in the presence of unknown loads. However, those methods are sensitive to the detection of false edges coming from noise or non-linear loads.

- Eventless methods are based on the simultaneous matching of multiple loads, and they find the set of energized appliances that best fit the measured load. Most efforts from researchers on this side are given methods based on probabilistic models such as hidden Markov models (HMM) [27, 28, 29, 30]. Other approaches includes sparse coding [31, 32], genetic algorithms [33] and integer programming [34, 35]. These methods provide better disaggregation performance [13] and are less sensitive to edge detection. However, they are more susceptible to the fundamental problem presented in the previous section.

Regarding the programming approaches, very few of them were focused on expanding the classical CO model. Some of these works are going to be discussed in the next subsection.

1.3.1 Related Work Based on Optimization

[33] shows the CO pointed in Hart's paper and discusses their equivalency with the knapsack problem. However, their paper is focused only on verifying Hart's statement regarding the MS. Five metaheuristics optimization approaches are evaluated. Their work does not expand the appliance model, such as new constraints to enhance the identification accuracy. They conclude that it is hard to disaggregate loads with similar power draws and proposes as future work a multi-objective optimization approach.

Kolter and Jaakkola in [27] formulate the NILM problem as a convex quadratic programming problem. The authors consider an extension to HMMs, called additive factorial hidden Markov models. Furthermore, authors in [27] describe an unsupervised learning procedure. However, the method needs a regularization parameter that changes for each problem. In addition, the optimization function is made over the full set of time periods, which make the method computationally expensive.

[34] formulates the NILM problem as an integer quadratic programming problem. The technique represents the problem, as a combination of waveforms from multiple loads. At any given one period of current, the overall load current is represented as a superposition of each current of the operating appliance. The overall current waveform is considered to be influenced by the waveform of each individual appliance as shown in the Figure 1.4. However, as disadvantages, the technique requires data at a very high sampling rate. The model is based on the reading of one cycle (60Hz or 50Hz), so a large number of points from each cycle is necessary. In addition, as mentioned by Zeifman in [13], the approach is somewhat naive since features are usually more robust than just unprocessed waveforms.

Finally, authors in [35] propose a load disaggregation method based on integer programming. The work proposes enhancements to CO, such as state transition diagram and median filtering, to deal with the MS. Most enhancements in [35] are included as an intensive preprocessing rather than constraints. In addition, their model relies only on instantaneous load samples. Hence, their model is limited to the use of constraints that do not depend on time measurements.

1.4 Objectives

Based on the efforts made by previous works, the goal of this work is to represent and solve the NILM problem as a MILP problem. We are especially concerned about expanding the classic CO problem in order to deal with complex load signatures and handle the MS problem.

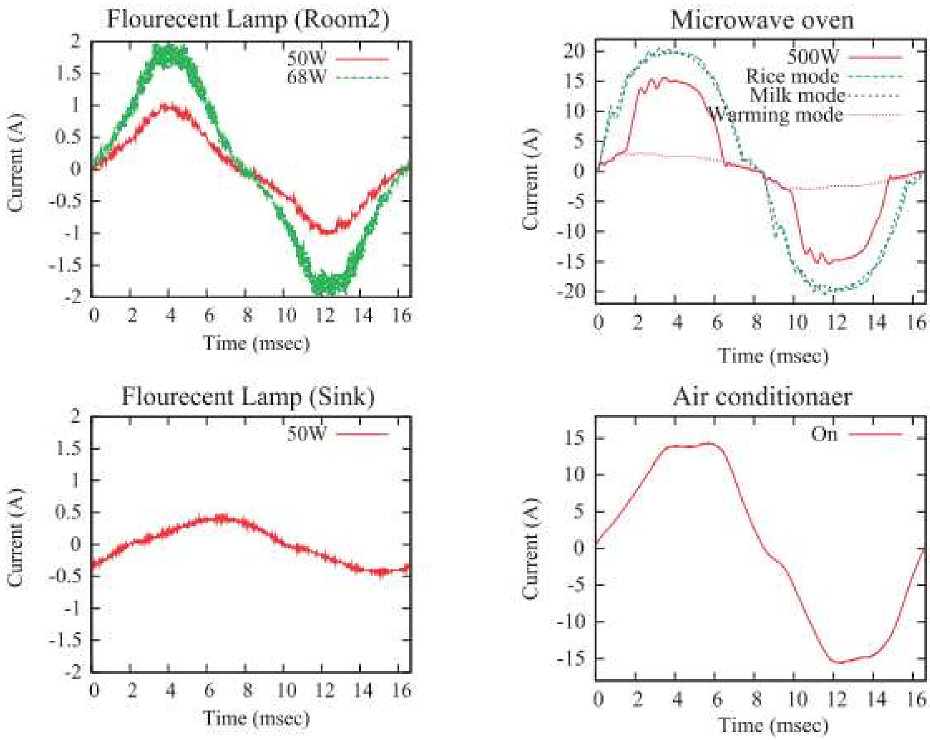


Figure 1.4: Example of current waveforms for different loads, used by [34]

The CO formulation was not very much explored in previous works and we seek to propose new constraints for optimizing the identification of loads. In addition, we investigate strategies to improve the running time of the problem, which is also one of its main weakness. Another objective of this work is to compare the proposed formulation with classic works. Most previous works are based on either event based techniques or probabilistic models and this work allows researchers to expand the range of possibilities and approaches to solve the NILM problem.

The Figure 1.5 shows the steps to approach the NILM problem using optimization. The green blocks refer to steps that are applied only to an unsupervised approach while the orange blocks are the steps for a supervised approach. As a difference, for the supervised approach, the input parameters are acquired from the appliance's data. In the unsupervised approach, those input parameters are extracted from a preprocessing step. One more goal of this work is to formulate an optimization algorithm with input parameters that can be acquired in both a supervised or unsupervised approach. The focus is in the optimization algorithm rather than in the methodology for extracting those input parameters. The methodology for extracting those features is another challenge. Strategies are presented in the Appendix B and C.

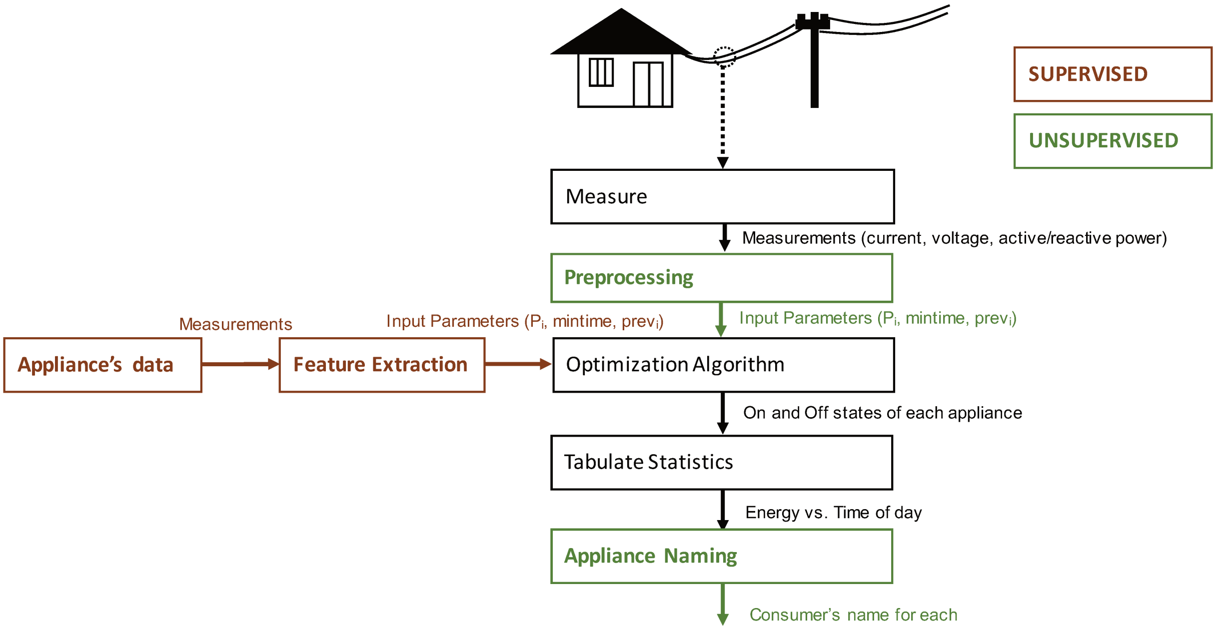


Figure 1.5: Steps for an optimization approach of NILM

1.5 Contributions

The main contributions of this dissertation are as follows:

- The NILM problem is represented as a MILP model, in which a new set of linear constraints is proposed to efficiently model the load signatures of the appliances.
- In order to enhance the computational performance, the proposed NILM is solved using a window-based algorithm in which the overall problem is segmented into small, coupled sub-problems that can be efficiently solved via MILP solvers.
- The proposed algorithm is suitable to smart meters with limited data such as only the current or the active power in low sampling rate. In addition, the proposed technique is also suitable for more measurements, if available.
- The model relies on generalized parameters that can be acquired from aggregated data.
- The MS is avoided in the test cases.

1.6 Organization of the Work

The next sections of this dissertation go in this way:

- Chapter 2 describes the background work that is used as the basis for the remaining chapters.
- Chapter 3 describes the main contributions of this work. A window-based math model for modeling the load signatures.
- Chapter 4 shows experiments and test cases with results for supervised settings.
- Finally, Chapter 5 presents the main conclusions and future work.

Additionally, three appendices are presented:

- Appendix A presents the full proposed mathematical model.
- Appendix B describes a supervised strategy for extracting the input parameters of the presented model.
- Appendix C presents unsupervised experiments.

Chapter 2

Fundamentals

This chapter presents the core concepts to support the next chapters. First, a general description of smart meters is given. Next, Hart's classical NILM technique is going to be described. Then, an overview of mathematical optimization is presented along with a linearization approximation. Finally, the NILM technique formulated as a CO problem is presented.

2.1 Smart Meters

One of the goals of this work is to formulate an algorithm suitable to smart meters. A general description of smart meters is given in this section along with their main limitations.

Smart meters are devices that record electrical measurements and provides them to the customer in near real time [36]. The measurements are acquired and displayed to the customer in increments of minutes, usually 5-minute intervals up to hourly intervals [37]. The power utility has access to those measurements for billing. Many power utilities around the world are currently replacing old electric-mechanical meters to smart meters. In Europe, there are more than 154 million units installed [38]. There's a strong engagement of power utilities with load disaggregation due to their interest in providing a better service to their customers [7].

As noted by Makonin in [39], cost is a key consideration for a power utility when installing millions of smart meters. Most smart meters from power utilities provide their measurements in low-frequency which is defined as a sampling rate lower than 1Hz. Makonin also notes that high-frequency load disaggregation is not a viable option since a data transmission rate higher than 1Hz might cause network and connection channels saturation. In addition, smart meters must be inexpensive in terms of computation and storage usage. Finally, [39] observes that sensors providing high-performance measurements are expensive which might cause an adoption barrier by the utility. Another constraint is the number of available measurements provided by smart meters. Power utilities usually provide simple measurements such as the active power which also constraints the development of algorithms for disaggregation. The addition of other types of measurements would also imply in cost increment in their development.

Due to those practical limitations in smart meters, one of the main requirements desirable for NILM algorithms is their ability to perform disaggregation in low frequency and with limited information.

2.2 NILM as a Pattern Recognition Problem

Hart's pattern recognition method for load disaggregation is described in this section. Most event-based NILM techniques follow similar steps. The technique is fundamentally based on event detection and classification of a pair of events with opposite direction. The major attention in this section is given to the characteristics of the algorithm. For further details, the reader may refer to [9] and [10]. The Figure 2.1 illustrates the core idea of Hart's method which is fundamentally based on the detection of steps in power measurements. Edges with similar step size and opposite directions can be associated with the same appliance. For example, Figure 2.1 shows the activation of a heater (about 1000 W) followed by a refrigerator (about 200W). Next two negative edges are observed, which can be associated to the deactivation of those same appliances.

The core of Hart's algorithm is illustrated in the chart presented in the Figure 2.2. As a summary, the measurements are first normalized. Next, their edges are detected and linked to similar ones. Next, appliances models are built based on groups (clusters) of edges with opposite signs. Those models are linked to new edges detected in real time. Finally, statistics can be inferred from those general models and those appliances can be named. More details on each item of the Figure 2.2 are going to be discussed in the following items:

- **Measure Power and Voltage:** Power and RMS voltage are averaged over intervals of one second. Voltage, real power, and reactive power are digitally acquired based on a high sampling rate.

- **Normalize Power:** The Equation (2.1) is used in order to compute the normalized total load power. The normalization translates into what the power would be if the utility provided a steady voltage and the load obeyed a linear model.

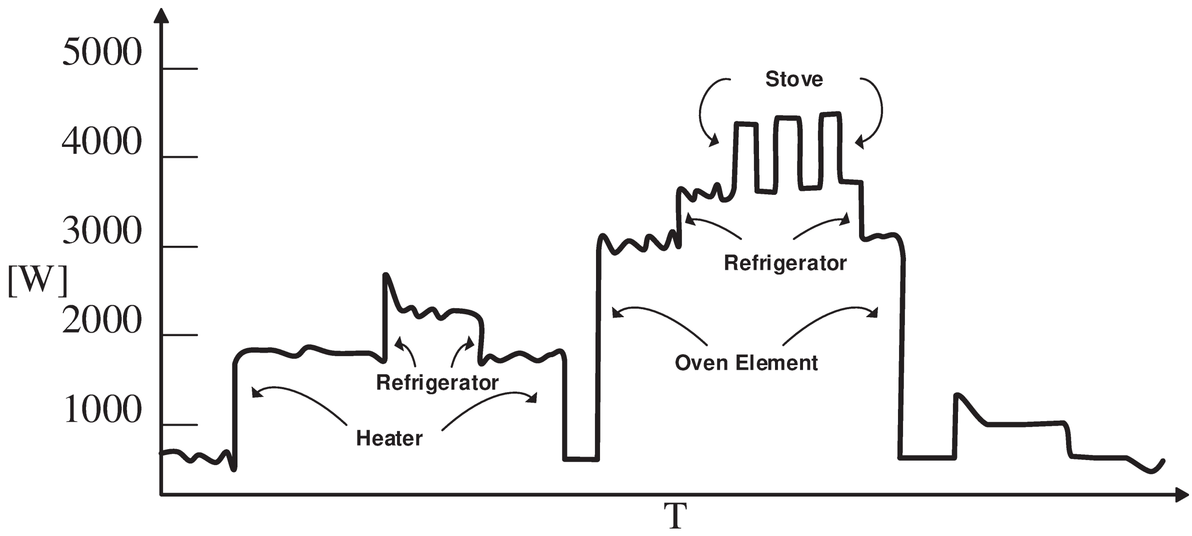


Figure 2.1: Illustration of edges that could be detected from a typical power measurement.

$$P_{norm}(t) = \left(\frac{127}{V(t)}\right)^2 P(t) \tag{2.1}$$

- **Edge Detection:** The sizes of step-like changes and their times of occurrence (time stamp) are obtained by an edge detection algorithm. Hart defines two types of periods: steady period and periods of change. A steady period is defined as periods where the input does not change more than a threshold during a certain number of samples. Any other period that does not meet this criterion is defined as periods of change. The step size is calculated as the average of a steady period. All the steps that are lower than a given noise level are discarded. The time stamp is provided by the time of the first sample in a changing period. The edge detection process is illustrated in the Figure 2.3.

- **Cluster Analysis:** The previously identified events are grouped into clusters. Each cluster is a set of events which are all close to each other. The Figure 2.4 illustrates some clusters that might appear after the edge detection process. This task might get very difficult if many clusters of different appliances are intersecting each other. More details regarding Hart's clustering algorithm are found in the MIT 1985 technical report available in [9].

- **Appliance Models:** Models of appliances with simple ON/OFF states or multiple states are constructed. For ON/OFF appliances, the model is constructed by taking the centroid of symmetrical positive and negative clusters. In other words, a pair of similar clusters representing positive and negative edges. The pair of centroids is matched involving a tolerance criteria. Models of appliances with multiple states are constructed by combining sequences of simple states.

