Temporal Mining Approaches for Smart Buildings Research

Huijuan Shao

Preliminary proposal submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science

Naren Ramakrishnan, Chair
B. Aditya Prakash
Chang-Tien Lu
Anil Vullkanti
Manish Marwah

Aug. 19, 2015
Arlington, Virginia

Keywords: Computer Science

Temporal Mining Approaches for Smart Buildings Research

Huijuan Shao

(ABSTRACT)

With the advent of modern sensor technologies, significant opportunities have opened up to help conserve energy in residential and commercial buildings. This research proposal focuses on three sub-problems. The first is energy disaggregation, i.e., separating the energy usage of each circuit or each electric device in a building using only aggregate electricity usage information from the whole house meter. The second sub-problem is to model daily life activity in a building with a view to automatically shutdown or startup units like the HVAC based on occupancy estimations. The last problem, non-invasive indoor activities tracking, aims to predict the locations of people inside a building. All these problems are approached using the methods of temporal data mining. Motif mining with constraints is exploited to distinguish devices with multiple states thus tackle the energy disaggregation problem. Time-gap constrained episode mining is used to detect activity patterns followed by the use of a mixture of episode generating HMM (EGH) model to predict home occupancy. Finally, the mixture EGH model can also help predict the location of a person to address non-invasive indoor activities tracking.

That this work received support from the HP labs is purely coincidental.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Electricity usage permeates all aspects of modern society. Its most conspicuous uses include urban contexts such as lighting, air conditioning, refrigeration, heating, and, powering appliances and gadgets but its penetration is pervasive across rural and industrial sectors. In 2014, the residential and commercial sector comprised nearly 40% of all the electricity generated in the U.S. [1]. Furthermore, our dependence on electricity will continue to grow as emphasis shifts away from fossil fuel based vehicles to electric vehicles.

While people generally agree on the importance of conservation and usage curtailment, they are often at difficulties to quantify *where, when* and *how much* electricity is consumed. Typically, residences and businesses receive monthly electricity bills indicating aggregate usage, with no information on the breakdown of consumption by appliances/devices, time of day, or day of week (this is an area in great flux, however). Research has shown that simply making such feedback available to users can reduce consumption by up to 50%, although typical saving are in the 9% to 20% range [1].

One obvious approach to determining the breakdown of consumption is to install power meters in every circuit (and sub-circuit) to capture consumption of individual devices in homes and offices. Such installation is costly and intrusive, making this option unviable in practice. An alternate solution, called energy disaggregation or non-intrusive load monitoring (NILM), first proposed by Hart [52], is to use analytics to *infer* the breakdown of consumption from an aggregate power measurement of a site. This drastically reduces the number of meters required per home/installation, typically to just one. Furthermore, depending on the analytics desired, it is possible to use the measurements already being recorded by a utility meter for disaggregation, especially in cases where utility companies have deployed smart meters. Energy disaggregation is hence today a booming area offering both challenging problems for data analytics and having practical relevance in a number of areas including sensor networks and building analytics.

Another approach to save energy in homes is to efficiently use electricity devices. In residential buildings, the biggest consumer of electricity is usually the HVAC (heating, ventilation, and cool-

ing) system, which generally accounts for 54% of the buildings electricity consumption [1]. How to automatically start up and shut down the HVAC unit is thus a key problem. One solution is to predict the occupancy at home is to begin by analyzing the activities of daily life inside the building. Based on the occupancy information, an automatic control system can be installed to operate the HVAC.

A related problem pertains to non-invasive indoor activities tracking. The goal here is to predict the locations of people inside a building without the use of invasive cameras.

1.1 Timeline

The first proposed research problem on energy disaggregation has been completed although the work will be extended to time-based motif mining and a new probabilistic models. The second task of activity of daily life patterns is underway. The third subject on non-invasive indoor activities tracking will start in this Oct.. A timeline of activities is shown in Figure 1.1.

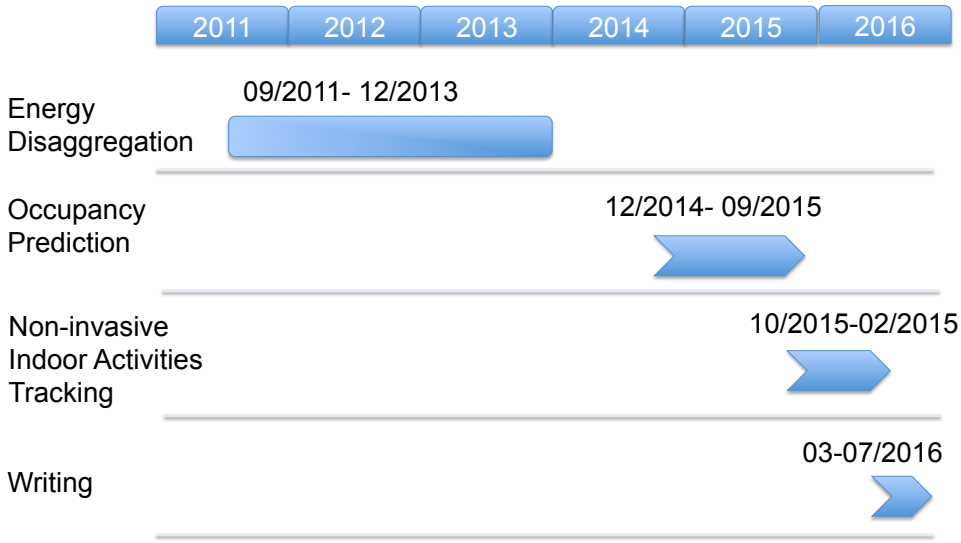


Figure 1.1: Timeline.

Chapter 2

Survey of Energy Disaggregation

2.1 Introduction

Electricity usage permeates all aspects of modern society. Its most conspicuous uses include urban contexts such as lighting, air conditioning, refrigeration, heating, and, powering appliances and gadgets but its penetration is pervasive across rural and industrial sectors. In 2014, the residential and commercial sector comprised nearly 40% of all the electricity generated in the U.S. [1]. Furthermore, our dependence on electricity will continue to grow as emphasis shifts away from fossil fuel based vehicles to electric vehicles.

While people generally agree on the importance of conservation and usage curtailment, they are often at difficulties to quantify *where, when* and *how much* electricity is consumed. Typically, residences and businesses receive monthly electricity bills indicating aggregate usage, with no information on the breakdown of consumption by appliances/devices, time of day, or day of week (this is an area in great flux, however). Research has shown that simply making such feedback available to users can reduce consumption by up to 50%, although typical saving are in the 9% to 20% range [1].