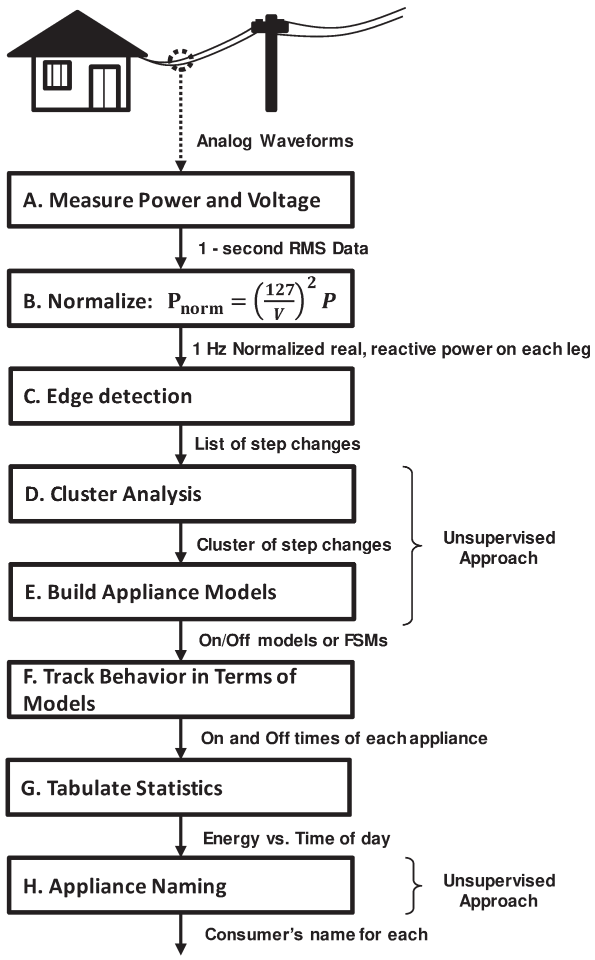


Figure 2.2: Core steps of Hart's algorithm [10].

- **Track Behavior:** The available appliance models are tracked using a decoding approach described in [40]. The approach is analog to a communication model where each appliance is considered a transmitter broadcasting electrical measurements as information. The house wiring is the communication channel and finally, the algorithm is the receiver. The decoding algorithm is based on a generalization of the Viterbi algorithm.[1] Giving a message source, the algorithm corrects errors that may occur in the channel such as insertions, deletions, and merges.

- **Tabulate Statistics:** Energy statistics can be computed from the power levels and time stamp obtained in the previous step. Some example of useful statistics is the operating power, total energy, and energy broken down by hour/day/week. The energy statistic is also useful for a conflict resolution step to check if there are more appliances activated than it should. It could be checked if the energy inferred from all appliances is higher than the total house power. This situation is likely to happen in cases where both OFF

[1]Optimal decoding technique based on dynamic programming

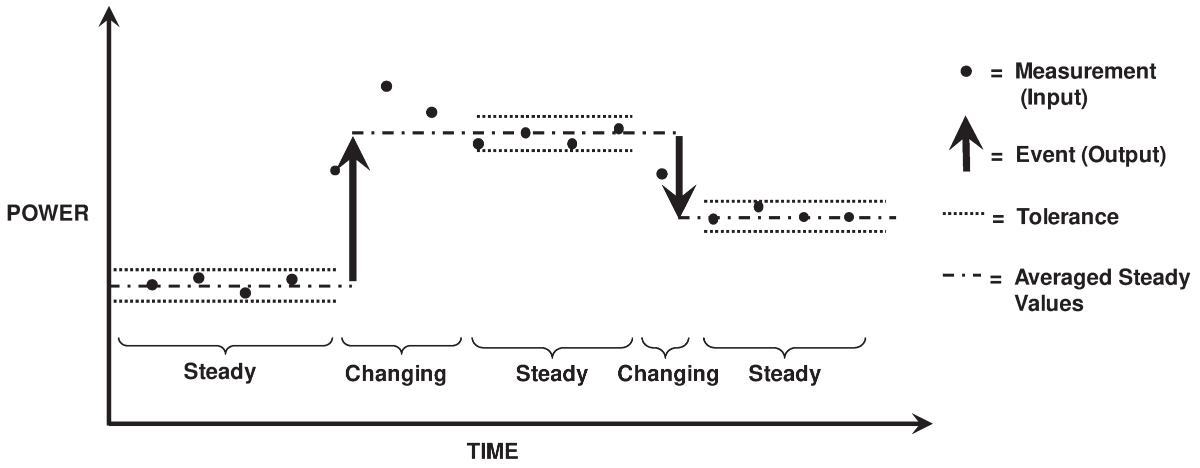


Figure 2.3: Illustration of the edge detection process of Hart's method [10].

and immediately following ON event of an appliance are missed.

- **Appliance Naming:** Finally, a name should be assigned to each appliance model detected from the collected data. Statistics based on the typical duration of an appliance can be used for this task. Another useful information is the operating power level (127V vs 220V) of the device which is useful to check if the appliance is a single phase or two phase element.

Although Hart's technique is quite old, it is still relevant since many other techniques are still based on the very same principles of event detection. The NILM problem may also be approached using eventless techniques. The next section is going to introduce mathematical optimization in order to describe the eventless approach.

2.3 Mathematical Programming

Mathematical programming is the process of minimization or maximization of an objective function of many variables, subject to constraints on the variables [41]. One of the crucial parts of mathematical programming is the model building process. A mathematical model involves a set of rules, constraints, equations, inequalities and logical dependencies that correspond to some relationship to the real world [42]. The equation (2.2) shows an example of optimization model.

$$\max_{x_i} \quad \sum_{i=1}^{n} v_i \, x_i \tag{2.2}$$

subject to

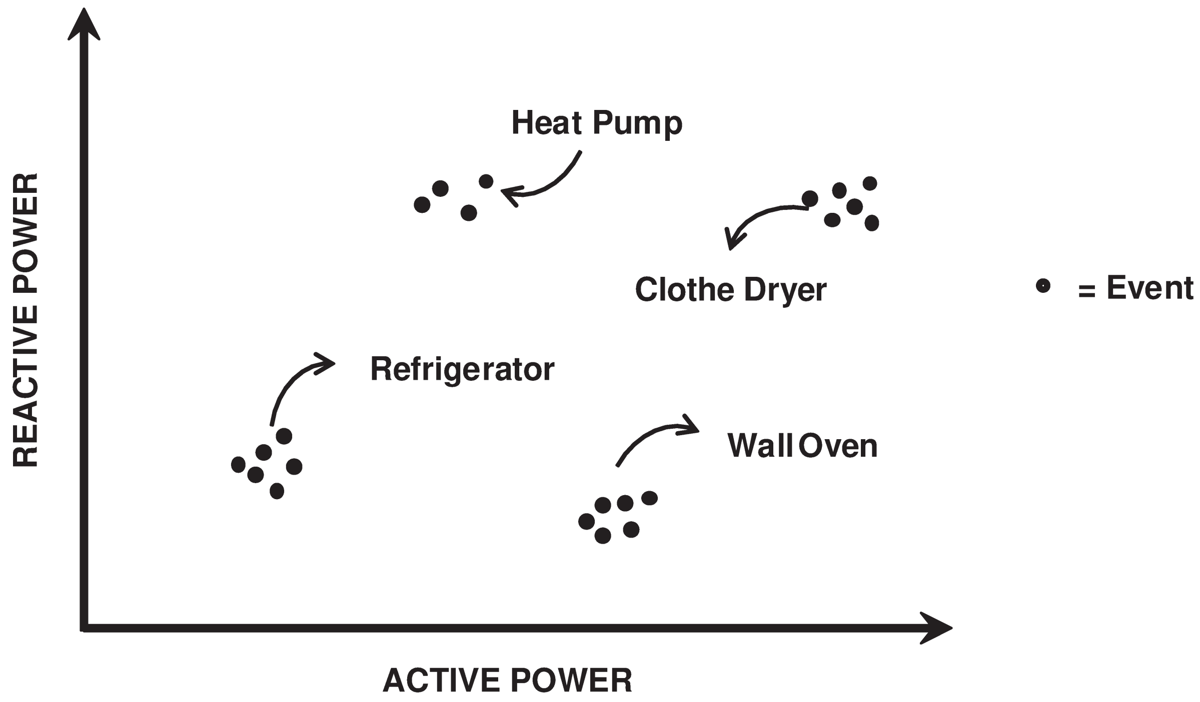


Figure 2.4: Hypothetical scenario of a clustering space.

$$\sum_{i=1}^{n} w_i \, x_i \; \leq \; W \quad \forall \, x \in \{0, 1\} \tag{2.3}$$

Equation (2.2) is a classic integer programming model also known as the knapsack problem. Given a set of $i = 1..n$ different items, where each item has a specific value v_i, the main goal is to maximize the overall profit given that each item has a weight w_i and it is not possible to overcome the maximum limit W. The variable of this problem x_i is an integer binary variable, hence, the name of this type of problem.

Most mathematical programming models can be classified into either linear programming (LP) models, non-linear programming (NLP) models or integer programming (IP) models [43, 41, 42]:

- **Linear Programming (LP):** The variables in the optimization function and in the constraints are linear. Involves three main features: a single linear expression (objective function) to be maximized or minimized; a series of constraints in the form of linear expressions that must meet a given inequality (for example >=, <= and =); and a set of coefficients on the right-hand side of the constraints. LP models assume that their variables may have fractional values.

- **Non-linear Programming (NLP):** When non-linear terms are incorporated into a model, we have a NLP model. Those models are usually more difficult to solve and more computationally expensive. The objective function and/or the constraints contain nonlinear variables. Some examples of non-linear terms are the product of two variables or the square of a variable.

- **Integer Programming (IP):** Assumes that some or all of the variables must take integer values (whole numbers). A special case of IP problems is those where their variables must assume binary values (one or zero). Those models are more difficult to solve than standard LP models.

Variation of the LP, NLP and IP models also exists. For example, a quadratic programming (QP) model has a quadratic objective function and linear constraints. A mixed integer programming (MIP) model combine both continuous and discrete (integer or binary) variables in the objective function. When the objective function strictly assumes linear terms, we have a mixed-integer linear programming (MILP) model. Finally, combinatorial optimization (CO) models is a branch of discrete optimization problems in which their solutions may be expressed in some kind of combinatorics such as shortest paths, sets, combinations, and permutations. CO models may also be formulated as IP problems.

Algorithms are deployed in order to solve a mathematical problem. They are referred to as solvers. Their solutions may either assume exact values under a given convergence or approximate solutions from heuristics. As an example, it is noted the commercial optimization software CPLEX [43], owned by International Business Machines (IBM). CPLEX solves linearly or quadratically constrained problems where the objective function can be expressed as a LP or a QP function. The variables in the model may assume either continuous or integer values.

Strategies and/or approximations can be applied to programming models with the purpose of improving the computational performance or simplification. One possible application will be seen in the next subsection.

2.3.1 Linearization of an Absolute Objective Function

The conversion of non-linear optimization functions to linear terms is always attractive. As an advantage, linear formulations are faster and guarantee the global solution. Boyd in [44] presents linear approximations of non-linear functions with a norm in the objective function. One of the approximations is made in a formulation similar to the Equation (2.4). The equation seeks to find the product of the variable x with the matrix A that approximates to b.

$$\min_{x} \quad |A\,x - b| \tag{2.4}$$

Where $A \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^{m}$ are problem data, $x \in \mathbf{R}^{n}$ is a variable. One of the possible ways to avoid the non-linearity of this function could be taking the square of $A\,x - b$. This way, the problem would be modeled to a quadratic programming problem. Alternatively, this problem may also be modeled to a linear programming problem. The linear model has as advantage that it is computationally faster. A linear formulation can be made by taking a new variable $t \in \mathbf{R}^{m}$ to be minimized. Next, new constraints can be used in order to approximate the absolute term in (2.4) into linear terms. The Equation (2.5) presents the norm approximation problem cast to a linear programming problem:

$$\min_{t} \quad t \tag{2.5}$$

subject to

$$Ax - b \leq t \tag{2.6}$$

$$Ax - b \geq -t \tag{2.7}$$

The approximation given by 2.5 is going to be deployed in the next section in order to make the model suitable to a MILP solver.

2.4 NILM as a Combinatorial Optimization Problem

The NILM problem can also be formulated as a combinatorial optimization problem, which does not require the detection of events. This formulation assumes that the whole house measurement can be decoded into individual components. This section presents the CO formulation presented by Hart in [10].

Let the measured variable (current or power) in the input of the house be given by $P(t)$, for each time t. The objective of the NILM would be to decode $P(t)$ into power states P_i, $\forall\, i \in \{1, ..., n\}$. The Equation (2.8), shows this assumption[10].

$$P(t) = \sum_{i=1}^{n} x_i(t)\; P_i + \epsilon \tag{2.8}$$

Where $x_i(t) \in \{0, 1\}$ is a boolean variable that decides the status of the power state i, at

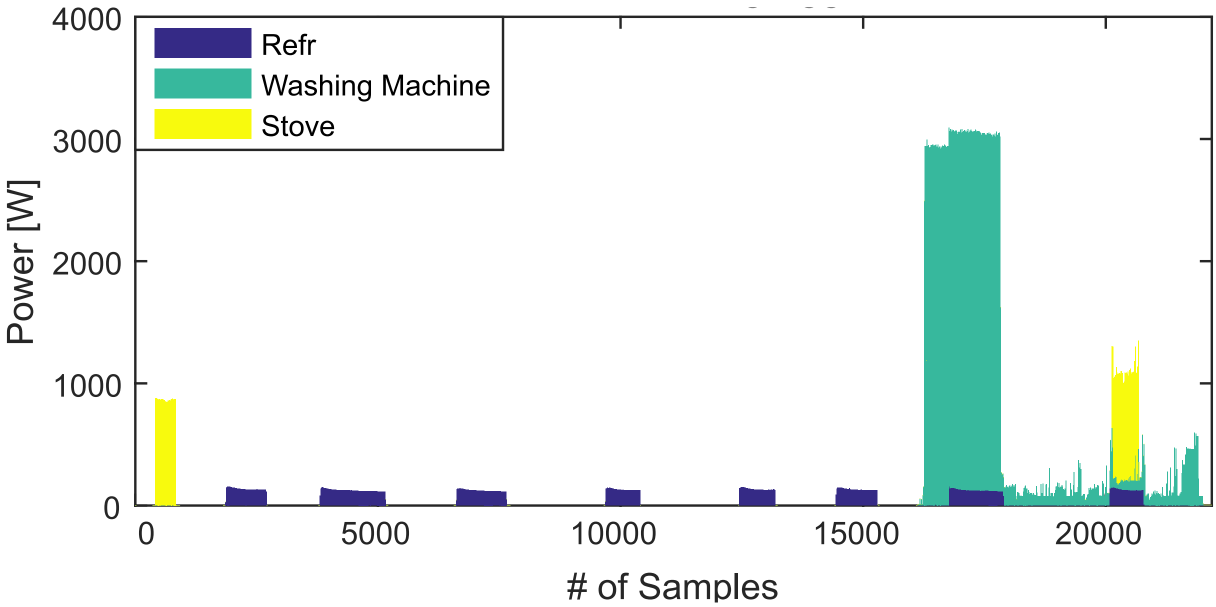


Figure 2.5: Illustration of scenario with three appliances

time t. ϵ is the noise error due to the approximation of $P(t)$ into a set of P_i states. n is the total number of power states for all appliances. Each appliance is associated with one or more power states. For example, an ON/OFF appliance (e.g., a stove) could be represented by a single state, while a washing machine could be represented by multiple states, since its power consumption changes over time.

Eventually, the goal of the NILM problem is to minimize $|\epsilon|$. The equation (2.9) rewrites (2.8) as a CO problem.

$$\min_{x} \quad \left| P(t) - \sum_{i=1}^{n} x_i(t)\ P_i \right| \tag{2.9}$$

Equation (2.9) aims at finding the combination of power states P_i that best approximate the measure $P(t)$. When other types of measurements are also available (such as reactive power, harmonics or distortion factor), the classical problem in (2.9) may include these measurements in a vector. As an illustration, the Equation (2.10) also includes the reactive power measurement $Q(t)$ and reactive power states Q_i.

$$\min_{x} \quad \left| \begin{bmatrix} P(t) \\ Q(t) \end{bmatrix} - \sum_{i=1}^{n} x_i(t) \begin{bmatrix} P_i \\ Q_i \end{bmatrix} \right| \tag{2.10}$$

In order to illustrate the terms P_i and $x_i(t)$ from the Equation (2.8), let's consider the scenario of the Figure 2.5. The presented scenario has three appliances: stove, washing machine, and a refrigerator. As shown in the Table 2.1, the refrigerator is approximated to the state 130W,

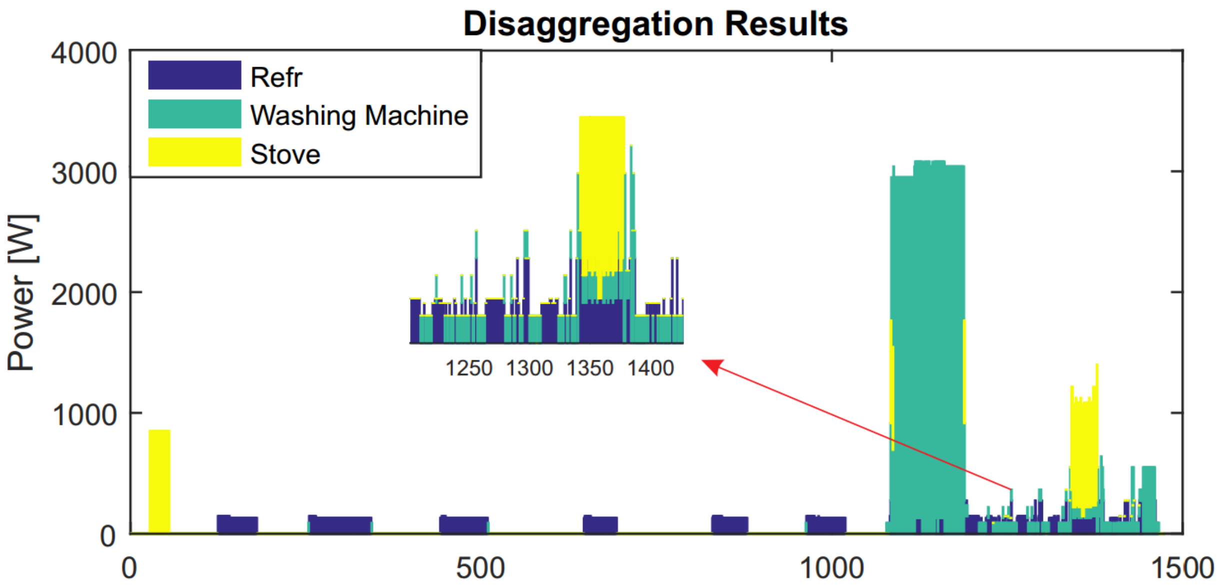


Figure 2.6: Illustration of the MS problem that occurs in the CO classic formulation

the stove is approximated to the state 850W, and finally, the washing machine is approximated to the three states 3000W, 90W and 500W. At the sample near to $t = 5000$ we observe that only the refrigerator is active, which corresponds to the state $i = 1$ in the Table 2.1. Hence, the expected value of the binary variable $x_i(t)$ is $x_i(5000) = [1, 0, 0, 0, 0]$.

Table 2.1: Example of possible states for the illustrated scenario

appl	i	P_i
Refrig.	1	130
Stove	2	850
Washer 1	3	3000
Washer 2	4	90
Washer 3	5	500

As discussed in the introduction, one of Hart's mains critiques of this formulation is the possible confusion made from loads with similar power states (MS problem). For example, consider a scenario where the set P_i contains the states 100W, 200W, and 302W. A measurement $P(t)$ of 301W could continuously switch between the states 100W + 200W and the state 302W due to the measurement noise.

The Figure 2.6 illustrates the MS problem for a hypothetical scenario with three appliances (refrigerator, washing machine, and a stove). The refrigerator is activated and deactivated multiple times in order to fit in the noise of the washing machine. The next subsection will reformulate the Equation (2.9), which will allow for more advanced constraints, presented in Chapter 3.

2.4.1 Mixed-Integer Linear Programming (MILP) Formulation

While simple, the CO formulation in the Equation (2.10) is not linear due to the absolute term in the objective function. In order to make the formulation suitable to linear solvers - which is faster and guarantees the optimal solution - the Equation (2.10) can be reformulated as a MILP problem, shown in (2.11)–(2.15). The linearization of the absolute function is made in an analog way of the linearization presented in the Section 2.3.1.

$$\min_{x_i(t)} \quad \sum_{t \, \in \, T} \delta_P(t) + \delta_Q(t) \tag{2.11}$$

subject to

$$P(t) - \sum_{i \, =1}^{n} x_i(t) \; P_i \; \leq \; \delta_P(t) \tag{2.12}$$

$$P(t) - \sum_{i=1}^{n} x_i(t) \; P_i \; \geq \; -\delta_P(t) \tag{2.13}$$

$$Q(t) - \sum_{i \, =1}^{n} x_i(t) \; Q_i \; \leq \; \delta_Q(t) \tag{2.14}$$

$$Q(t) - \sum_{i=1}^{n} x_i(t) \; Q_i \; \geq \; -\delta_Q(t) \tag{2.15}$$

Where $\delta_P(t)$ and $\delta_Q(t)$ in (2.11) are the errors in the active and reactive power. Those errors are defined as the difference between the aggregated power states and the actual measurement (i.e., the approximation error). Besides the linear formulation, another difference from (2.10) is that the time T is included in the objective function. The advantage of doing so is the possibility of adding new time-dependent constraints in order to improve the model's accuracy. Those constraints are going to be seen in the next Chapter.

2.5 Summary

This chapter has presented the base work to substantiate the next chapters. First, smart meters were described in order to show that NILM algorithms should be suitable to low frequency and limited data. Next, Hart's algorithm framework was described which is one of the main event-based techniques. Next, an overview of mathematical optimization was presented, which is one of the main tools of this dissertation. Finally, the CO formulation was presented, which is the core formulation of this work. The next chapter is going to explore some further possibilities of the CO formulation.

Chapter 3

Expanding the Combinatorial Optimization Model

As shown in the introduction, very few progress was observed in the CO formulation. The Chapter 2 described the CO model in more details. This chapter proposes new constraints and strategies for expanding the CO model. We are especially interested in constraints for modeling the signature of a load and improve its performance. First, a visual intuition is presented. Next, three set of constraints are proposed for defining a load signature such as the transition of states and minimum time. Strategies and constraints for decreasing the computation time are also introduced. Finally, the full proposed model is presented.

3.1 Visual Intuition

Before introducing the set of equations, a visual example is going to be presented. The disaggregation problem can be imagined as a puzzle where the goal is to find the best set of pieces to fit a given space. An example of this scenario is illustrated in the Figure 3.1. In the example, the goal is to combine some of the colored blocks on the left in order to create the gray image on the right.

In order to handle this problem, first, a model for each of the pieces could be created. This model could consider the height and width of a rectangular block. Next, more complex pieces could be created as a set of rectangular blocks, such as the purple block on the button right. Finally, in order to solve the puzzle, this scenario could be treated as a combinatorial problem.

The proposed NILM model is similar to the same analogy. The goal is to find the set of appliance's models that better fit in the whole aggregated power of a window of measurements.

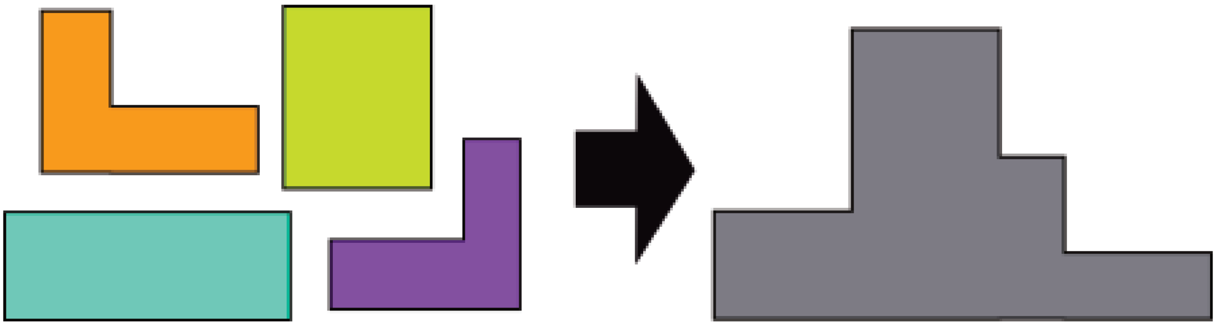


Figure 3.1: Illustration of the problem with a puzzle

The only difference is that a minimum width is defined instead of a fixed width. More details are presented in the next section.

3.2 Load Signature Constraints

In this section, a new set of constraints are presented to model the load signatures. Many of those constraints are similar and inspired from models of the operation of thermal units in the unit commitment problem, proposed by authors in [45]. The Figure 3.2 presents the load signature of two hypothetical appliances: a washing machine and a stove. The orange line is the original measurement while the black lines are their approximation.

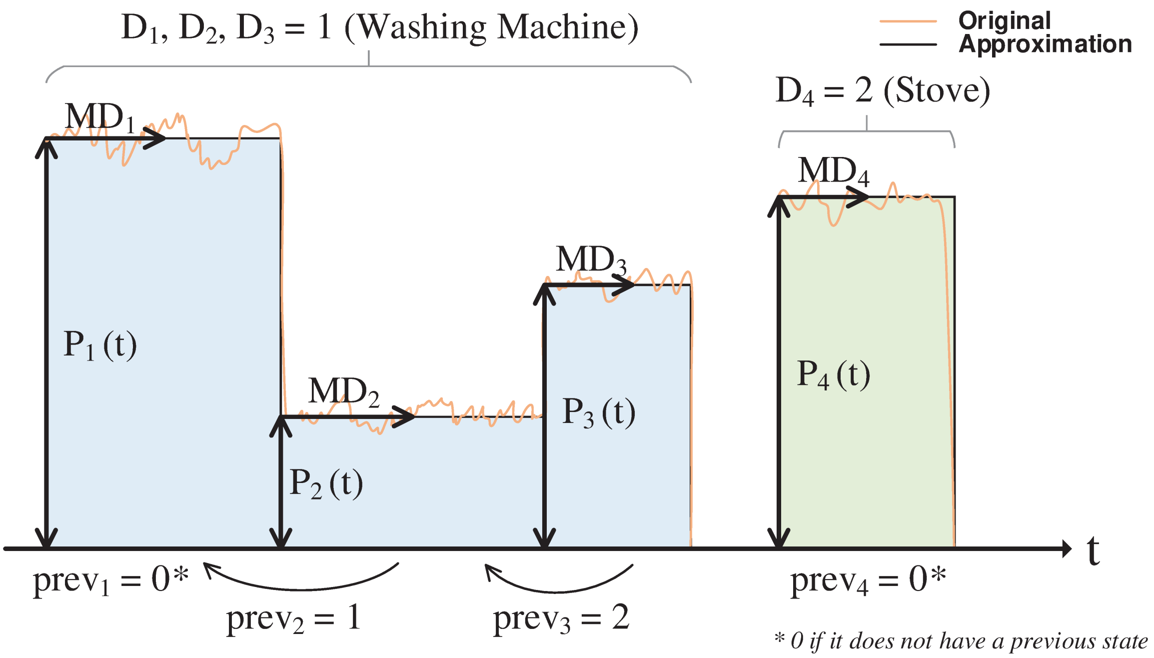


Figure 3.2: Two hypothetical load signatures to be modeled

The washing machine's load signature can be approximated using three power states ($P_1(t)$,

$P_2(t)$ and $P_3(t)$) while the stove's has one single power state ($P_4(t)$). In order to model these loads, the following hypotheses are considered:

- Multiple states $P_i(t)$ (also multiple reactive power states, if available) from the same appliance cannot be activated simultaneously;
- Some loads work as finite-state machines, i.e., a given state is only activated if another state of the same appliance has finished;
- A state i can have a minimum time MD_i in which it should remain ON.

The former hypotheses are used to formulate a set of constraints used to efficiently represent the load signatures within the proposed MILP model. A detailed analysis of each one of the former hypotheses is shown in the following subsections.

3.2.1 Avoiding Multiple States From the Same Appliance

Parameter D_i identifies the appliance's index of each state i. For example, the states $i = 1, 2, 3$ in Figure 3.2 are associated to the washing machine, that is, $D_1 = D_2 = D_3 = 1$, where the number 1 means "washing machine". Likewise, D_4 refers to the stove, identified by the number 2. In order to avoid simultaneously allocation of different power states from the same appliance, the constraint (3.1) can be included.

$$\sum_{i \in S | D_i = j} x_i(t) \leq 1 \quad \forall t \in T, j \in D \tag{3.1}$$

Constraint (3.1) limits the sum of the states $x_i(t)$ from the same appliance to one. This way, $x_1(t)$, $x_2(t)$ or $x_3(t)$ cannot be simultaneously activated since D_1, D_2 and D_3 are associated to the same appliance.

3.2.2 Linking the Transition Between Power States

In Figure 3.2, power state $P_2(t)$ should be ON only if the power state $P_1(t)$ has finished. Likewise, we could also fix the power state $P_3(t)$ to be ON only if $P_2(t)$ has finished. The goal is to include a constraint that allows a specific power state to be activated only if a previous one (from the same appliance) has finished. In order to do so, two binary variables are used to determine the transition from an ON state to an OFF state, and vice versa. The two variables are called $up_i(t)$ (turned ON) and $dw_i(t)$ (turned OFF). Then, the linking constraints are given by (3.2)–(3.3).

$$x_i(t) - x_i(t-1) = up_i(t) - dw_i(t) \quad \forall i \in S, t \in T \tag{3.2}$$

$$up_i(t) + dw_i(t) \leq 1 \quad \forall i \in S, t \in T \tag{3.3}$$

In constraint (3.2), $up_i(t)$ will be 1 only if the decision variable $x_i(t)$ makes a transition from 0 to 1, at time t. Likewise, $dw_i(t)$ will be 1 only if $x_i(t)$ makes a transition from 1 to 0, at time t. Finally, (3.3) prevents $up_i(t)$ and $dw_i(t)$ to be simultaneously 1.

Using constraints (3.2)–(3.3) and the parameter prev_i, we can now link the transition between two states using the parameter $prev_i$ in the equation (3.4).

$$up_i(t) = dw_{\text{prev}_i}(t) \quad \forall i \in S, t \in T \mid \text{prev}_i > 0 \tag{3.4}$$

As an example, the state $i = 2$ of the washing machine in Figure 3.2 can only change from OFF to ON (i.e., $up_2(t) = 1$) if the state $i = 1$ has change from ON to OFF (i.e., $dw_1 = 1$), at a given time t.

3.2.3 Minimum Active Time

The last proposed hypotheses is a minimum active time of a state. Parameter MD_i in Figure 3.2 establishes the minimum number of time samples in which the state i should be kept activated. The constraint in (3.5) is proposed to carry out this process.

$$\sum_{k\,=\,t}^{t+MD_i-1} x_i(k) \geq MD_i\,[x_i(t) - x_i(t-1)]$$

$$\forall i \in S, t \in 2 \ldots |T| - MD_i + 1 \tag{3.5}$$

Where $|T|$ is the total number of samples, i.e., the last sample in T. Constraint (3.5) forces $x_i(t)$ to be 1 for at least MD_i time samples.

3.3 Window-based formulation

The size of the time set T in the Equation (2.11) increases the computational burden since the variable $x_i(t)$ is linked to the number of time periods. For example, if T corresponds to measurements of a full day, with one measurement per minute and 10 states, the number of binary variables $x_i(t)$ would be $24 * 60 * 10 = 14400$. The number of combinations to be solved by the algorithm would be 2^{14400} which is not solvable in polynomial time [46]. In order to shorten the complexity expansion, the set of time periods can be split into smaller, homogeneous windows. Let $\tilde{T} \subset T$ be the set of time periods within a window, where $\tilde{T} = \{T_0, T_0 + 1, \cdots, T_0 + m\}$. The parameter m is the number of time periods within the window, and T_0 is the initial period for each window. The Figure 3.3 illustrates the separation of the time periods where T is split into three subsets of time periods $\tilde{T}$.