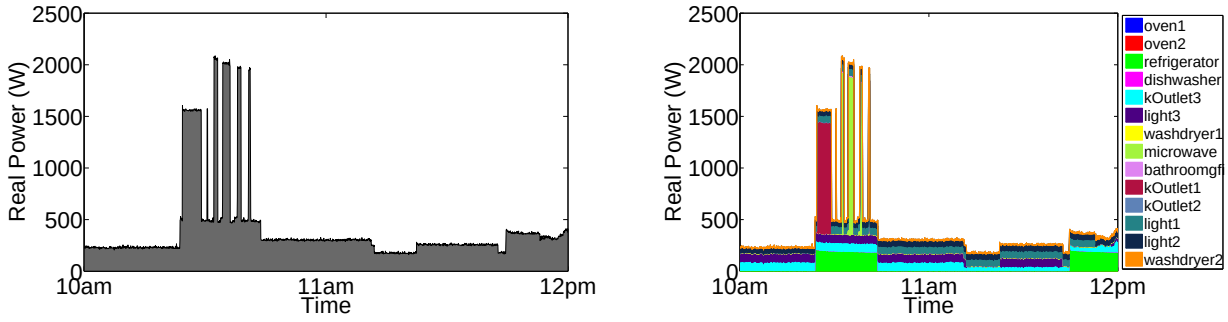
One obvious approach to determining the breakdown of consumption is to install power meters in every circuit (and subcircuit) to capture consumption of individual devices in homes and offices. Such installation is costly and intrusive, making this option unviable in practice. An alternate solution, called energy disaggregation or non-intrusive load monitoring (NILM), first proposed by Hart [52], is to use analytics to *infer* the breakdown of consumption from an aggregate power measurement of a site. This drastically reduces the number of meters required per home/installation, typically to just one. Furthermore, depending on the analytics desired, it is possible to use the measurements already being recorded by a utility meter for disaggregation, especially in cases where utility companies have deployed smart meters.

Energy disaggregation is hence today a booming area offering both challenging problems for data

analytics and having practical relevance in a number of areas including sensor networks and building analytics. Our goal in this paper is to provide a comprehensive survey of recent advances in the area of energy disaggregation with a focus on the data mining and machine learning algorithms used.

2.1.1 What is Energy Disaggregation?

Hart [52] first proposed the idea that power measurements at the main electric meter in a home can be used to deduce what appliances are turned on and how much electricity they are consuming. Figure 3.3 (a) shows aggregate power measurement, such as that at a main electric meter, from 10 am to 12 noon on a particular day. The goal of energy disaggregation is to decompose this consumption into its constituents as shown in Figure 3.3 (b), which shows fourteen disaggregated devices. It shows that, for example, the refrigerator turns on twice – from 10:15 am to 10:40 am, and then from 11:50 am to 12:00 noon. At other times, the refrigerator stays off.



(a) (b)

Figure 2.1: (a) Aggregate power. (b) Disaggregated information about devices and their power usage patterns.

Energy disaggregation research can be understood in terms of the *features* that can be extracted from power measurements and the underlying *algorithms* used.

One way to think of *features* is in terms of the sampling frequency of meters. Low sampling frequency data is typically sampled at less than 1 Hz, while high sampling frequency data is sampled at higher than 1 Hz. Some features derivable from low frequency data are real power, reactive power, low-order harmonics, and time of day. In addition to the ones inferable from low frequency data, features from high frequency data include many more characteristics such as harmonics, and current or voltage waveforms. Transient state features are only available from high frequency data. These features relate to transitory behavior seen in the current and voltage waveforms when a device is turned on or off. On the other hand, steady state features are stable features that persist after a device has changed its state. These can be obtained from both low sampling frequency data and high sampling frequency data.

Another way to classify features is in terms of: AC power and non-AC power. AC power characteristics are related to current or voltage, whereas non-AC power features include power line noises, device correlation, and contextual features like time or date, or weather information.

Initially, only AC power features such as real power and reactive power were studied [52]. With advances in electrical meter technology and availability of less expensive meters, the transient state generated when a device turns on or off could be recorded and used to identify devices [116]. Further, the raw current waveform [119], voltage waveform [75], and the transform of the current waveform [22] can also be exploited as features. Harmonics of non-linear devices have also been studied [22]. Non-AC power features, such as power line noises [106], time of day, and device correlations [65] are often combined with AC power features in modern systems.

Algorithms applicable to energy disaggregation can be categorized into supervised learning algorithms, unsupervised learning algorithms, and semi-supervised learning algorithms. The supervised learning algorithms include kNN methods [116], support vector machines [106], neural networks [111], genetic programming [12], sparse coding [70], as well as combinations of supervised learning algorithms [101]. Optimization algorithms used in the area of energy disaggregation have been drawn from integer programming [124], dynamic programming [11], and the viterbi algorithm [138]. Unsupervised learning algorithms have only been recently used in the last few years, and include hierarchical clustering [48], factorial hidden Markov models (FHMMs) [65], additive factorial approximate MAP (AFAMAP) [71], difference FHMM [105], and motif mining [115]. Semi-supervised learning algorithms [61, 75] have also been proposed.

2.1.2 Challenges

The field of energy disaggregation has evolved over the last twenty years; while some applications have achieved qualified success, there are several challenging problems that still need to be addressed before energy disaggregation can be used more widely. Some of these problems include:

1. The number of devices is typically unknown and can only be approximately estimated based on background information.
2. The number of power levels of each device is unknown. Some devices such as lights may have only two steady states, viz. on and off. Other devices have several steady states. For example, a microwave can operate in the states of defrost, heat with low power, or heat with high power. Estimating the exact number of states of a device is a hard problem.
3. Several devices may share the same real power and it is hard to distinguish these devices from only the recorded aggregated power values. For example, a light and a monitor could consume the same amount of real power (e.g., around 38W). With more devices that share the same real power, additional features are necessary to disambiguate among them.
4. Many devices may turn on or off at the same time. A PC and printer likely turn on and off together, thus making it difficult to separate them from the aggregated power.

5. Instead of having a discrete range of power levels, there are devices whose power consumption levels vary continuously, e.g., variable speed devices (VSD), and lights with dimmers. Once their power usage is aggregated with that from other devices, the disaggregation problem becomes increasingly difficult.

6. Some devices are always on and seldom operated by users. Because the operations on these devices are rare, it is hard to identify these devices from prior historical data.

The above problems are exacerbated in the case of commercial buildings. While the voltage in residential buildings is typically 110 or 220 volts, the voltage in commercial buildings is traditionally higher, at 208 or 460 volts. Three-phase power is usually split into single phase or two phases before reaching residential buildings. In contrast, commercial buildings commonly use three phases. Further, the devices in these two types of buildings are different. Residential buildings usually have devices such as microwaves, refrigerators, ovens, lights, washers/dryers, and air-conditioners. The start-up duration of these devices is short before they come to steady states. Commercial buildings install more VSDs including heating, ventilation, and air conditioning (HVAC) systems, variable-speed motor devices, and dimmable lighting. Further, bank of lights typically connect into a circuit together and are powered on/off at the same time. Generally, we face greater challenges in commercial buildings than in residential buildings. Norford and Leeb study non-intrusive load monitoring challenges for commercial buildings [102]. First, load detection in commercial buildings is harder because there are many devices powered on and off together. Second, the start-up transient state of devices in commercial building is much longer than those in discrete devices, which dominate residential buildings. Finally, in commercial buildings, reactive power is reduced to make loads resistive, such as fluorescent lamp fixtures.

2.1.3 Scope of This Survey

Our objective is to provide an introduction to this space for a data mining audience. While surveys exist on energy disaggregation, e.g., [137], [87], and [140], they are mostly aimed at an electrical engineering audience and are not suitable for data mining practitioners. Our survey provides both the background knowledge necessary and an overview of all aspects of machine learning and data mining as applied to energy disaggregation.