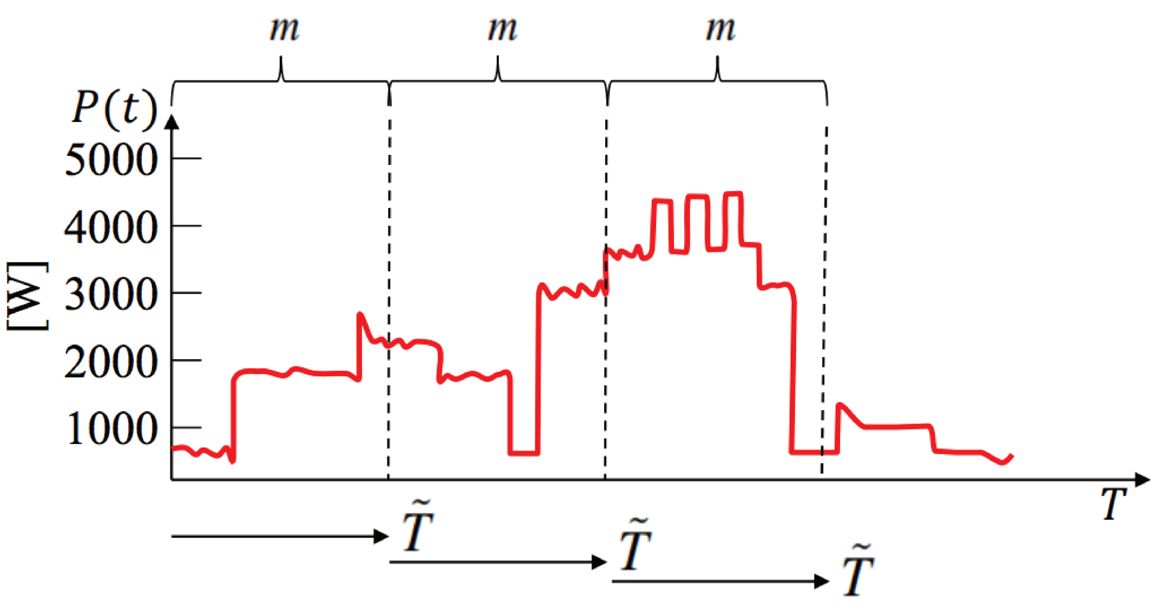


Figure 3.3: New time space in a window-based algorithm

In order to decrease the computational burden, besides the window formulation, it is possible to define a constraint to deactivate all the devices when the readings of a window is lower than a given threshold (for example $TH = 20W$). The constraint (3.11) presents a constraint under those conditions.

$$x_i(t) = 0 \quad \forall t \in \tilde{T}, i \in S | P(t) \leq TH \tag{3.11}$$

The MILP problem can now be written using a sequenced optimization process as shown in Algorithm 1, where T_{end} is the last period of T. The previously presented constraints have to be adapted to a window-based case. The next two subsecions will present the adaptation of the transition between states and the minimum time to the Algorithm 1.

repeat

let $\tilde{T} = \{T_0, \cdots, T_0 + m\}$

solve

$$\min_{x_i(t)} \quad \sum_{t \,\in\, \tilde{T}} \delta_P(t) + \delta_Q(t) \tag{3.6}$$

s.t.:

$$P(t) - \sum_{i=1}^{n} x_i(t)\ P_i \ \le\ \delta_P(t) \tag{3.7}$$

$$P(t) - \sum_{i=1}^{n} x_i(t)\ P_i \ \ge\ -\delta_P(t) \tag{3.8}$$

$$Q(t) - \sum_{i=1}^{n} x_i(t)\ Q_i \ \le\ \delta_Q(t) \tag{3.9}$$

$$Q(t) - \sum_{i=1}^{n} x_i(t)\ Q_i \ \ge\ -\delta_Q(t) \tag{3.10}$$

let $T_0 = T_0 + m + 1$

until T_0 > T_{end};

Algorithm 1: NILM using a window-based algorithm

3.3.1 Window-Based Transition of States

The constraints presented in the Section 3.2.2 to model the transition of states is going to be re-formulated to a window-based case. The set of equations in (3.12)–(3.14) makes this translation from (3.2)–(3.3):

$$x_i(t) - x_i(t-1) = up_i(t) - dw_i(t) \quad \forall i \in S, t > T_0 \tag{3.12}$$

$$x_i(T_0) - X_i = up_i(t) - dw_i(t) \quad \forall i \in S, t = T_0 \tag{3.13}$$

$$up_i(t) + dw_i(t) \le 1 \quad \forall i \in S, t \in \tilde{T} \tag{3.14}$$

The constraints (3.12) and (3.14) are analog to (3.2) and (3.3), but updated to the new time set $\tilde{T}$. In addition, the starting time of each window T_0 is also used in the new formulation. The constraint (3.13) is new and links the state transitions between windows. The parameter X_i saves the state $x_i(t)$ of the last time period of each window, necessary for the initialization of $x_i(t)$ in the next window.

In a similar way to the constraint (3.4), the constraint (3.15) links the transition of states.

The main difference is that now the subset $\tilde{T}$ is used.

$$up_i(t) = dw_{\text{prev}_i}(t) \quad \forall i \in S, t \in \tilde{T} \mid \text{prev}_i > 0 \tag{3.15}$$

3.3.2 Window-Based Minimum Active Time

Now the constraints to model the minimum active time presented in the Section 3.2.3 will also be adapted to a window-based case. The constraint (3.5) is translated into three constraints in (3.16)–(3.18).

$$\sum_{k\,=\,T_0}^{T_0+G_i} [1 - x_i(k)] = 0 \quad \forall i \in S \tag{3.16}$$

$$\sum_{k\,=\,t}^{t+MD_i-1} x_i(k) \geq MD_i\,[x_i(t) - x_i(t-1)]$$
$$\forall i \in S, t \in G_i + T_0 \ldots T_f - MD_i + 1 \tag{3.17}$$

$$\sum_{k\,=\,t}^{T_f} x_i(k) \geq x_i(t) - x_i(t-1)$$
$$\forall i \in S, t \in T_f - MD_i + 2 \ldots T_f \tag{3.18}$$

Each new window is split into three time segments, one for each constraint, as illustrated in the Figure 3.4. The constraint (3.16) activates a certain state that was already activated at the end of the previous window, but for less than MD_i samples. Parameter G_i is the number of periods in which the state i must remain active at the beginning of each window. It is calculated as $G_i = \min\{T_f, [MD_i - NP_i]\,X_i\}$. NP_i is the number of time periods in which i has been activated in the previous window, given by the equation (3.19)

$$NP_i = \sum_{k\,=\,T_f-MD_i+2}^{T_f} x_i(k) \quad \forall i \in S \tag{3.19}$$

Constraint (3.17) is analog to the constraint (3.5). Finally, constraint (3.18) is used to represent the operation at the final portion of the window, when there are less than MD_i samples available. It forces a given state $x_i(t)$ to be ON until the end of the window, only if it has been

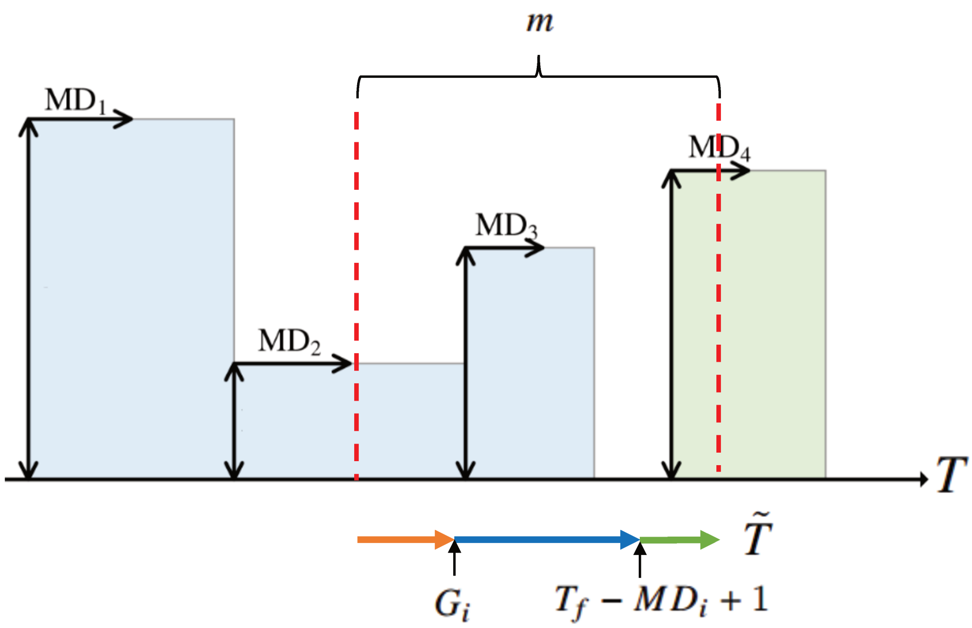


Figure 3.4: New time segments for each new window

activated at any moment within this final interval. The size of MD_i has as limit the length m of the window, i.e, $MD_i \leq m$.

In order to illustrate how this set of constraints work, suppose that the window in the Figure 3.4 has a size $m = 100$ samples. In addition, suppose that the stove (green load) has a minimum ON time of $MD_4 = 25$ samples. In case the stove is activated at sample 90 of the window, there are only 10 samples remaining until the window ends (which is lower than MD_4). The constraint (3.18) will force the state $i = 4$ to be ON until the window ends (until T_f). In addition, the variable G_4 will receive the value of 15 samples which is the number of samples left to be ON in the next window. In the next window, the constraint (3.16) will force the stove to be active for $G_4 = 15$ samples.

3.4 Full Proposed Model

The full proposed model for a window-based case is given by the Algorithm 2. As it will be illustrated in the chapter with experiments, this kind of optimization problem can be solved with the help of standard convex mathematical optimization software. The full formulation in details without shortening equations is available in Appendix A.

repeat

let $\tilde{T} = \{T_0, \cdots, T_0 + m\}$

solve

$$\min_{x_i(t)} \quad \sum_{t \, \in \, \tilde{T}} \delta_P(t) + \delta_Q(t)$$

s. t.

$$(3.7) \text{ ... } (3.18)$$

let $T_0 = T_0 + m + 1$

until $T_0 > T_f$;

Algorithm 2: Proposed NILM using a window-based algorithm.

3.5 Summary

This chapter has presented a set of constraints for modeling the load signature, which are the main contributions of this work.The classic NILM CO model was expanded for modeling load signatures in a computationally efficient way. Constraints were introduced for modeling the load signature based on three features: power state, minimum time and sequence of states. Finally, a window-based formulation was presented in order to decrease the computational burden.

Chapter 4

Experiments

Two supervised test cases are conducted to evaluate the performance of the NILM method proposed in Algorithm 2:

- Test Case A compares two experiments: one of them using only the active power and the other also including the reactive power.
- Test Case B presents uses a longer sampling period and more appliances.

In addition, the proposed method is compared with CO, implemented in Algorithm 1 and Hart's pattern recognition method. The pattern recognition method is obtained from the open-source NILM Toolkit (NILMTK) [47] implemented in Python. The simulation environment was the AMPL modeling language [41] and the commercial solver CPLEX [43]. The PC configuration was: Intel Xeon 2.4 GHz and 32 GB of memory.

4.1 The AMPds Dataset

The Almanac of Minutely Power Data Set (AMPds) [48] is chosen for evaluation. AMPds contains two years of electricity measurements at one-minute intervals. Those measurements are collected from a single household in Vancouver, BC, Canada. A total of 19 appliances from the circuit breaker are collected with measurements of active power, reactive power, apparent power, frequency, current, and voltage. The dataset was released in order to help researcher and related to test NILM models, systems, and prototypes. This dataset is chosen since it uses a sampling rate very close to commercial smart meters. In addition, this dataset offers a wide range of appliances to be modeled and tested. Besides electrical measurements, other types of consumption are also included in the dataset such as water and natural gas. The Table 4.1 lists the 19 loads along with their IDs.

Table 4.1: Loads measured by the data set AMPds

ID	Load
B1E	North Bedroom
B2E	Master/South Br
BME	Basement Plugs and Lights
CDE	Clothes Dryer
CWE	Clothes Washer
DNE	Dining Room Plugs
DWE	Dishwasher
EBE	Electronics Workbench
EQE	Security/Network
FGE	Kitchen Fridge
FRE	HVAC/Furnace
GRE	Garage
HPE	Heat Pump
HTE	Instant Hot Water Unit
OFE	Home Office
OUE	Outside Plug
TVE	Ent TV/PVR/AMP
UTE	Utility Room Plug
WOE	Wall Oven

4.2 Metrics

Two metrics adapted from [47] are used to evaluate the accuracy of the proposed methodology. The first one is referred as Error in Total Energy (ETE) shown in equation (4.1). ETE measures how well the energy consumed by each appliance was predicted. The second metric is referred as Error in Assigned Power (EAP), shown in the equation (4.2). EAP measures how well each appliance was correctly assigned at each time slice t. Both metrics are normalized by the appliance's total energy consumption $y_j(t)$. The second metric provides more insight since it evaluates the correctness of the identification for each time slice. Unlike the first metric, the EAP does not counterbalance appliances identified at different time periods.

$$ETE_j = \frac{|\sum_t y_j(t) - \sum_t \hat{y}_j(t)|}{\sum_t y_j(t)} \tag{4.1}$$

$$EAP_j = \frac{\sum_t |y_j(t) - \hat{y}_j(t)|}{\sum_t y_j(t)} \tag{4.2}$$

For a given appliance j, $y_j(t)$ is its true power measurement and $\hat{y}_j(t)$ is the predicted power at each time instant t. Ideally, both errors should be zero. However, the value of zero is never achieved due to the approximation of the load signature. Hence, even in cases where the NILM algorithm detects the correct load, an error would exist. It worth mentioning that both metrics can achieve error rates higher than 100%. For example, if the sum of predicted power $\hat{y}_j(t)$ is higher than twice the ground truth power $y_j(t)$, the error in ETE in higher than 100%.

The Figure 4.1 illustrates the difference of both metrics. In both graphs (ground truth and predicted), the very same energy amount is consumed by the same load. Hence the ETE is 0% due to no difference in energy usage. However, the EAP error is 200% since the algorithm made two errors: first, it missed the activation of the green load at the time that it was activated; second, it wrongly predicted the load was activated at a time that it should be OFF.

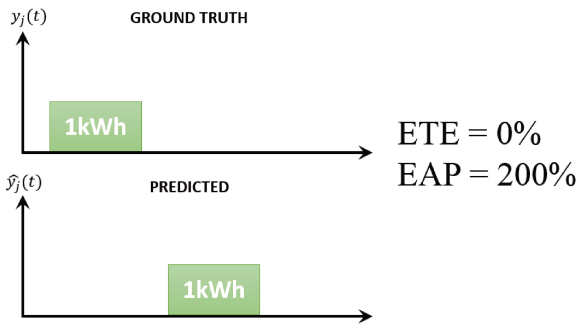


Figure 4.1: Illustration of the ETE and EAP error metrics

4.3 Supervised Case 1

The first test case is going to evaluate the performance of the algorithm for a one-day scenario. The input measurements along with the expected ground truth are presented in the Figure 4.2.

4.3.1 Experimental Settings

The input data was created with the power measurements of six appliances: Dryer (CDE); Dishwasher (DWE); Fridge (FGE); Heat Pump (HPE); Wall Oven (WOE); and Television/Entertainment (TVE). The full-time range has 24 hours, with 1440 measurements. All the appliances are represented by 14 power states, as shown in the Table 4.2. Moreover, Table 4.2 contains two columns with the previous states (prev_i) and the minimum number of periods (MD_i). Each appliance index D_i and abbreviation are also informed in Table 4.2. These parameters in Table 4.2 were acquired using the supervised approach described in Appendix B. As an example, $\text{prev}_4 = 3$ means that the state $i = 4$ is allowed to be ON only after $i = 3$. The window's length was

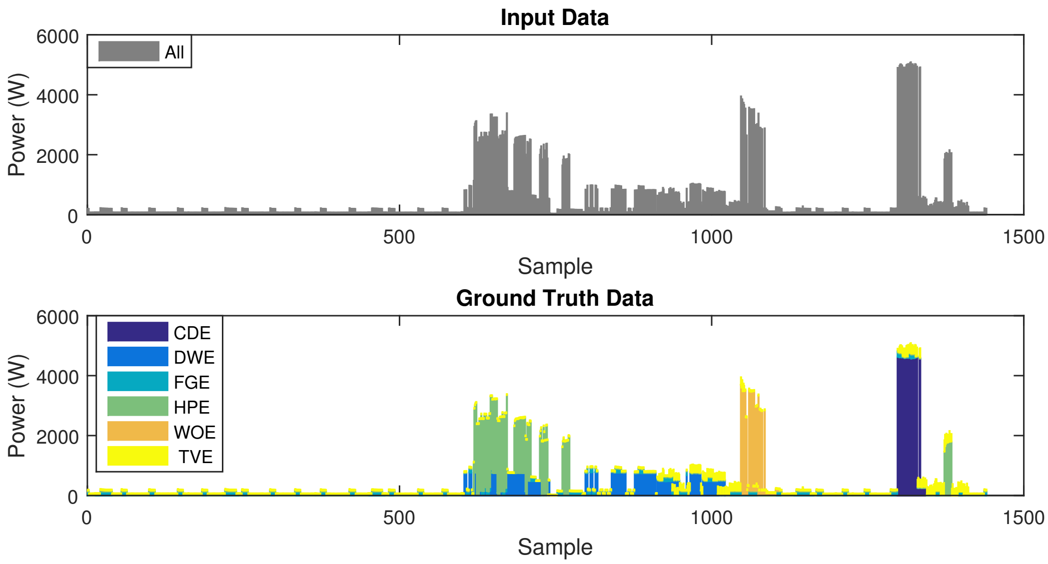


Figure 4.2: Model's input measurements and expected ground truth

chosen in order to minimize the running time. A window's length of 60 measurements was used.

Table 4.2: Input Data for the Experimental Setup in Tests Case A

appl	**i**	D_i	**prev**$_i$	P_i (W)	Q_i **(VAr)**	MD_i
CDE	1	1	0	4606	413	20
CDE	2	1	1	252	413	5
DWE	3	2	0	751	34	5
DWE	4	2	0	478	0	15
DWE	5	2	0	136	34	15
FGE	6	3	0	129	6	7
HPE	7	4	0	37	17	30
HPE	8	4	0	1807	324	10
HPE	9	4	7	2435	429	30
WOE	10	5	0	3442	141	5
WOE	11	5	0	3305	133	5
WOE	12	5	0	2796	130	1
TV	13	6	0	38	13	30
TV	14	6	0	239	31	30

4.3.2 Graphical Results

The difference between the results of the proposed model and the two other algorithms can be illustrated using graphics. Two figures are presented:

- Figure 4.3 presents the results when considering only the active power from the Table 4.2.

- Figure 4.4 presents the results when considering both the active and reactive power in the Table 4.2.

When comparing the Figure 4.3 with the Figure 4.4, the proposed model performs well even when using limited data (only the active power).

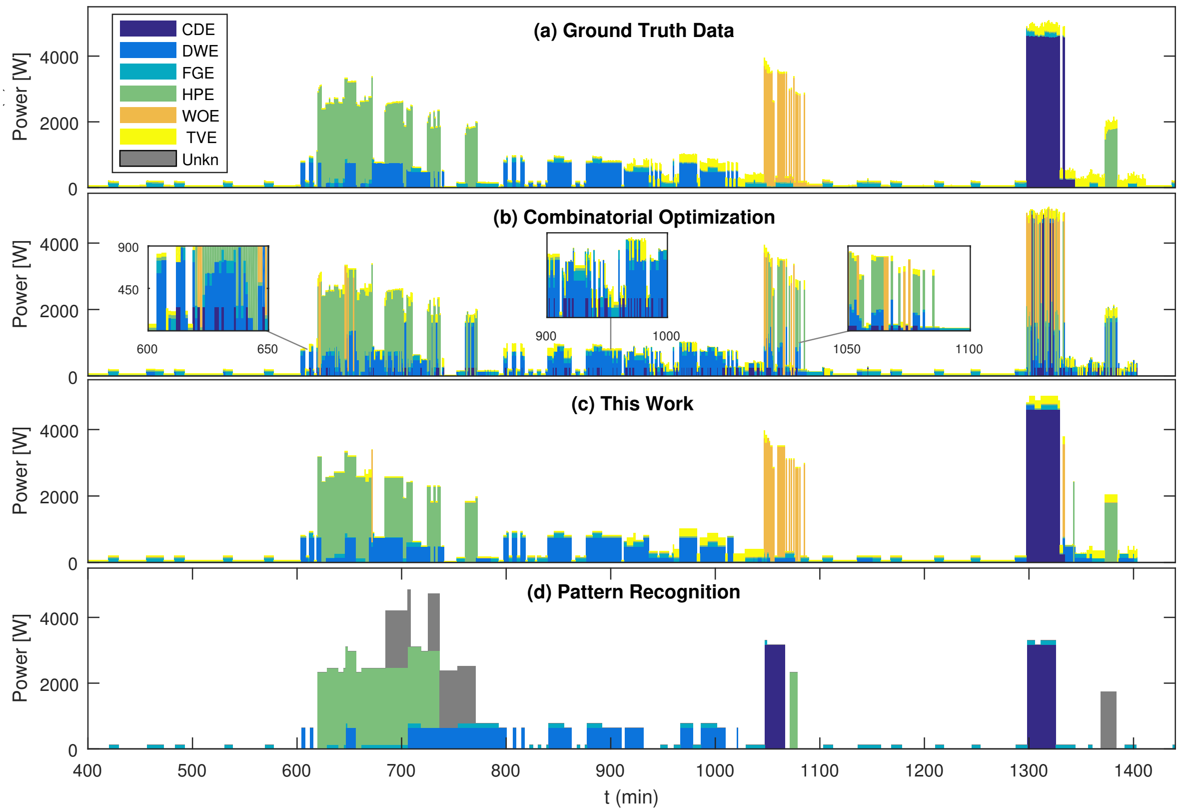


Figure 4.3: Results and comparison of the proposed model with the other two techniques when considering only the active power measurements

In the graphs, (a) contains the ground truth data with the expected results. (b) has the CO method implemented using the Algorithm 1. (c) has the results using the model of this work when using the Algorithm 2. Finally (d) compares with Hart's pattern recognition method. Both graphs are going to be discussed in details in the next subsections:

Using only active power measurements

The Fig. 4.3.b shows the disaggregation results applying only the classic CO from Algorithm 1. The highlighted zooms show a series of unrealistic activation of different loads, which is an example of the MS problem discussed in the introduction section. Fig. 4.3.c shows the disaggregation results using the full proposed model in the Algorithm 2. The minimum time constraints (3.16)-(3.18) help the model to avoid multiple load switching. Hart's algorithm

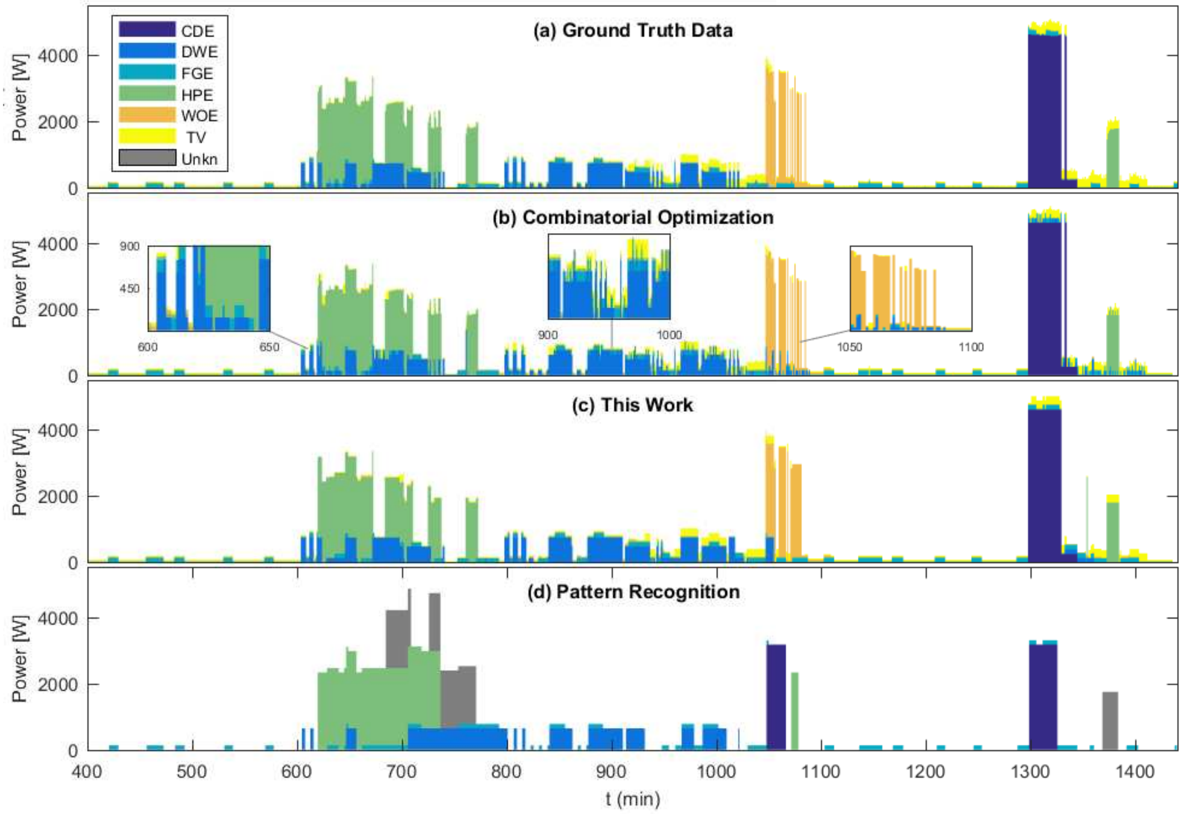


Figure 4.4: Comparison of the real measurements (ground truth) with the proposed model and two other methods using both active and reactive power measurements

Fig. 4.3.d shows the disaggregation results for Hart's method. As Hart's implementation in NILMTK is unsupervised, four out of the six appliances were identified. One unknown load was also identified, which seems to be associated with the heat pump (HPE). As an example of edge dependency, the turn off event of the HPE was not detected at $t = 672$. Hence, when the load was turned on again at $t = 684$, the pattern recognition algorithm assigned the event to a different (unknown) load. The same problem also happens in the next few events and are assigned to the unknown load.

Using both active and reactive power measurements

The Figure 4.4.b shows that the CO results are expected to improve when using more than one kind of measurement. For example the CDE is better identified when comparing with the Figure 4.3.b. No big visual change is observed in the proposed model in the Figure 4.4.c when compared with the Figure 4.3.c. However, as it will be observed in the numerical results, the identification metric of the proposed method slightly improved for all the appliances. Finally, Hart's method shown in the Figure 4.3.d did not change since the technique was implemented using the active and reactive power in both cases.

4.3.3 Numerical Results

The previously presented metrics ETE and EAP are applied to the test case. The Table 4.3 presents the numerical results when considering only the active power. The Table 4.4 presents the results when also considering the reactive power.

In the Table 4.3, the lowest EAP for each appliance is highlighted. The average of the EAP error for the CO method is about 88%. The proposed model has a lower EAP for all loads. The average of the proposed method is 20.25%. Regarding Hart's method, all EAP values are over 50%. The ETE error is lower than the other two methods; however, the ETE metric considers the energy consumption of the overall time period.

In the Table 4.4, all the values in EAP were improved in the proposed model when comparing to the Table 4.3. Thus, the addition of new features, when available, helps on improving the method's accuracy. The average of ETE for the proposed model decreased from 8.62% to 7%. In addition, the average of EAP for the proposed model decreased from 20.25% to 15.5%. The results for CO have also highly improved from 22.3% to 6.61% (ETE), and from 88.41% to 20.28% (EAP), respectively. For pattern recognition, the same results are presented since the original method was implemented with the reactive power. The appliances CDE and WOE performed better for the CO than for the proposed model. The WOE's EAP is 0.5% for CO versus 6.1% for the proposed model. In addition, WOE's EAP is 9.5% versus 12.2%. The proposed method performed slightly worse since the model of those two appliances is approximated. They are formed by states that continuously turn ON and OFF, and the model approximates them to a minimum time.

Table 4.3: Error for the Supervised Case 1 with the Active Power Only

	CO (%)		This Work (%)		Patt. Rec. (%)	
	ETE	EAP	ETE	EAP	ETE	EAP
CDE	50.8	98.2	7	**7.3**	8.7	84.1
DWE	39.1	89	16.8	**29**	5.6	74.1
FGE	7.1	70.7	3.6	**43.9**	3.6	53.3
HPE	6.2	57	1.1	**5.2**	0.1	65.9
WOE	17.7	160.1	5.5	**15.4**	-	-
TVE	12.9	55.5	17.7	**20.7**	-	-
Average	22.3	88.41	8.62	**20.25**	4.5	69.35

When comparing both Table 4.3 and the Table 4.4, it is possible to note that the proposed model performed well even in a scenario with limited data (only the active power). The EAT metric is directly correlated with the MS problem. CO significantly improved when including the reactive power. As shown in the Table 4.3, when using only the active power, the proposed

Table 4.4: Error for the Supervised Case 1 Including the Reactive Power

	CO (%)		This Work (%)		Patt. Rec. (%)	
	ETE	EAP	ETE	EAP	ETE	EAP
CDE	0.2	**0.5**	5.7	6.1	8.7	84.1
DWE	11.5	28.1	14	**23.2**	5.6	74.1
FGE	3.9	43.5	0.3	**28**	3.6	53.3
HPE	1.3	5.6	0.7	**4**	0.1	65.9
WOE	6.7	**9.5**	2.2	12.2	-	-
TVE	16.1	34.5	19.3	**19.7**	-	-
Average	6.61	20.28	7	**15.5**	4.5	69.35

model improved the CO method and decreased 60% of the EAP error (80.41% to 20.25%).

4.4 Supervised Case 2

4.4.1 Experimental Setup

The second test case considers a longer period of one week rather than one day. Hence, the total number of samples is 10080 measurements (one per minute). In addition, the test case 2 considers seven appliances: Basement Plugs and Lights (BME); Dryer (CDE); Dishwasher (DWE); Fridge (FGE); Forced Air Furnace (FRE); Heat Pump (HPE); and Television/Entertainment (TVE). All the appliances are represented by 13 power states. Table 4.5 shows the data used as input for the model. For this test scenario, both the active and reactive power are considered. The chosen window's length has 40 measurements.

Table 4.5: Input Data for the Experimental Setup in Test Case C

appl	**i**	D_i	**prev**$_i$	P_i (W)	Q_i **(VAr)**	MD_i
BME	1	1	0	333	26	15
BME	2	1	0	407	26	5
CDE	3	2	0	4569	412	25
CDE	4	2	3	247	407	5
DWE	5	3	0	751	34	5
DWE	6	3	0	478	0	15
FGE	7	4	0	129	8	7
FRE	8	5	0	105	26	20
HPE	9	6	0	37	17	30
HPE	10	6	0	1807	324	10
HPE	11	6	0	2435	429	30
TVE	12	7	0	38	13	30
TVE	13	7	0	239	31	30

4.4.2 Graphical Results

The graphic in the Figure 4.5 shows a sub split of 1000 minutes for comparing the proposed technique with the other two algorithms. The CDE is identified by the proposed technique, however, with a different shape when compared with the ground truth data. This is due to the minimum time approximation of the model of the CDE appliance. The HPE appliance is visually very similar to the ground truth. The CO method is continually activating the BME appliance in instants that the DWE was expected to be active, which does not occur in the proposed method. Hart's method identified the FGE, the HPE, and the CDE appliances. At the instant between the sample 7500 and 7600, it is observed the activation of an unknown appliance. This unknown appliance was activated due to the detection of a false event. The HPE turned off and the CDE turned on in the next sample which created a false step of about 2000W. This confusion is not made by the proposed technique since it does not depend on event detection.

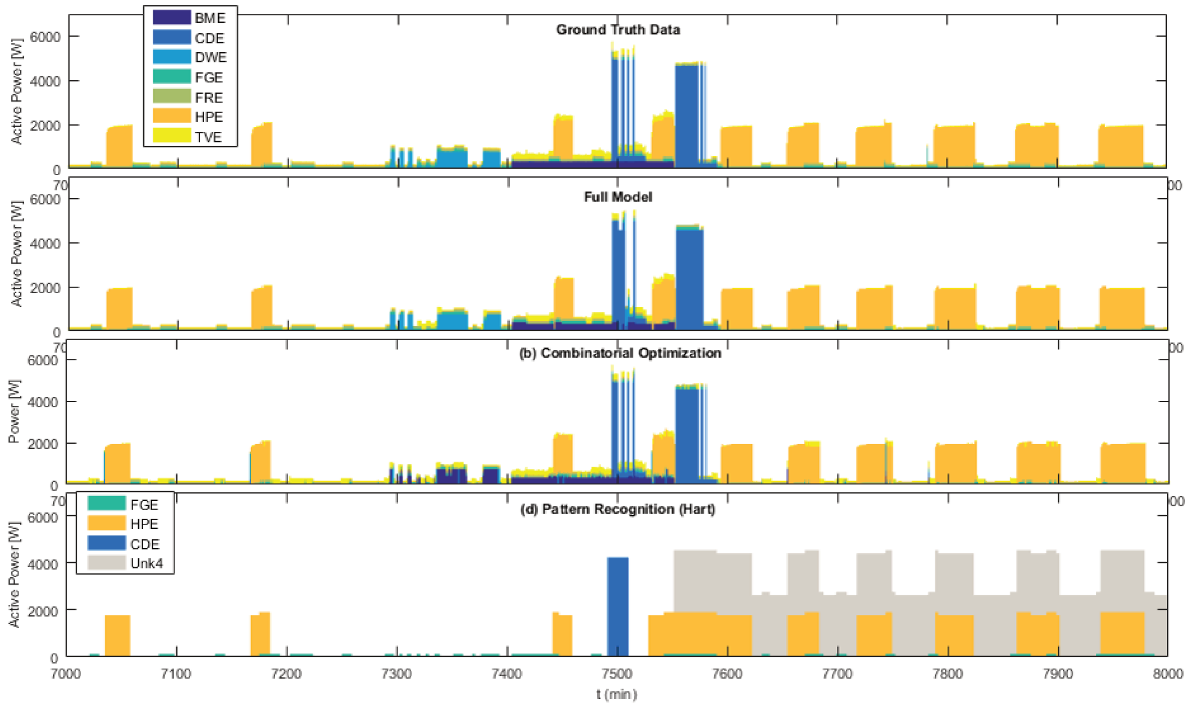


Figure 4.5: Comparison of the real measurements (ground truth) with the proposed model and two other methods

4.4.3 Numerical Results

Table 4.6 shows the numerical results. When comparing with CO, the proposed model got a lower EAP error for 6 out of the 7 appliances. The EAP for CDE was 3.3% for the CO while it was 12.2% for this work. The proposed model got a higher EAP error for the CDE since the original appliance is made of multiple ON/OFF which are approximated by the proposed technique (refer to the region near to the sample 7500 in the Figure 4.5). The highest error for the proposed technique was for the DWE appliance, however, it was still a better result when compared with the CO and it was not identified by the pattern recognition technique. Regarding the energy prediction, the proposed technique got an ETE error lower than 10% for most of the appliances, except the FRE which is 10.8%. The average of the ETE error was 5.86% which is significantly lower than the two other techniques (19.15% and 20.6%). This means that the proposed technique would perform a better estimation of energy consumption when compared with the two other techniques.

Table 4.6: ETE and EAP metric for the supervised case 2.