In all survey papers, it is helpful to scope out what the survey does *not* cover. The problem of disaggregation resurfaces in the context of other utilities besides electricity, e.g., water [35], natural gas [46], and music [112]. We do not cover these domains here and focus exclusively on electricity. Second, there are many problems that appear related at first glance, e.g., blind source separation [21, 36, 85] but are quite distinct from disaggregation. In the case of blind source separation, the goal is to separate sources from at least as many observations whereas in the case of disaggregation, only one aggregate signal is provided.

2.1.4 Organization

The contents of this survey are organized as follows. In Section 2.2, we introduce some basic conceptions of power, electricity, and electrical devices. Next, in section 2.3, we list several historical definitions of energy disaggregation and present our working definition for the survey. Characteristics which are used to disaggregate devices are categorized in Section 2.4. It also describes how to setup an experimental testbed and record necessary data with meters. Section 2.5 summarizes a range of algorithms that have been historically used for energy disaggregation. Section 2.6 takes up the important aspect of defining evaluation measures for disaggregation. Section 2.7 enumerates some tools, datasets, and software available to data mining researchers. Finally, Section 2.8 identifies promising research direction in this space.

2.2 A Primer on AC power

We will briefly review some background on concepts in AC power before we describe algorithms for energy disaggregation.

2.2.1 Electricity Transmission

The power we use in our homes and offices is generated at power plants and transmitted to buildings. Figure 2.2 illustrates how power is transmitted and transformed. Initially, a power plant generates 3-phase electrical power. The voltage is stepped-up to several hundred kilo-volts for transmission. In power substations, transformers decrease the voltage. Usually after several substations, the voltage is decreased to 4,800 volts as medium voltage power. This medium voltage power is then split for two different kinds of usage: residential and industrial. To supply power for industrial or commercial buildings, a 3-phase transformer changes the voltage to 208 volts or 460 volts. Finally, a three-wire power service is delivered to end users. To transmit power to residential buildings, a 3-phase transformer again steps down the voltage. The power is then transmitted by power poles, and a power drum decreases voltage to around 110 volts in the U.S. or 220 volts in other countries. In the end, a 2-phase or 3-phase power service is connected into a home for usage.

2.2.2 Circuits and Devices

Normally power in residential or commercial buildings connects through two or three main phases. Many circuits then draw power from these main phases in parallel or in series. While most residential devices connect to a single phase some heavy duty appliances require a two-phase connection. Figure 2.3 (a) depicts a typical connection in residential buildings. There are two main phases:

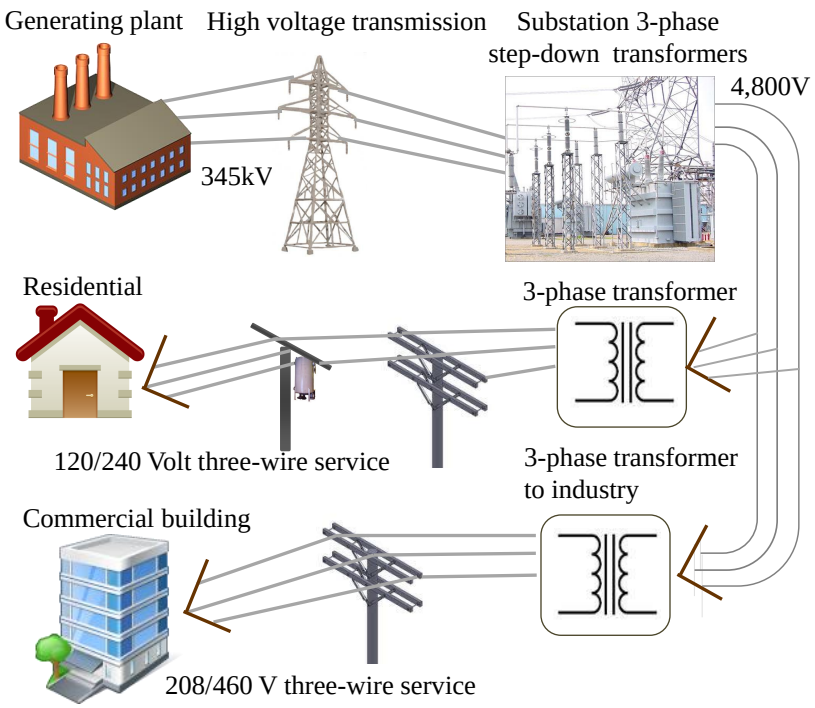


Figure 2.2: Electricity generation and transmission to residential and commercial buildings.

phase1 and phase2. Three circuits connect into these two phases. In the first circuit, two lights connect in series to phase 1 and the ground. In the second circuit, a washer/dryer connects to both phases. In the third circuit, a television connects to phase2 and the ground. Note that it is possible that several devices connect to one phase in a circuit.

An example of a circuit in a commercial building is depicted in Figure 2.3 (b). Devices in any circuit connect to two or all of the three phases. There are two circuits connecting to phase1 and phase3 in red. Both these circuits supply power to a bank of lights. Therefore, when people switch on/off, these lights powers on/off at the same time. A copier/printer draws power from phase1 and phase2 in yellow. A computer server connects to all three phases in blue.

2.2.3 Voltage and Current

The voltage transmitted from a power plant is typically 3-phase sinusoidal. Figure 2.4 depicts the waveform of the three phases of AC power. Each phase V_1, V_2, V_3 has a sinusoidal voltage waveform. Between each phase, there's a phase angle difference of $\pi/3$.

These three voltages can be represented mathematically as the following three equations. In these equations, ω represents the frequency of power. While the frequency varies by country, it is 50 or 60 Hz in most places. For example, it is 60 Hz in the U.S..

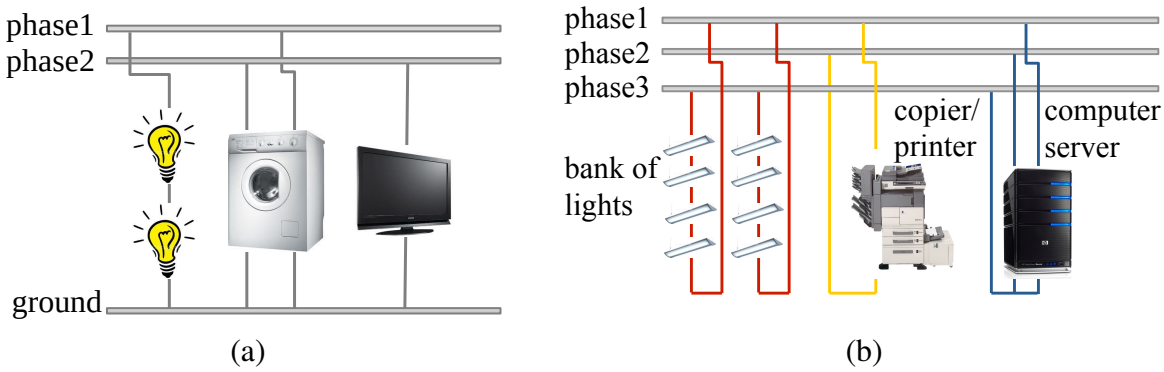


Figure 2.3: Example of a circuit in (a) residential building and (b) commercial building.