	CO (%)		This Work (%)		Patt. Rec. (%)	
	ETE	EAP	ETE	EAP	ETE	EAP
BME	2.7	35.4	5.7	**22.2**	-	-
CDE	0.6	**3.3**	4.7	12.2	42.8	52.2
DWE	8.7	146.5	6.4	**44.9**	-	-
FGE	44.2	81.1	7.6	**38.2**	15.4	64.1
FRE	26.6	26.8	10.8	**11.1**	-	-
HPE	2.7	6.7	2.7	**4**	3.6	16.1
TVE	48.6	93.7	3.1	**4.2**	-	-
Average	19.15	56.21	5.86	**19.54**	20.6	44.1

4.5 Summary and Analysis of Results

This chapter presented two experiments in order to evaluate the performance of the NILM method. Supervised scenarios were considered in order to validate the optimization model. The public data set AMPds was used for validation. As discussed in [49], a good NILM algorithm should address the following six requirements: use typical meter features, a minimal accuracy of 80%, no training, real-time capability, scalability, and flexibility.

The presented NILM algorithm accomplishes with most of the former requirements, since: uses only low frequency features; accomplishes the minimum 80% requirement (considering the complement of ETE's average); the parameters in the input table make it suitable for a no training (unsupervised) setting; the results could be shown in real-time to the user at every m measurements; scalability will depend on the way in which the input table is constructed and how it is updated; finally multiple appliances types are handled (multi-state, ON/OFF, constant) thanks to the parameters MD_i and $prev_i$.

Chapter 5

Conclusion

In this dissertation, a NILM method based on MILP optimization was proposed. The classic CO model was expanded for increasing the accuracy and performance. As the main contribution, a new set of linear constraints were proposed to efficiently model the behavior of appliances. The proposed model enhanced the classic CO model in order to avoid the MS problem. Also, a window-based algorithm is proposed in order to improve the computational complexity. The proposed algorithm surpassed the test cases for the identification of appliances. As the main advantage, the algorithm does not require data in high resolution, hence low-cost smart meters are sufficient to deploy it. Results were accomplished with a one-minute reading resolution. The inclusion of other features besides the active power is optional and help to improve the accuracy.

As main challenges, this algorithm demonstrated to work well in a supervised scenario. However, as shown in the appendix, there's work to be done in unsupervised cases and how to deal with unknown loads. As proposed in the appendix, one possible approach is to apply an preprocessing set to learn all the possible steps. Another challenge to be addressed in this algorithm is the computational complexity. Although the use of windows is helpful to limit the complexity growth in the time dimension, the algorithm is still susceptible to scenarios with too many states. In tests cases, the maximum number of states was about 20 states. One possible solution in this context is to apply many models in parallel where each one would be specialized in identifying one specific appliance.

5.1 Future Work

The following list shows points that might be considered for future work in this area:

- Investigate preprocessing techniques for extracting the input parameters from the aggregated data (unsupervised approach);
- Study modifications of the presented model in order to allocate unknown loads without the need of a preprocessing step;
- Propose an automatic feature extraction and disaggregation based on the methodologies presented in the appendix;
- Implement semi-supervised models in which appliance models are learned from edge detection of the aggregated data and deployed to the input parameters;
- Use different performance metrics and datasets such as the Reference Energy Disaggregation Data Set (REDD);
- Implement probabilistic functions: for example, penalization for turning on on unlikely times (like a washing machine at 3 AM);
- Implement the concept of an occupied and non-occupied type of loads.

Bibliography

[1] Scott Krueger. *T&D World Magazine: Smart Electricity Meters to Total 780 Million in 2020, Driven by China's Roll-out*. 2015. URL: http://tdworld.com/ami/smart-electricity-meters-total-780-million-2020-driven-china%5C%2Ds-roll-out.

[2] US Energy Information Administration. *How many smart meters are installed in the United States, and who has them?* 2016. URL: http://www.eia.gov/tools/faqs/faq.cfm?id=108&t=3.

[3] Sarah Darby. *The effectiveness of energy feedback on energy consumption*. Tech. rep. Environmental change institute, Apr. 2006.

[4] Electric Power Research Institute. *Residential Electricity Use Feedback: A Research Synthesis and Economic Framework*. Tech. rep. Feb. 2009.

[5] K. Carrie Armel et al. "Is disaggregation the holy grail of energy efficiency? The case of electricity". In: *Energy Policy* 52 (Jan. 2013), pp. 213–234.

[6] Benjamin Nicole, Kien Trung Nguyen, and Illia Azarkh. *How can NIALM help to perform predictive and curative maintenance?* 3rd European Workshop Non-Intrusive Load Monitor. Oct. 2016. URL: http://www.nilm.eu/nilm-workshop-2016/.

[7] Bérénice Huquet. *EDF Position Paper on NILM*. 3rd European Workshop Non-Intrusive Load Monitor. Oct. 2016. URL: http://www.nilm.eu/nilm-workshop-2016/.

[8] Aapo Hyvarinen, Juha Karhunen, and Erkki Oja. *Independent Component Analysis*. 1st. New York, NY: Wiley-Interscience, 2001.

[9] G.W. Hart. "Prototype Nonintrusive Appliance Load Monitor". In: *MIT Energy Laboratory Technical Report and Electric Power Research Institute Technical Report* (1985).

[10] G. W. Hart. "Nonintrusive Appliance Load Monitoring". In: *Proceedings of the IEEE* 80.12 (Dec. 1992), pp. 1870–1891.

[11] Oliver Parson. *Overview NILM field*. 2015. URL: `http://blog.oliverparson.co.uk/2015/03/overview-of-nilm-field.html`.

[12] Stephen Makonin. "Investigating the switch continuity principle assumed in Non-Intrusive Load Monitoring (NILM)". In: *2016 IEEE Canadian Conf. Elect. Comput. Eng.* Vancouver, BC, May 2016, pp. 1–4.

[13] M. Zeifman and K. Roth. "Nonintrusive appliance load monitoring: Review and outlook". In: *IEEE Trans. Consum. Electron.* 57.1 (Feb. 2011), pp. 76–84.

[14] Marisa B. Figueiredo, Ana de Almeida, and Bernardete Ribeiro. "An Experimental Study on Electrical Signature Identification of Non-Intrusive Load Monitoring (NILM) Systems". In: *Int. Conf. Adaptive and Natural Computing Algorithms (ICANNGA 2011)*. Berlin, Germany, Apr. 2011, pp. 31–40.

[15] Mario Berges et al. "Learning Systems for Electric Consumption of Buildings". In: *Proc. 2009 ASCE Int. Workshop Computing Civil Eng.* Austin, TX, June 2009, pp. 1–10.

[16] Jon Froehlich et al. "Disaggregated End-Use Energy Sensing for the Smart Grid". In: *IEEE Pervasive Computing* 10.1 (2010), pp. 28–39.

[17] Y. H. Lin, M. S. Tsai, and C. S. Chen. "Applications of fuzzy classification with fuzzy c-means clustering and optimization strategies for load identification in NILM systems". In: *2011 IEEE Int. Conf. Fuzzy Systems (FUZZ-IEEE 2011)*. Taipei, Taiwan, June 2011, pp. 859–866.

[18] P. Ducange, F. Marcelloni, and M. Antonelli. "A Novel Approach Based on Finite-State Machines with Fuzzy Transitions for Nonintrusive Home Appliance Monitoring". In: *IEEE Trans. Ind. Informat.* 10.2 (May 2014), pp. 1185–1197.

[19] M. Nguyen et al. "A novel feature extraction and classification algorithm based on power components using single-point monitoring for NILM". In: *2015 IEEE 28th Canadian Conf. Elect. Comput. Eng. (CCECE)*. Halifax, NS, May 2015, pp. 37–40.

[20] J. Gillis and W. G. Morsi. "Non-intrusive load monitoring using orthogonal wavelet analysis". In: *2016 IEEE Canadian Conf. Elect. Comput. Eng. (CCECE)*. Vancouver, BC, May 2016, pp. 1–5.

[21] C. Duarte et al. "Non-intrusive load monitoring based on switching voltage transients and wavelet transforms". In: *2012 Future of Instrumentation Int. Workshop (FIIW) Proc.* Gatlinburg, TN, Oct. 2012, pp. 1–4.

[22] A. Zoha et al. “Acoustic and device feature fusion for load recognition”. In: *2012 6th IEEE Int. Conf. Intelligent Systems*. Sofia, Bulgaria, Sept. 2012, pp. 386–392.

[23] C. Zhou, S. Liu, and P. Liu. “Neural Network Pattern Recognition Based Non-intrusive Load Monitoring for a Residential Energy Management System”. In: *2016 3rd Int. Conf. Inform. Sci. Control Eng. (ICISCE)*. Beijing, China, July 2016, pp. 483–487.

[24] S. Biansoongnern and B. Plangklang. “Nonintrusive load monitoring (NILM) using an Artificial Neural Network in embedded system with low sampling rate”. In: *2016 13th Int. Conf. Elect. Eng./Electron., Comput., Telecommun. Inform. Technology (ECTI-CON)*. Chiang Mai, Thailand, June 2016, pp. 1–4.

[25] L. Mauch and B. Yang. “A novel DNN-HMM-based approach for extracting single loads from aggregate power signals”. In: *2016 IEEE Int. Conf. Acoust., Speech Signal Processing (ICASSP)*. Shanghai, China, Mar. 2016, pp. 2384–2388.

[26] Jack Kelly and William Knottenbelt. “Neural NILM: Deep Neural Networks Applied to Energy Disaggregation”. In: *2nd ACM Int. Conf. Embedded Systems Energy-Efficient Built Environments*. Seoul, South Korea, Nov. 2015.

[27] J. Z. Kolter and Tommi Jaakkola. “Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation”. In: *J. Machine Learning Research Proc. Track* 22 (Apr. 2012), pp. 1472–1482.

[28] J. Zico Kolter and Matthew J. Johnson. “REDD: A public data set for energy disaggregation research”. In: *Proc. Workshop Data Mining Applicat. Sustainability (SustKDD)*. San Diego, CA, Aug. 2011, pp. 1–6.

[29] Hyungsul Kim et al. “Unsupervised Disaggregation of Low Frequency Power Measurements”. In: *Proc. 2011 SIAM Int. Conf. Data Mining*. Mesa, AZ, Apr. 2011, pp. 747–758.

[30] S. Makonin et al. “Exploiting HMM Sparsity to Perform Online Real-Time Nonintrusive Load Monitoring”. In: *IEEE Trans. Smart Grid* 7.6 (Nov. 2016), pp. 2575–2585.

[31] J. Zico Kolter, Siddarth Batra, and Andrew Y. Ng. “Energy Disaggregation via Discriminative Sparse Coding”. In: *Proceedings of the 23rd International Conference on Neural Information Processing Systems*. Vancouver, BC: Curran Associates Inc., 2010, pp. 1153–1161.

[32] A. Rahimpour et al. “Non-Intrusive Energy Disaggregation Using Non-negative Matrix Factorization with Sum-to-k Constraint”. In: *IEEE Trans. Power Syst.* (2017). to be published.

[33] Dominik Egarter and Wilfried Elmenreich. "Load disaggregation with metaheuristic optimization". In: *4th D-A-CH Energieinformatik Conf.* Klagenfurt, Austria, Nov. 2015.

[34] K. Suzuki et al. "Nonintrusive Appliance Load Monitoring Based on Integer Programming". In: *Proc. SICE Ann. Conf., (The University Electro-Commun.)* Tokyo, Japan, Aug. 2008, pp. 2742–2747.

[35] M. Z. Ali Bhotto, S. Makonin, and I. Bajic. "Load Disaggregation Based on Aided Linear Integer Programming". In: *IEEE Trans. Circuits Syst. II, Exp. Briefs* (2016). to be published.

[36] Smart Energy GB. *Smart meters explained*. Accessed: 2017-08-20. URL: https://www.smartenergygb.org/en/about-smart-meters/what-is-a-smart-meter.

[37] David Kathan. *Demand Response and Advanced Metering*. Tech. rep. Washington, D.C., United States: Federal Energy Regulatory Commission, Dec. 2008, p. 65.

[38] Berg Insight. *Smart Metering in Europe*. Tech. rep. East Lansing, Michigan: Berg Insight, 2017, p. 260.

[39] Stephen William Makonin. "Real-Time Embedded Low-Frequency Load Disaggregation". PhD thesis. Simon Fraser University, Aug. 2014.

[40] G. W. Hart and A. T. Bouloutas. "Correcting dependent errors in sequences generated by finite-state processes". In: *IEEE Transactions on Information Theory* 39.4 (July 1993), pp. 1249–1260. ISSN: 0018-9448. DOI: 10.1109/18.243442.

[41] Robert Fourer, David M. Gay, and Brian W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. 2nd. Pacific Grove, CA, USA: Brooks/Cole-Thomson Learning, 2003.

[42] H.P. Williams. *Model Building in Mathematical Programming*. Wiley, 2013. ISBN: 9781118506189. URL: https://books.google.com.br/books?id=YJRh0tOes7UC.

[43] IBM ILOG CPLEX V12.1. *User's Manual for CPLEX*. IBM Corp. 2009.

[44] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004. ISBN: 0521833787.

[45] M. Carrion and J. M. Arroyo. "A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem". In: *IEEE Trans. Power Syst.* 21.3 (Aug. 2006), pp. 1371–1378.

[46] Frank Neumann and Carsten Witt. "Combinatorial Optimization and Computational Complexity". In: *Bioinspired Computation in Combinatorial Optimization: Algorithms and Their Computational Complexity*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 9–19. ISBN: 978-3-642-16544-3. DOI: 10.1007/978-3-642-16544-3_2. URL: https://doi.org/10.1007/978-3-642-16544-3_2.

[47] Nipun Batra, Jack Kelly, Oliver Parson, et al. "NILMTK: An Open Source Toolkit for Non-intrusive Load Monitoring". In: *5th Int. Conf. Future Energy Systems (ACM e-Energy)*. Cambridge, UK, June 2014.

[48] Stephen Makonin et al. "Electricity, water, and natural gas consumption of a residential house in Canada from 2012 to 2014". In: *Scientific Data* 3 (June 2016), pp. 1–12.

[49] M. Zeifman. "Disaggregation of home energy display data using probabilistic approach". In: *IEEE Trans. Consum. Electron.* 58.1 (Feb. 2012), pp. 23–31.

[50] David Arthur and Sergei Vassilvitskii. "K-means++: The Advantages of Careful Seeding". In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '07. New Orleans, Louisiana: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035. ISBN: 978-0-898716-24-5. URL: http://dl.acm.org/citation.cfm?id=1283383.1283494.

[51] David M. Blei and Michael I. Jordan. "Variational inference for Dirichlet process mixtures". In: *Bayesian Anal.* 1.1 (Mar. 2006), pp. 121–143. DOI: 10.1214/06-BA104. URL: http://dx.doi.org/10.1214/06-BA104.

[52] Ahmed Zoha et al. "Non-Intrusive Load Monitoring Approaches for Disaggregated Energy Sensing: A Survey". In: *Sensors* 12.12 (2012), pp. 16838–16866.

[53] The OpenEnergyMonitor Project. *EmonPi Hardware Technical Documentation*. Accessed: 2017-08-25. URL: https://wiki.openenergymonitor.org/index.php/EmonPi.

[54] The OpenEnergyMonitor Project. *About the OpenEnergyMonitor Project*. Accessed: 2017-08-25. URL: https://openenergymonitor.org/?q=about.

Appendix A

Full Proposed Mathematical Model

The full proposed model in details for a window based case is presented in the next page of this Appendix. The algorithm was implemented using the mathematical programming language AMPL with the very same equations.

repeat

let $\tilde{T} = \{T_0, \cdots, T_0 + m\}$

solve

$$\min_{x_i(t)} \quad \sum_{t \in \tilde{T}} \delta_P(t) + \delta_Q(t)$$

s. t.

$$P(t) - \sum_{i=1}^{n} x_i(t)\, P_i \ \leq \ \delta_P(t) \tag{A.1}$$

$$P(t) - \sum_{i=1}^{n} x_i(t)\, P_i \ \geq \ -\delta_P(t) \tag{A.2}$$

$$Q(t) - \sum_{i=1}^{n} x_i(t)\, Q_i \ \leq \ \delta_Q(t) \tag{A.3}$$

$$Q(t) - \sum_{i=1}^{n} x_i(t)\, Q_i \ \geq \ -\delta_Q(t) \tag{A.4}$$

$$x_i(t) = 0 \quad \forall t \in \tilde{T}, i \in S \mid P(t) \leq TH \tag{A.5}$$

$$x_i(t) - x_i(t-1) = up_i(t) - dw_i(t) \quad \forall i \in S, t > T_0 \tag{A.6}$$

$$x_i(T_0) - X_i = up_i(t) - dw_i(t) \quad \forall i \in S, t = T_0 \tag{A.7}$$

$$up_i(t) + dw_i(t) \leq 1 \quad \forall i \in S, t \in \tilde{T} \tag{A.8}$$

$$up_i(t) = dw_{\text{prev}_i}(t) \quad \forall i \in S, t \in \tilde{T} \mid \text{prev}_i > 0 \tag{A.9}$$

$$\sum_{k=T_0}^{G_i} [1 - x_i(k)] = 0 \quad \forall i \in S \tag{A.10}$$

$$\sum_{k=t}^{t+MD_i-1} x_i(k) \geq MD_i\,[x_i(t) - x_i(t-1)]$$
$$\forall i \in S, t \in G_i + T_0 \ldots T_f - MD_i + 1 \tag{A.11}$$

$$\sum_{k=t}^{T_f} \{x_i(k) - [x_i(t) - x_i(t-1)]\} \geq 0$$
$$\forall i \in S, t \in T_f - MD_i + 2 \ldots T_f \tag{A.12}$$

let $T_0 = T_0 + m + 1$

until $T_0 > T_f$;

Algorithm 3: Full proposed model in details.