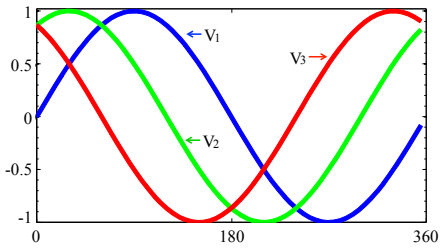


Figure 2.4: Three phase power waveform.

$$
\begin{aligned}
V_1 &= V\sin(\omega t) \\
V_2 &= V\sin(\omega t + \frac{2\pi}{3}) \\
V_3 &= V\sin(\omega t + \frac{4\pi}{3})
\end{aligned}
$$

When a circuit is activated by a sinusoidal source voltage with frequency ω, a current in this circuit is generated. The relationship between current and voltage depends on the impedance in the circuit. Ideally, there are three types of impedance: resistor, inductor, and capacitor. Resistors draw power and generate heat. An example of this is an electrical stove. Capacitors store energy in an electrical field. Inductors store electrical energy in a magnetic field. Figure 2.5 shows three idealized AC circuits with only resistor R in the unit of ohm (Ω), inductor L in the unit of henry (H) or capacitor C in the unit of faradays (F) where $V_s(t)$ and current $i(t)$ are AC voltage and current.

The i-v relationship for each circuit element of these three types of impedance is described by the following formulas. For the resistor circuit, according to Ohm's law $V = IR$,

$$i_R(t) = \frac{V_s(t)}{R} = \frac{A}{R}cos(\omega t). \tag{2.1}$$

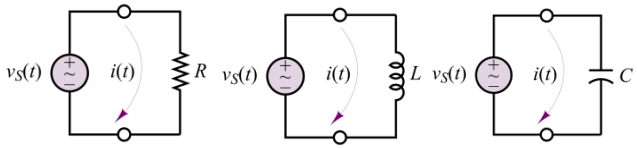


Figure 2.5: AC Circuit of basic loads: resistor, inductor, and capacitor (courtesy: [51]).

For the inductor circuit, the relationship between current and voltage is

$$i_L(t) = \frac{A}{\omega L} cos(\omega t - \frac{\pi}{2}) \tag{2.2}$$

For the capacitor circuit, the relationship between current and voltage is

$$i_C(t) = \omega C A cos(\omega t + \frac{\pi}{2}) \tag{2.3}$$

where A represents the amplitude, and ω denotes the frequency.

In practice the impedance of any electrical device is composed of at least one of these three types: resistors, inductor, and capacitor. A device may include several resistor units or inductor units or capacitor units. For example, the mainboard of a computer typically contains a number of capacitors.

2.2.4 Real Power and Reactive Power

In the field of electrical engineering, real power and reactive power are concepts used to characterize the power consumption of electric devices. Meters typically measure current in amperes (A), voltage in volts (V), real power in watts (W) and reactive power in volt-ampere reactive (VAR).

A scatter plot of real power and reactive power for different devices is given in Figure 2.6. The water heater, IR light, and fan only consume real power (no reactive power) because these devices are composed exclusively of resistors. The values of real power of these three devices are different from each other. The refrigerator and water pump have similar reactive power at around 450 VARs, but their real power values are 750 W and 250 W, respectively.

Real power and reactive power values can be obtained by the values of voltage, current, frequency of AC power and the phase angles of voltage and current. Suppose we have voltage, $v(t) = Vcos(\omega t - \theta_V)$ and current, $i(t) = Icos(\omega t - \theta_I)$, where ω is the base frequency of AC power, θ_V denotes the phase angle of voltage and θ_I represents the phase angle of current.

Instantaneous real power at time t is given by:

$$P(t) = v(t) \cdot i(t) \tag{2.4}$$

The average root mean squared (RMS) real power usage over a period of time is typically what is used to measure power consumption in our electricity bill. Assume V and I represent the maximal

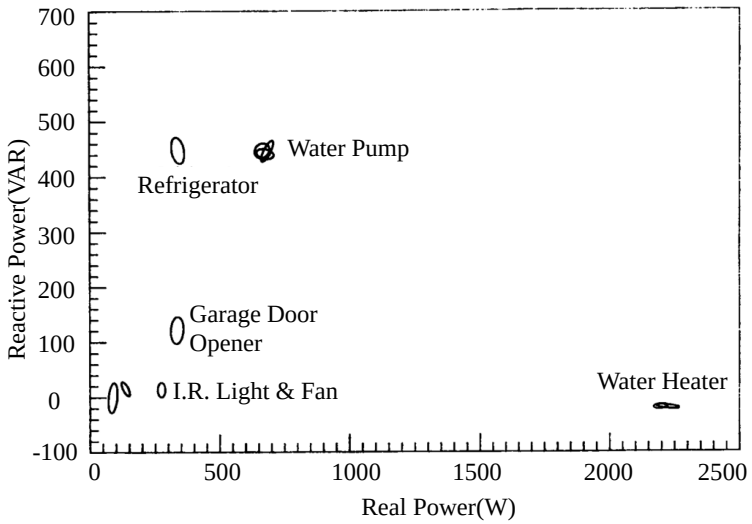


Figure 2.6: Real and reactive power for different devices (courtesy: [52]).

value of voltage and current, then $V_{rms} = \tilde{V} = \frac{V}{\sqrt{2}}$ and $I_{rms} = \tilde{I} = \frac{I}{\sqrt{2}}$. The average power P_{av} is the inner product of voltage and current $P_{av} = \tilde{V}\tilde{I}cos\theta$, where θ is the phase difference between voltage and current, i.e. $\theta = \theta_V - \theta_I$.

The relationship between real power and reactive power is summarized in Figure 2.7 and by Equation (2.5b).

$$S = \tilde{V}\tilde{I}cos\theta + j\tilde{V}\tilde{I}sin\theta \tag{2.5a}$$

$$S = P_{av} + jQ \tag{2.5b}$$

where S is the apparent power, P is the average real power, and Q is the reactive power.

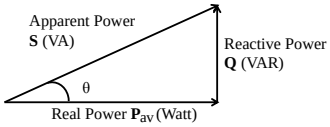


Figure 2.7: Power Triangle.

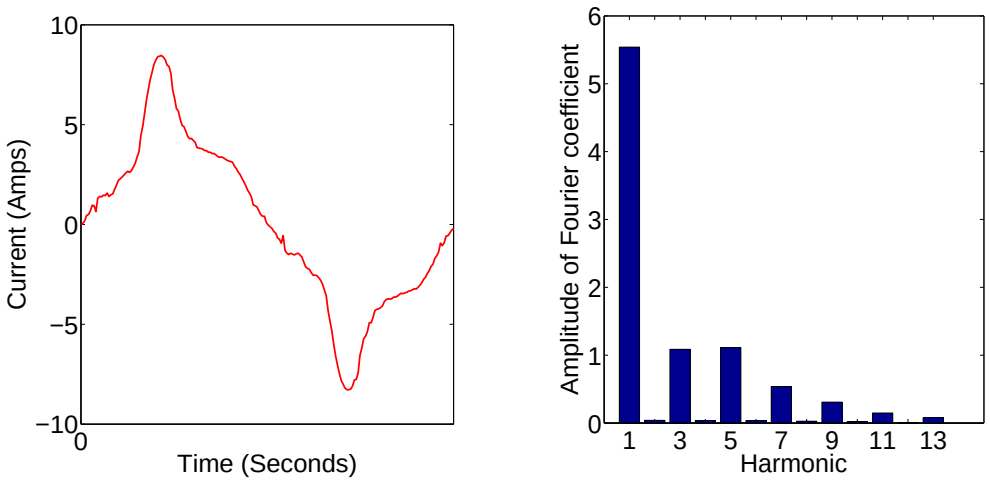
From Figure 2.7, we can calculate the power factor, $\cos\theta$. For resistive devices, the power factor is equal to 1, which means there is only real power consumed when the device is on. For pure inductive or capacitive devices, the power factor equals 0, which means there is only reactive power consumed when the device is on. If a device has resistor $R\ ohms$, inductor $X_L\ henries$, and capacitor $X_C\ farads$, then the real power and reactive power values are as given by Equations

(2.6a) and (2.6b).

$$P_{av} = R \cdot I^2 \tag{2.6a}$$
$$Q = (X_L - X_C) \cdot I^2 \tag{2.6b}$$

2.2.5 Harmonics

For those circuits containing an inductor or capacitor, the current waveform is typically non-sinusoidal as shown in Figure 2.8 (a), an example taken from the current waveform of a cycle of Circuit4 in the BLUED dataset described later in the survey [9]. This (or any) type of waveform



(a) (b)

Figure 2.8: Circuit 4 (a) Current Waveform and (b) Harmonics.

can be expressed as a Fourier series. Consider a periodic waveform $x(t) = x(t + T_0)$, where T_0 is the period. $x(t)$ can be rewritten as

$$x(t) = \sum_{n=0}^{\infty} A_n cos(\frac{2\pi nt}{T} + \theta_n) \tag{2.7}$$

where $\omega = \frac{2\pi}{T} = 2\pi f$, A_n denote the amplitudes and θ_n denote the phases. Here ω is the fundamental frequency; the integer multiples of basic frequencies 2ω, 3ω, and so on are referred to as *harmonics*. Figure 2.8 (b) depicts a Fourier spectrum of the current waveform of Figure 2.8 (a). The x-axis is the ordered harmonics and y-axis shows the amplitude of harmonics. Typically, odd harmonics are of interest, e.g., 3rd harmonic, 5th harmonic, and so on.

2.3 Definition of Energy Disaggregation

Since Hart first defined the problem of energy disaggregation [52], its exact definition has varied slightly. To make the problem more tractable and to tailor it to specific use cases, researchers have made varying assumptions about what information is given. The bare bones version of disaggregation, as originally proposed by Hart [52], assumes that only aggregated current and voltage data is available as features. Other researchers assume additional information is known, such as, the total number of devices[1], the number of steady power consumption states of devices and their corresponding power levels, non AC power features. There are thus numerous formulations of the energy disaggregation problem and these distinctions need to be considered while their performance is compared. Broadly, the disaggregation problem can be solved in supervised, unsupervised or semi-supervised settings.

In supervised learning approaches, labeled data is available for a period of time, that is, the on/off state of each device is known. For instance, [87] assumes that all devices are known and formulates the objective function as one of minimizing error between the disaggregated devices and the corresponding ground truth devices. In [71] it is assumed that the power levels of individual devices are known. The problem then is to find the on/off events for different devices over a period of time. The input to supervised learning methods is thus training data consisting of aggregated power and non-power features over time T, each device's power consumption over T. Once a model is trained, given new data over time T', the output is the disaggregated power for each device over T'.

Unsupervised or semi-supervised energy disaggregation is a harder problem because, comparing to supervised learning approaches, the only known information is the aggregated data. The number of devices/circuits or the characteristic of each device, such as power levels are unknown [48, 133] or assumed by the researchers to be known [61, 65, 71, 105, 115]. In spite of the difficulties, the disaggregation results of unsupervised approaches may achieve as good as that of supervised approaches. Note that evaluation of any method, whether supervised or unsupervised, requires that ground truth information is available (number of devices and their on/off power states).

In addition to power related features such as current, voltage, or the non-powers features such as time of day, day of week, month season, weather information, may be given. The definition of energy disaggregation regarding power features is generalized as follows.

Energy Disaggregation:

Given the aggregate power consumption, $Y = y_1, ..., y_T$, and a set of power related and contextual features, $f = f_1, ..., f_T$ over a period of time T, the problem is to estimate the disaggregated power consumption of M devices $\hat{X_m} = \hat{x}_1^{(m)}, ...\hat{x}_t^{(m)}, ...\hat{x}_T^{(m)}, m \in [1, M]$, such that a loss function on the

[1]Disaggregation can only be meaningfully performed for devices/appliances whose power consumption is over a minimal threshold, typically 50 to 200 W. The number of devices here refer to those above this threshold.

sum of the power consumption of the M devices and the aggregate power consumption is minimal.

$$\min_{\hat{x}_t^{(m)}}\{\sum_{t=1}^{T}\mathscr{L}_t(\sum_{m=1}^{M}\hat{x}_t^{(m)}, y_t)\} \tag{2.8}$$

where $\mathscr{L}_t$ is the loss function between the sum of M estimated time series at t, and y_t is the ground truth aggregated power feature at time t. $\mathscr{L}$ is usually $\mathscr{L}1$-norm $\sum_{m=1}^{M}|\hat{x}_t^{(m)} - y_t|$ or $\mathscr{L}2$-norm $\sum_{m=1}^{M}(\hat{x}_t^{(m)} - y_t)^2$.

For supervised learning, the ground truth of M time series $X_m = x_1^{(m)}, ...x_t^{(m)}, ...x_T^{(m)}, m \in [1, M]$ corresponding to M circuits or devices is also given.

2.3.1 Technology Timeline

The evolution of approaches to energy disaggregation is summarized in Figure 2.9. The algorithms for this problem have developed through several stages by incorporating features of increasing levels of sophistication. In the first stage of development, algorithms were based on the features of real and reactive power, transient startup of current or power, and harmonics. In the next stage of development, algorithms were based on wavelet transform of current, duration time of specific steady state of real power, the waveform of current or voltage, and current/voltage noise. In the current stage of development, algorithms use eigenvalues of current, devices correlation. As people have been trying different features - the ones used recently say EMI are not necessarily better, nor are most people using them. They are just novel, and good results were reported, but there may be other reasons like having to install sensors etc that everybody may not adopt them.