Appendix B

Extracting Input Parameters

The math model presented in Chapter 3 requires as input the power level, the minimum running time and sequence of states of each load. The extraction of those input parameter may be either with a visual inspection or automated. The visual inspection is useful in order to validate the model. However, automatic extraction becomes necessary when applying the model to a real world scenario. This appendix presents a strategy for extracting these input parameters both manually and automated.

B.1 Supervised Setting

In the supervised setting, the challenge is to extract the main operational level, minimum time and sequence of states from appliance measurements. The flowchart in the Figure B.1 illustrates how a NILM optimization algorithm would receive the input parameters. Given a set of measurements of an appliance (such as the active and the reactive power), feature extraction is performed in order to acquire the input parameters to the optimization algorithm. The feature extraction process can be split into three steps:

1. Extract the main states from clustering algorithms;
2. Extract the minimum time from histograms of continuously active states;
3. Extract sequence of states from table of combinations of states

Further details of those items are presented in the next subsections.

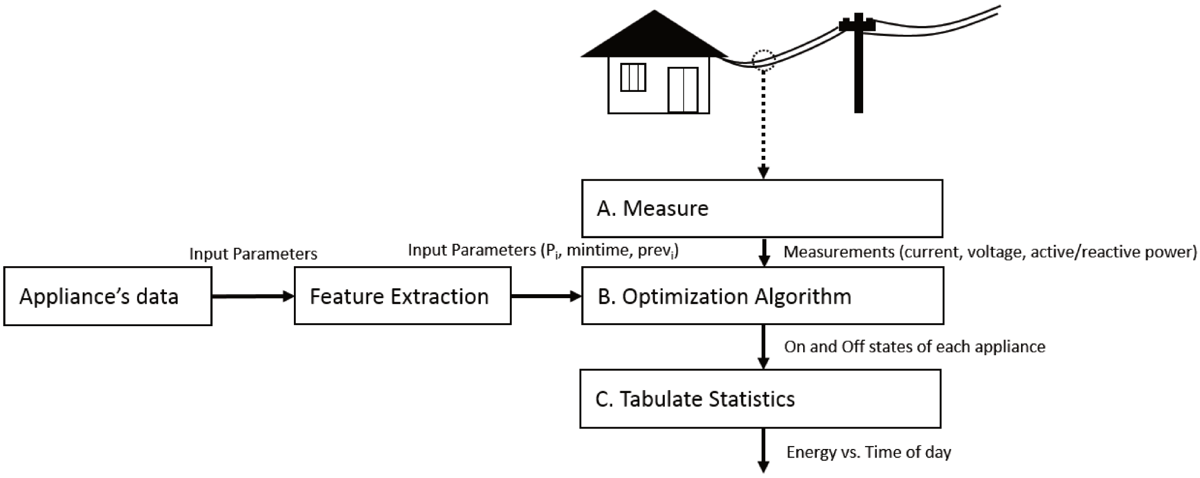


Figure B.1: Flowchart of a supervised NILM algorithm using optimization

B.1.1 Main States

The most representative operating states of a given appliance can be extracted either manually with visual plots or automatically by using clustering algorithms.

Manual Extraction

In the manual extraction, histograms can be used when only one measurement is available or scatter plots for two or more. When only one measurement is available, the histogram plots will display the most relevant states of the given appliance. The Figure B.2, illustrates how the histogram plot of a hypothetical appliance might look like. The appliance of the Figure B.2 is composed of three main active power states A, B, and C. If those states are well defined, a histogram plot will show three normal distributions around A, B, and C.

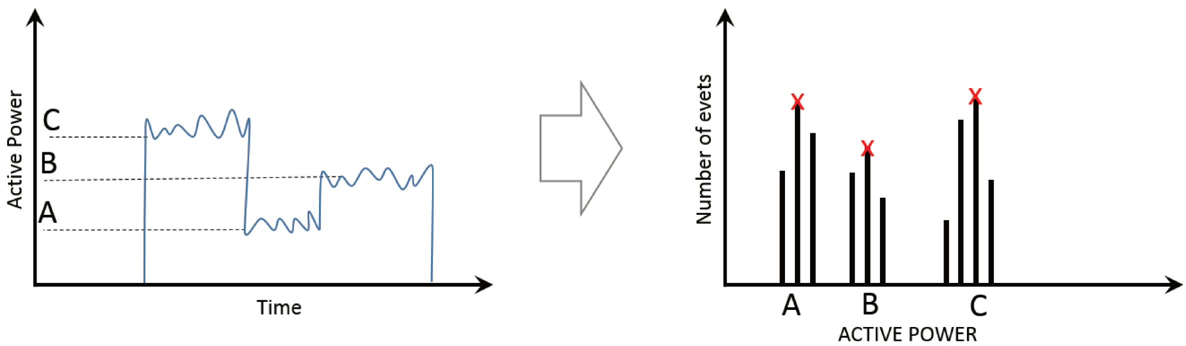


Figure B.2: Visualization of the main states with one measurement

When more than one measurement is available, the most relevant operating states may be visualized using scatter plots. For example, the Figure B.3 shows the scatter plot of an appliance with two measurements (active and reactive power). A plot of the active power vs the reactive power will appear as a set of measurements around the main states. The median of each cluster can be taken as the main representative state.

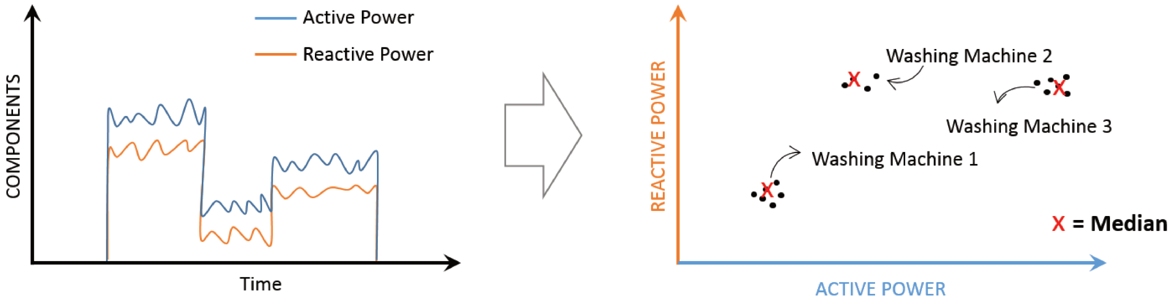


Figure B.3: Visualization of the main states with two measurement

Automated Extraction

The extraction process can also be automated by using clustering algorithms. There are many ready implementations of clustering algorithms in the major data science tools such as MATLAB, Python (Sklearn) and R. Those algorithms range from simple methods such as K-means [50] up to more advanced methods such as Bayesian Gaussian Mixtures [51]. A clustering algorithm can be directly applied to a set of measurements independently of its dimension. Some clustering algorithms require as inputs the number of clusters to be identified (K-means, for example). Some others try to automatically define the number of clusters. For example, the Bayesian Gaussian Mixtures has as an input parameter the maximum number of components to be identified. The Figure B.4 shows an example of an application of the Bayesian Gaussian Mixtures algorithm in four appliances. On the left, there's a plot of four appliances (power versus time). The active power is in red and the reactive power is in blue. On the right of the Figure B.4, those same appliances are plotted with the active power in the x-axis and the reactive power in the y-axis. The green, red and blue dots are the clusters identified by the clustering algorithm. The algorithm also returns the centroid of each cluster, which can be used as main states for the optimization algorithm of Chapter 3. The top left scatter plot identified only two clusters. The clustering algorithm works well only when the clusters are well defined. The button right plot shows that the algorithm identified two clusters in a suspicious region that could be considered only one. It is worth noting that the automatic extraction of main power states was not evaluated against other feature extraction strategies since this is not the focus of this dissertation. A review of alternative feature extraction strategies may be found in [52].

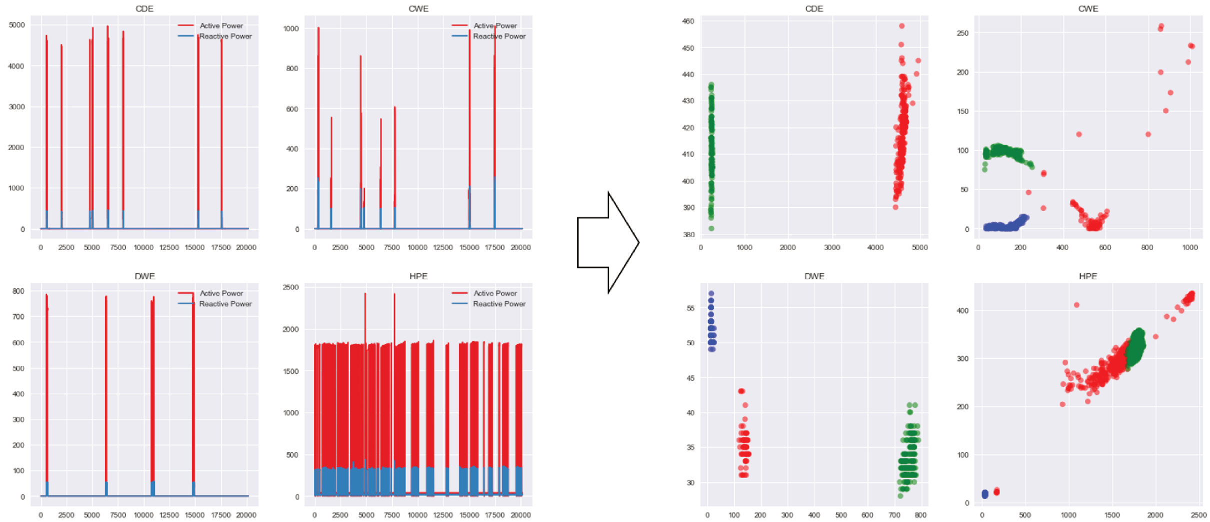


Figure B.4: Example of a clustering algorithm applied to find the main states of four appliances

B.1.2 Minimum Time

The minimum time may also be extracted by either visual inspection or automated. The visual inspection of plots of time versus power state might help in giving an idea of the minimum time parameter. In order to automate this process, it is possible to approximate me measurements of an appliance to their main states, which are found in the steps described in the previous subsection. Next, a set of activated time can be created for each power state. For example, consider a hypothetical appliance with state $x(t) = [0011100011000111110]$. The activated time in $x(t)$ means that the appliance was first activated for 3 samples, later it was activated for 2 samples and finally, it was activated for 4 samples. A set with the continuously activated time for this hypothetical appliance would be $[3, 2, 4]$. It is possible to create a histogram with the set of continuously activated time. The Figure B.5 illustrates a histogram of activated time created for a cyclic appliance. The minimum time can be extracted in the left tail region of the histogram. Percentiles are a useful measurement for extracting the minimum time. A percentile indicates the value below a given percentage of observations in which a group of observations falls. For example, a percentile the 10th percentile is the value (or score) below which 10% of the observations may be found.

The Figure B.6 shows an example of extraction of the minimum time for a heat pump. One of the main states of the appliance is 339W. A set of activated time for the state 339W is extracted and a histogram is created. The right in the Figure B.6 shows a histogram. The first quartile of the histogram (25th percentile) is 5.25 minutes, which means that the minimum time of 75% of activations was higher than 5.25 minutes. This value could be used as an input parameter for the model. It is worth noting that this example considers a small time scenario

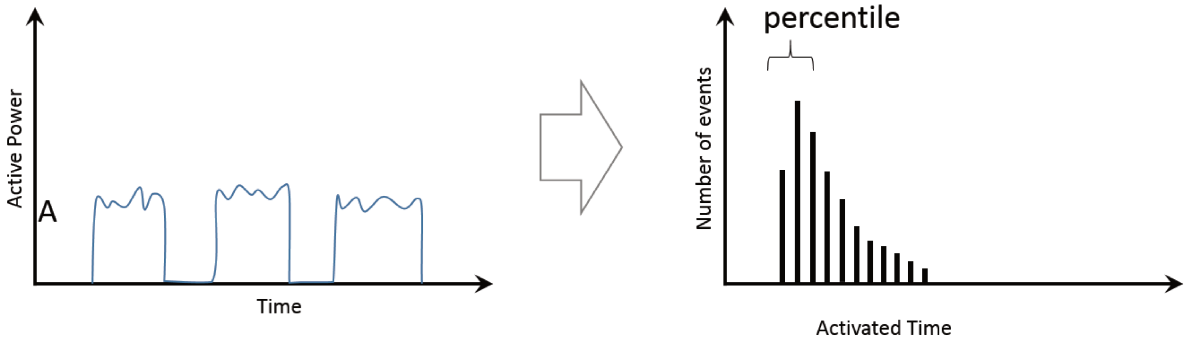


Figure B.5: Illustration of extraction of the minimum time

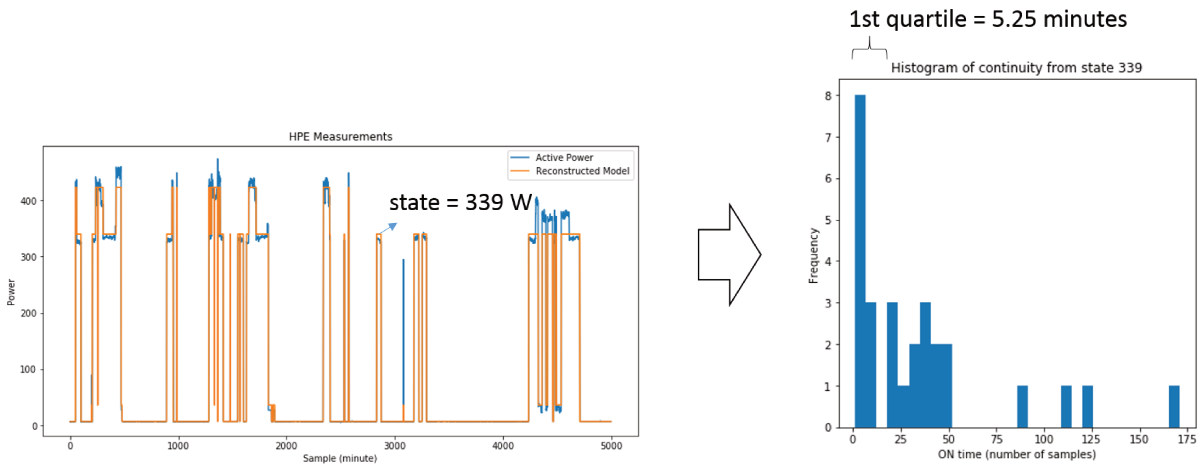


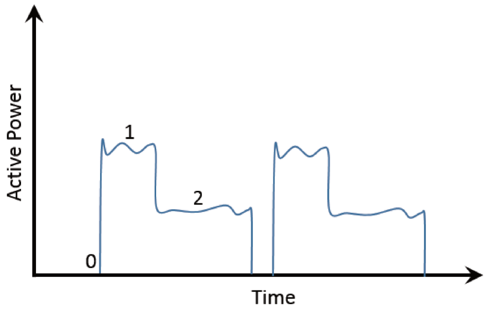
Figure B.6: Example of minimum time extraction for a heat pump

and a better minimum time extraction could be achieved with a larger set of measurements.

B.1.3 Sequence of States

Finally, the extraction of the sequence of states may also be inferred by visual inspection or automated. In the visual inspection, we simply check in the measurement's plots those transition of states which always occur together. The automation extraction can be performed by creating a model with the approximated states presented in the first subsection. Then, the transition of states can be tabulated.

For example, the Figure B.7 presents a hypothetical cyclic appliance with two operating states: state 1 and state 2. A third state 0 is also considered for instants where the appliance is OFF. Next, the sequence of those states can be tabulated in the way presented on the right side of the Figure B.7. When the frequency of a transition between states is very high, this transition can be used as input parameter for the model. For example, in the Figure B.7, the transition of the state 0 (OFF) to the state 1 occurred 90% of the times. Hence this transition could be used



From → To	Frequency
0 → 1	90%
0 → 2	10%
1 → 0	50%
1 → 2	50%
2 → 0	5%
2 → 1	95%

Figure B.7: Illustration of extraction of minimum states

as input parameter for the model.

B.2 Summary

A strategy for extracting the input parameters was presented. This extraction can be performed either using visual inspection or automatically. Clustering algorithms were proposed to automate the extraction of power states. The cluster's centroids can be used as input parameter. Next, those main states can be associated with continuously activated times in order to extract the minimum time. Finally, a table with the transition of those states can be created in order to extract the transition of states.