At the same time, algorithms adopted in this area experience an accelerating progress; from supervised learning algorithms, including optimization algorithms and statistical models; until unsupervised learning and semi-supervised learning. Supervised learning employs each circuit/device's data as training data, which is laborious to collect. These algorithms are marked in yellow in Figure 2.9. The green boxes in Figure 2.9 represent unsupervised or semi-supervised disaggregation algorithms. They include rule-based approach; pair-wise match and neural network which were proposed in 1990s; k-nearest neighbor (KNN); support vector machine (SVM) and kernel based subspace classification (KSC); general likelihood ratio; genetic algorithm; auto-regression and moving average; radial basis function network (RBFN); decision tree; adaBoost; Bayesian classifier; space coding; dynamic bayesian network; closure rules; viterbi algorithms; dynamic programming, integer programming, and nonnegative tensor factorization. Also, the algorithm which is often employed in information processing area, wavelet transform, was used in 2000 [22]. Since 2006, unsupervised and semi-supervised algorithms have become the preferred approach to identify devices. These include hierarchical clustering, factorial HMM, duration probability density function (PDF), approximate factorial additive MAP (AFAMAP), difference FHMM which adds prior knowledge of device, hierarchical dirichlet process hidden semi-Markov model, motif mining, and contextually supervised source separation. In the next two sections, we explain the features and the algorithms in detail.

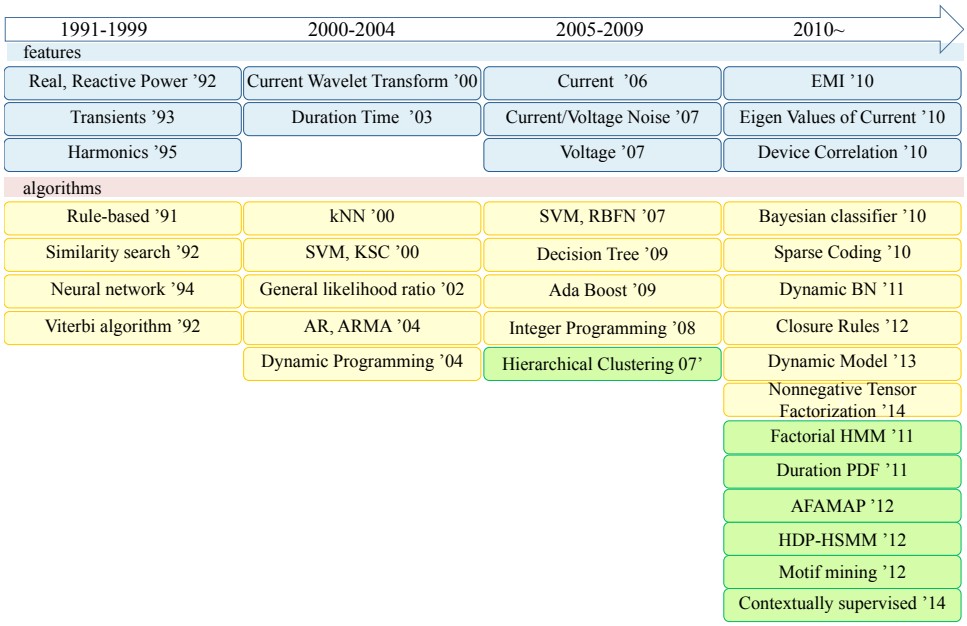


Figure 2.9: Energy Disaggregation Definition Example.

2.4 Disaggregation Features

In this section we briefly outline several features that are used in disaggregation and classify them based on different types like AC vs. non-AC or steady vs. transient state. We presented some of these features in section 2.2 like - voltage, current, real power, reactive power, harmonics generated by current and voltage. Some more basic and derived features include: startup of current; waveform of current; wavelet of current waveform; eigenvalue of current waveform; voltage waveform; voltage noise; EMI by gauging noise; electromagnetic field around the devices; duration time of power levels by calculating current and voltage; and time correlation of devices.

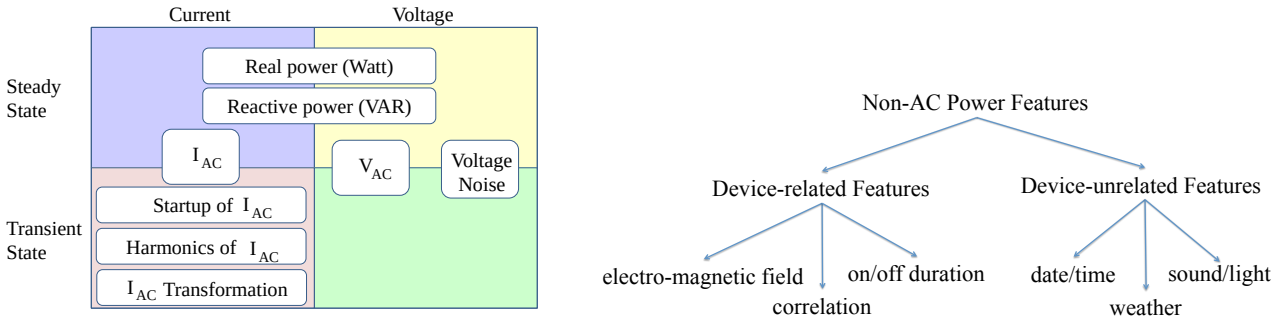


Figure 2.10: Category of (a) AC Power Features and (b) Non-AC Power Features.

Figure 2.10 (a) displays the classification of AC power features. AC power features are related to

current or voltage. These features can also be classified based on the stability of operating states. The steady state refers to the stable state after a device turns on. For example, real power, reactive power, and apparent power are all steady state features. Transient states refer to variable states during a very short period of time when a device turns on or off. Transient state features are generally derived from the startup shape of current, voltage, harmonics or harmonics transformations.

Non-AC power features are summarized in Figure 2.10 (b). These non-AC power features are classified into two categories - device-related and device-unrelated. Device-related category includes electromagnetic field (EMF); operation correlation between devices and etc. The EMF is produced when certain devices are on. The device-unrelated category is comprised of date or time features, such as month of the year, day of the month, day of the week and time of the day. Also, it includes the ambient temperature, which plays a crucial role in determining the functioning of the HVAC system. Further the device-irrelevant category include features like sound and light produced by an electrical device. A sensor can be installed near a device to record such features.

2.4.1 AC Power Features

Steady state

As shown in Figure 2.10, steady states include real power, reactive power, current, voltage, and voltage noise. This data can be either read directly from meters sampled at low frequency or calculated indirectly from high frequency voltage and current data. Suppose the basic current/voltage frequency of AC power, ω, is 60 Hz and the sampling frequency of recorded data is f, i.e., there are in total $f/60$ number of sample points in each cycle. The real power value is the average product of current and voltage in a cycle as in Equation (2.9).