Appendix C

Unsupervised Experiments

Experiments were also performed considering that the measurements of each appliance are not available. In this scenario, a preprocessing step is necessary in order to extract the input parameters from the aggregated data. The Figure C.2 illustrates the flowchart of an unsupervised NILM process when using optimization. A preprocessing strategy is described in the next section along with two test cases.

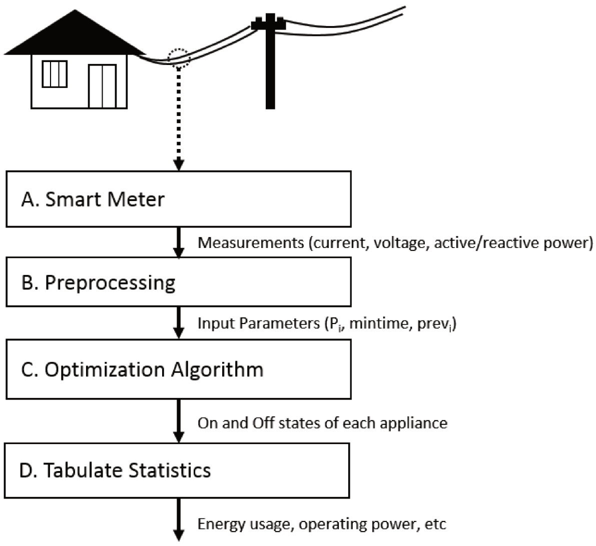


Figure C.1: Flowchart of an unsupervised optimization algorithm

C.1 Preprocessing Algorithm

Although it is not the goal of this work to propose an optimal preprocessing unsupervised method, some basic and simple strategies are explored. The challenge here is to extract the main operational state, the minimum time and the sequence of states from the aggregated data.

The approached strategy was to create generic appliance models based on edge detection. The developed algorithm can be summarized in the steps presented in the Figure C.2. For a given aggregated data, a horizon of measurements is considered (for example 5 hours). Next, all the absolute edges' values (representing the activation and deactivation) of the horizon are detected. Then, the most relevant edges are clustered and their centroids detected. For example, the K-Means [50] clustering algorithm enables the definition of a number of centroids to be found in the data. Those centroids are inputted as power states parameters in the optimization algorithm. An always-ON state (standby) is also extracted from the aggregated data by taking the minimum measurement of the horizon. Finally, a generic minimum time is also set as parameter (between 4 and 10 minutes for this work). No sequence of states is considered for those generic appliances. The implementation of this algorithm is going to be discussed next.

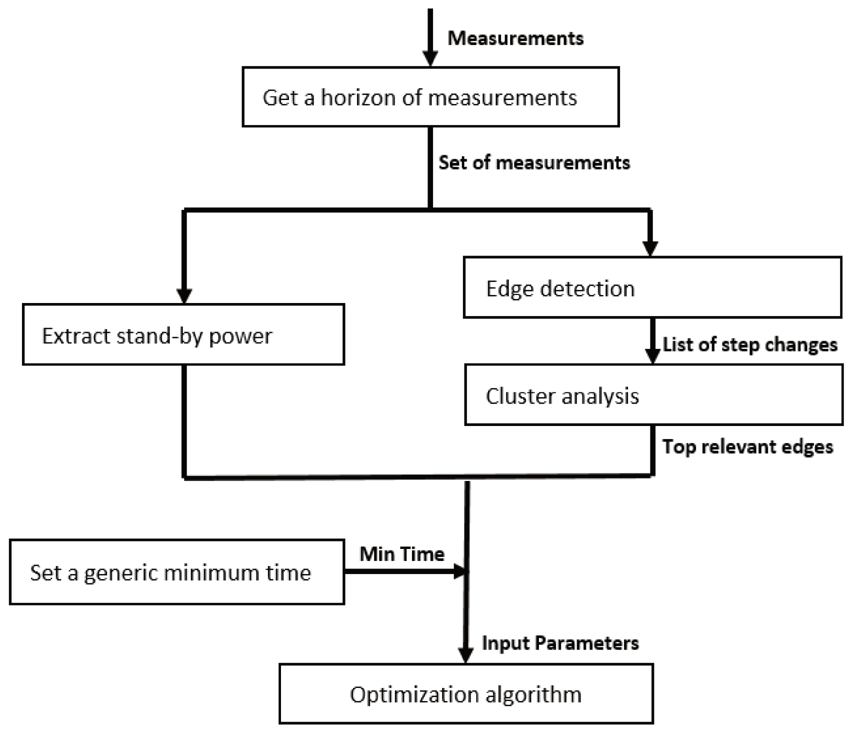


Figure C.2: Preprocessing algorithm

C.1.1 Preprocessing Algorithm Implementation

The preprocessing algorithm was implemented in Python. An integration with the AMPL software was made in order to call the solver after the table with input parameters is filled. The algorithm segments the measurements into horizons which are sent to the solver as input data. Each horizon of measurements can be split into windows in order to speed up the process. The optimization model was modified in order to accept standby states. The edges are extracted using Hart's edge detection algorithm, implemented in NILMTK. The K-Means clustering algorithm is applied to identify the top 4 or 5 states.

C.2 Test Case 1

The preprocessing algorithm was applied to the data of the first test case presented in Chapter 5. Only the active power was considered in order to have an extreme case of limited data for an unsupervised setting. The visual results are presented in the Figure C.3. The clustering algorithm was set to detect the five most relevant edges. The detected states were 675W, 3180W, 1807W, 127W and 4545W. Also, a standby power of 64W was extracted. Those 6 states were inputted to the optimization algorithm. Also, a generic minimum time of 5 minutes was defined. The Figure C.3 shows that the CDE appliance was not correctly assigned, although its edge was identified (4546W). This is due to the fact that the TV was also activated at the same time that CDE was. Since the TV edge of 239W was not identified, the aggregated data was filled with the WOE and the HPE. The label assignation of each appliance was made manually based on their edges and similarity behavior with the ground truth data. In a real world setting, those labels could be inputted by the user. The WOE appliance was also allocated in instants where the DWE and the HPE were predominant.

The Table C.1 shows the numerical results. The EAP error was about the same when comparing with the two other techniques. In average, the proposed model performed better than CO but worse than Hart's. The addition of more measurements is likely to help, although this possibility was not explored in this work. The presented preprocessing algorithm can serve as a starting point for a more advanced strategy in future works. For example, it is possible to implement a hybrid version of Hart's algorithm were appliance models are detected (such as the refrigerator) and combine with those generic load models.

Table C.1: EAP and EAT metric for the unsupervised case

	CO (%)		This Work (%)		Patt. Rec. (%)	
	ETE	EAP	ETE	EAP	ETE	EAP
CDE	94.3	94.3	85.8	85.8	8.7	**84.1**
DWE	23.9	**57.2**	29.6	63.3	5.6	74.1
FGE	32.6	62.3	21.6	58.7	3.6	**53.3**
HPE	11.9	61.2	6.9	**61.1**	0.1	65.9
WOE	145	205.6	104.2	**180.8**	-	-
TVE	23.7	76.9	11.3	**71.6**	-	-
Average	55.23	92.9	43.2	86.88	4.5	**69.35**

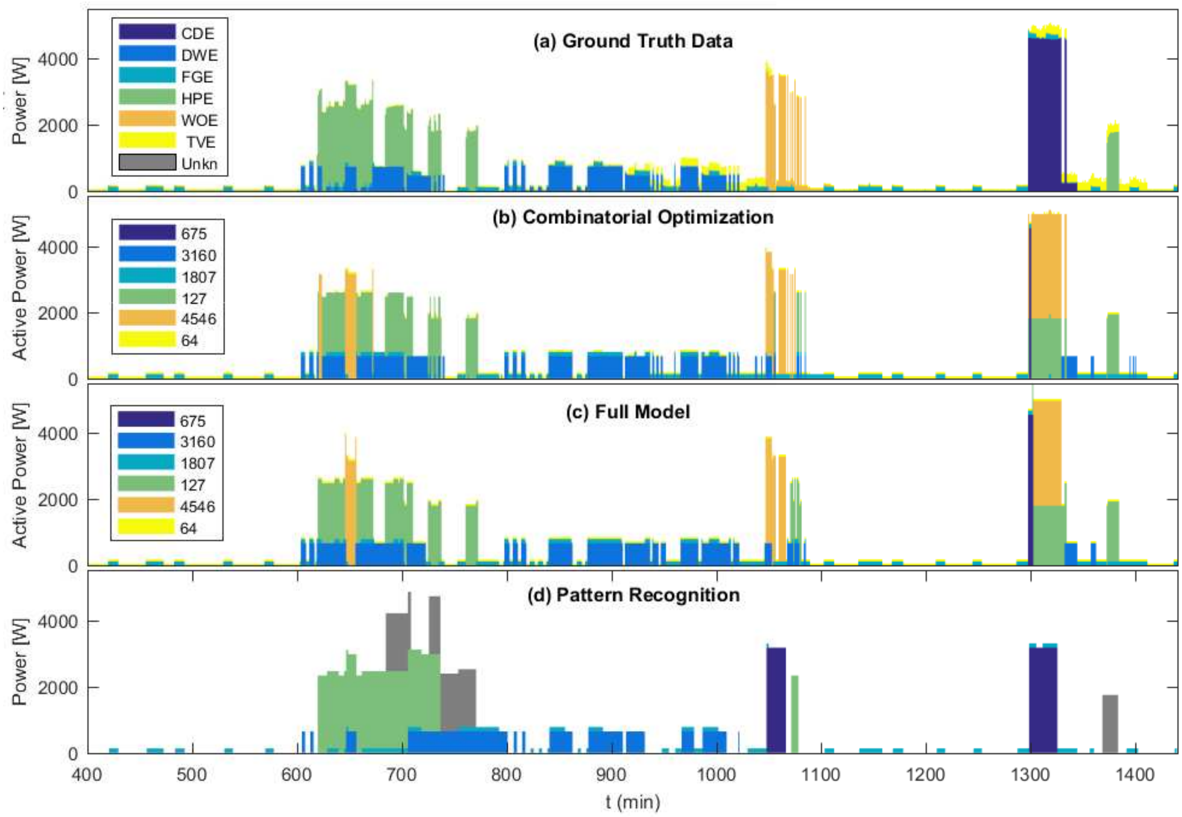


Figure C.3: Visual results for an unsupervised experiment

C.3 Test Case 2

The second test case considered as input data the measurements from a smart meter installed locally. The open source smart meter emonPi [53] by the UK based OpenEnergyMonitor [54] project was installed in a local household and their measurements processed. More details about the smart meter are given below along with some experiments.

C.3.1 Smart Meter Overview

The emonPi is an open source Raspberry Pi based energy monitor with two single-phase current sensors as input. The smart meter also accepts as input voltage and temperature sensors, although they were not installed. The smart meter also works as a server where its data can be accessed via Ethernet, WiFi or even from the internet using their website or Android App. The Figure C.4 shows the installed smart meter. Two current transformers (CT) are connected to the two main phases of the circuit panel. One of the main advantages of this smart meter is that since it is open source, it allows for modification of its internal code and implementation of disaggregation algorithms.

Figure C.4: Open source smart meter installed locally for experiments

C.3.2 Experiments

Six months of data were acquired by the smart meter with a 30 seconds interval sampling rate. The previously described preprocessing algorithm was implemented in order to detect generic appliances from those local measurements. A blind detection of generic appliances is performed, i.e., no pre-input of known appliances is performed.

The Figure C.5 shows an example of detected appliances based on their edges. The top graph shows the original measurements while the buttom graph shows the generic detected appliances. The 4 main detected states were 91W, 5931W, 1359W and 1636W. The detected standby state was 313W. For each new horizon, those four states and the standby state are updated.

The local household has some relevant devices such as two refrigerators and three electric showers. The preprocessing algorithm can be optimized for identification of only those appliances by adding personalized info as input parameter. Unfortunately, it is not possible to compare the measurements with the ground truth since the single appliances are not being monitored. However, it is possible to empirically infer some of the expected appliances to be ON based on the aggregated data. In addition, statistics can be tabulated based on those generic appliances. For example, the Figure C.6 shows histograms of the activation higher than 1500W in the last three months. It is possible to observe in the histogram of June that there is a certain number of activation for the area higher than 7000W which was not observed in the previous two months. This might indicate that a new appliance in this level is being used in the household. The appliance can be identified by tracking all the edges in the power level 7000W.

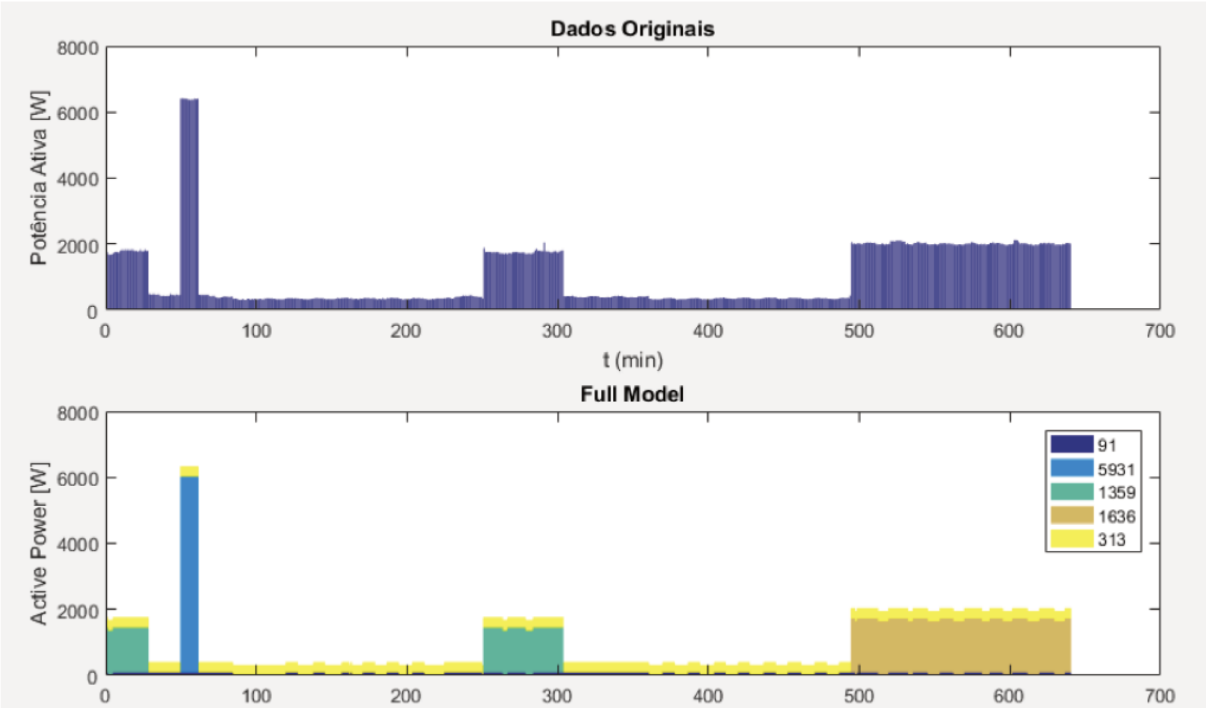


Figure C.5: Example of generic appliances detected by the algorithm

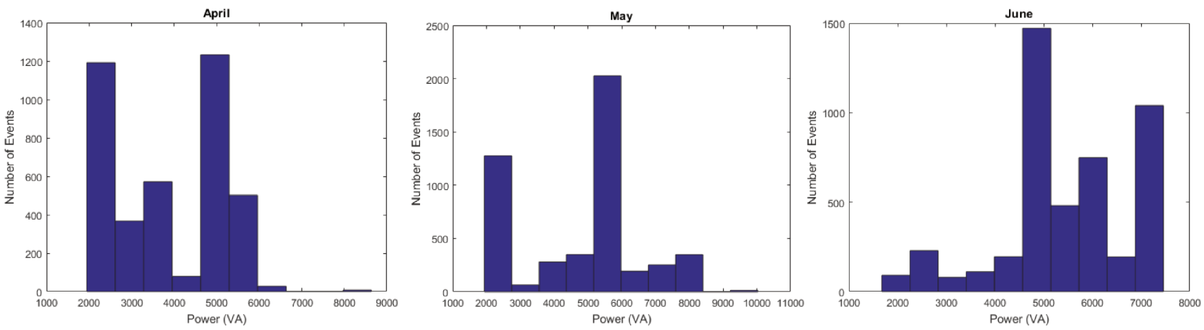


Figure C.6: Histogram analysis from the detected power levels

C.4 Summary

The proposed model works very competitively with other disaggregation methods as long as the input parameters are well defined. The use of a simple preprocessing algorithm serves as a starting point for the detection of appliances in an unsupervised scenario. Expansions of the preprocessing algorithm are possible in order to enhance its capability to extract the input parameter of the optimization model proposed in this work.