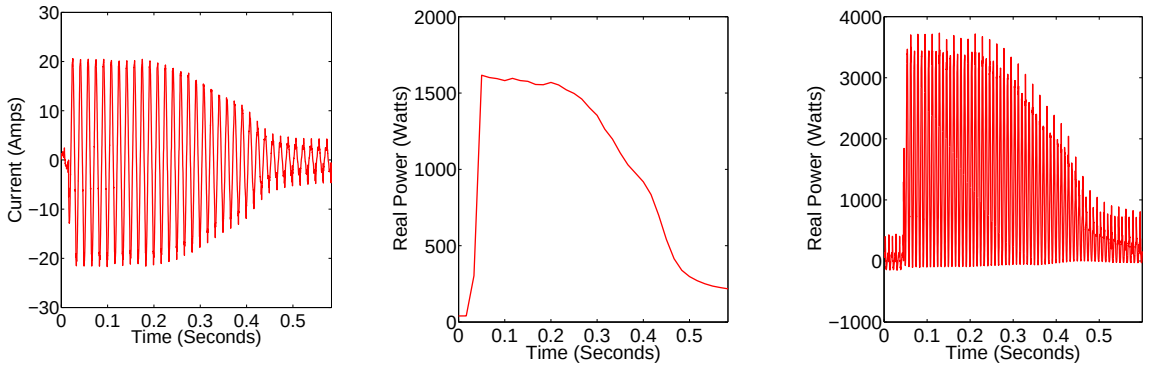
$$P_{av} = \frac{\sum_{t}^{t+f/60} v(t) \cdot i(t)}{f/60} \tag{2.9}$$

Real power is the most basic feature and used by almost all prior work in energy disaggregation [11, 12, 44, 52, 96, 109]. Reactive power is also widely used as a feature, e.g., in [39, 52, 76]. Figure 2.6 shows real power and reactive power features of different devices. For some devices real and reactive power are sufficient in distinguishing between them. The refrigerator and water pump have similar reactive power but different real power - thus using the real power feature, we can separate them. The refrigerator and the garage door opener have similar real power but different reactive power - thus the reactive power is the distinguishing feature in this case. Steady state features are also derived from the variations of real or reactive power. For instance, in [98], the slopes of both active and reactive power are extracted as vectors.

Transient state

High frequency data, from which the current waveform or voltage waveform can be recovered, offers rich features that can be applied to energy disaggregation. These features include the startup of current, harmonics of current, harmonics of voltage, voltage noise and its transformations.

Startup duration and transient power: Startup duration and transient power are recorded when a device is turned on. Usually a non-linear device, like a microwave, has such a distinguishing feature. When this kind of device turns on, the power usage usually changes to a temporary high value for few or milliseconds, then jumps into a steady state for a longer time. This temporary startup duration and shape feature varies from one device to another. Comparing the transient power changes with the steady state, the trail of power changes against time looks like a spike or a curve with changing slope. Figure 2.11 shows examples of current, average real power and instantaneous real power in the first 0.5 seconds of a refrigerator turning on in the BLUED dataset [9]. In figure 2.11 (a), there are three areas in this waveform. When $0 < t < 0.02s$, the current is in a steady state. When $0.02s \leq t \leq 0.45s$, the current is in a transient state, during which the amplitude of the current changes rapidly. When $t \geq 0.45s$, current comes again into steady state. Figure 2.11 (b) shows the shape of corresponding average power. It jumps to 1600 watts in a very short period of time, then gradually decreases to 200 watts. This real power is calculated by every cycle of 1/60 second by the Equation (2.9). Figure 2.11 (c) depicts the instantaneous real power. There are 200 points in each cycle. The instantaneous power changes very frequently.



(a) (b) (c)

Figure 2.11: (a) Transient and Steady State of a Sinusoidal Current from a Refrigerator. Transient Shapes for a Refrigerator (b) Real Power and (c) Instantaneous Real Power.

The transient energy is calculated as $E_{transient} = \int_{t_s}^{t_s+\delta t} v(t)i(t)dt$, where t_s is the start time and δt is the startup duration. And the corresponding real power is calculated as $P(t) = \frac{dE_{transient(t)}}{dt} = v(t) \cdot i(t)$.

This startup duration and shape of current or power feature can be used standalone or by integrating with other features. [123] gives a typical example of the latter case. The startup duration and the

shape of current or power feature is combined with real power and reactive power to distinguish each device among refrigerator, washing machine, and fluorescent light. Note that this transient startup may be called transient spectral envelope [116], or transient power.

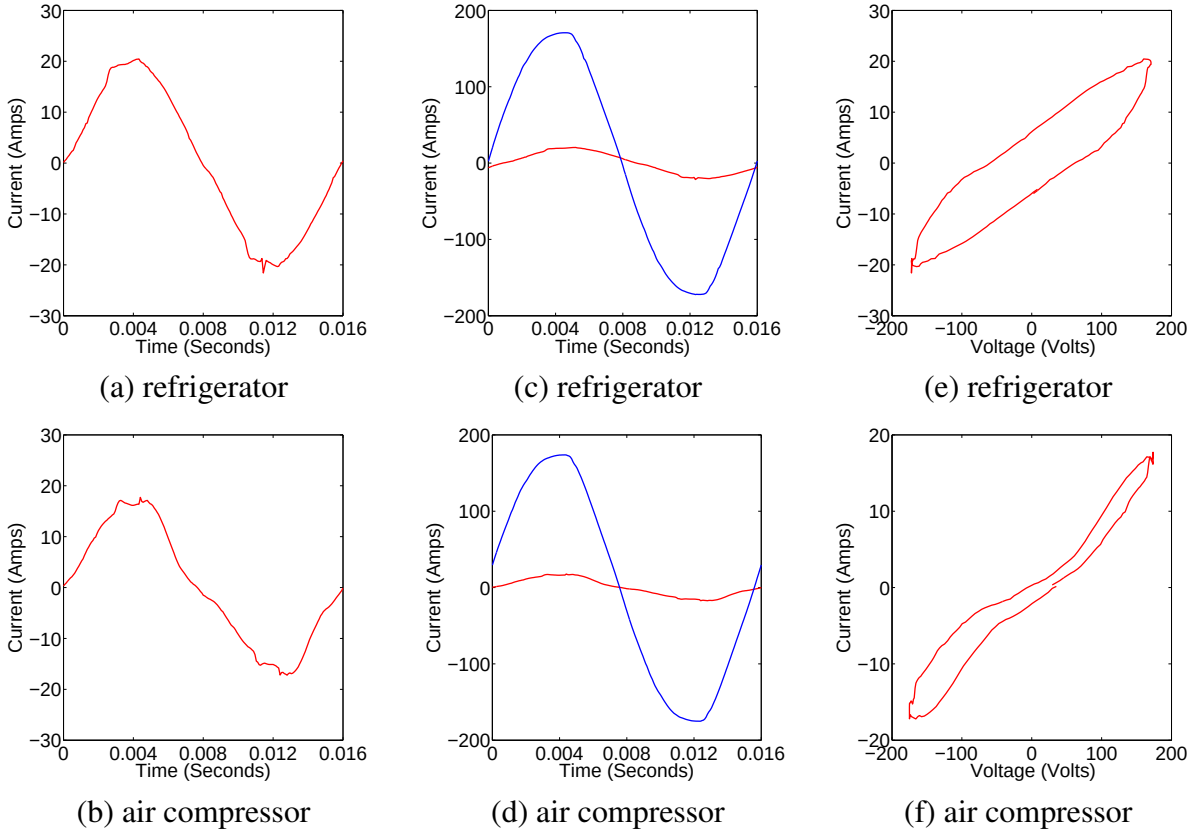
Current or voltage waveform: Current waveform I_{ac}, which can be simply read from high frequency recorded data, is a typical feature to discriminate devices. The waveforms generated by non-linear devices are very different because of the waveform distortion brought by each device. Figure 2.12 (a-b) illustrates the current waveform of two devices, refrigerator and compressor in BLUED dataset. From them, we can see that both the magnitudes and the distortions of the current waveforms differ from each other. The maximum current magnitude of a refrigerator is 20 Amps while that of compressor is around 16 Amps.

Current or voltage waveform features have been applied in previous work. Unprocessed current waveform is regarded as a feature in [124]. This paper shows that the raw current waveforms of a microwave oven and a toaster oven are different in a cycle. Therefore, these raw current waveforms can be used to separate these two devices. However, raw current waveforms are prone to change with noise. Also, [40] analyzes the current waveform of eight devices and classifies them as A/C, refrigerator, compressor, fan (VSD), elevator (converter), elevator (M/G set), fluorescent lights and computers. Similarly, standalone voltage waveform is used as a feature in previous work. The distortion is mainly generated by non-linear devices. By analyzing the voltage shapes, we can figure out which device is on. The difference of voltage waveform is that the maximum value of voltage is approximately 116V in the U.S.. [34] uses the distortion of voltage waveform to separate transient shapes as described in [119]. The combination of current and voltage waveform is proposed as features. [53] shows that the disaggregation results perform better than adopting only either the current waveform or the voltage waveform.

In order to overcome the shortcomings of raw current or voltage waveforms, several variations or transformations of these waveforms have been proposed. The first is voltage_current (VI) or current_voltage (IV) trajectory. It is useful because for dynamic devices such as air conditioner, the current waveform may vary from cycle to cycle. Figure 2.12 (e-f) illustrates the current trajectory difference between two devices, a refrigerator and an air compressor in the BLUED dataset. From Figure 2.12 (c) and (d), we can see that there is slight difference between the current and voltage. But comparing the current against the voltage as Figure 2.12 (e) and (f), the V-I trajectory is quite different. [75] utilizes the geometrical properties of V-I trajectory to sift devices.

The transformations of current or voltage waveform, including the Fourier transform, the wavelet transforms, and the eigenvalue decomposition are also useful. [122] utilizes both the short-time Fourier transform (STFT) [22] employs the wavelet transforms of current or voltage waveforms to identify devices. The eigenvalue of current or voltage waveforms is analyzed as a feature in [87]. Figure 2.13 depicts an example of how two devices a circuit and dining room light can be identified by eigenvalues. These two devices both have large first eigenvalues and small second eigenvalues. The difference between these two devices lie in that the first eigenvalue of the dining room light is larger than the first eigenvalue of the circuit.

Harmonics: Harmonics is another variation of current or voltage waveform. Harmonics are the



(a) refrigerator (c) refrigerator (e) refrigerator

(b) air compressor (d) air compressor (f) air compressor

Figure 2.12: Current waveform of (a) a refrigerator and (b) an air compressor. The current and voltage of (c) a refrigerator and (c) an air compressor. The V-I trajectories of (e) a refrigerator and (f) an air compressor.

integer multiplies of the fundamental frequency of the waveforms. They are generated by non-linear devices such as VSDs, electronic ballasts for fluorescent lighting, switching power supplies, or rectifiers when these devices start up or after they are on. These waveforms are distorted to be non-sinusoidal thus reflect the inherent characteristics of devices. It plays an important role to help distinguish devices when two devices share the same real power and reactive power. Harmonics can only be obtained from high frequency data.

Figure 2.14 (a) and (b) illustrate that an air compressor and refrigerator have similar real power. By analyzing the first three harmonics, each of them can be identified. The magnitude of the first harmonic of air compressor is larger than that of the refrigerator. And the magnitude of the second harmonic of air compressor is smaller than the magnitude of the second harmonic of the refrigerator.

Harmonics have been employed in prior work [4, 76, 81, 97, 119, 131]. Generally only the odd harmonics are utilized. As the best of our knowledge, the highest employed harmonics is the 15 odd harmonics [119]. VSDs are hard to distinguish but harmonics can be used to separate them.

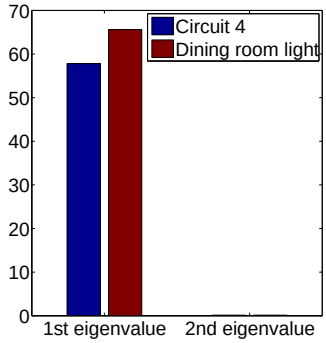
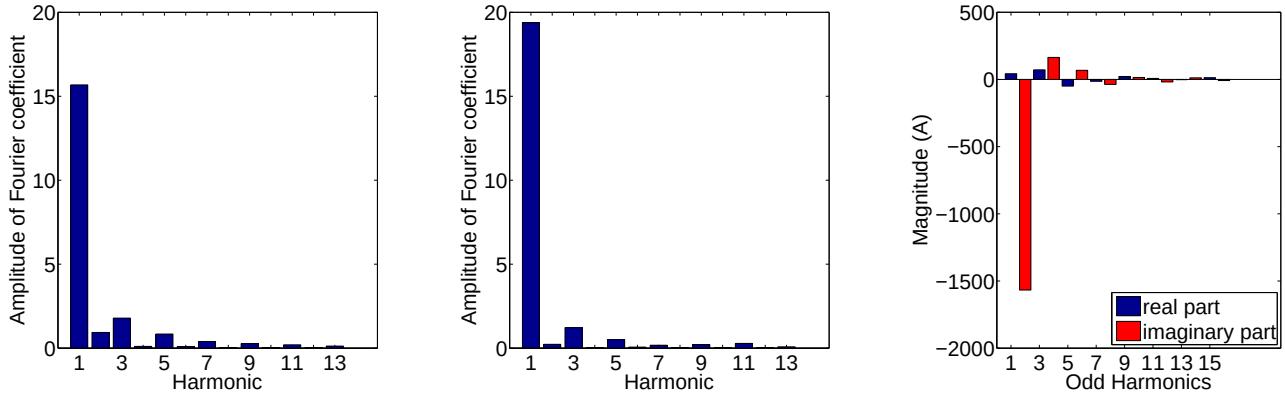


Figure 2.13: The eigenvalue of a circuit and dining room light.



(a) (b) (c)

Figure 2.14: Harmonics Feature of (a) a refrigerator and (b) an air compressor (c) real and imaginary part of odd number of harmonics of a refrigerator.

[80, 81] discovered that any VSD generates a unique high harmonic power, which is identical among devices and effective for disaggregation. Applying Gaussian random process to the power usage of VSD, the kth apparent harmonic power, is calculated as $A_k = \sqrt{P_k^2 + Q_k^2}$, where k is an order number of harmonics. The correlation pattern between the real power and the kth apparent harmonic power is detected as a characteristic of each VSD.

The real and imaginary parts of harmonics are also fully used as features. Usually we only use the odd harmonics. The real part is calculated as $x_n = I_{(\frac{n+1}{2})} cos\theta_{(\frac{n+1}{2})}$ when n is odd; the imaginary part is calculated as $x_n = I_{\frac{n}{2}} sin\theta_{\frac{n}{2}}$, for n as even numbers, where I_n is the magnitude of the nth current harmonics and θ_n is the phase angle of the nth current harmonics. Figure 2.14 (c) shows the real part and imaginary part of the odd number of harmonics of a refrigerator. Also, [119] gives an example of how to separate devices by the real and imaginary part of harmonics.

A variant of harmonics is spectral envelope. It is a short-time average of harmonics and was

proposed as a device feature in [76, 84]. Further, harmonics can be used in conjunction with other features to disaggregate devices. [131] introduces a switching-function to identify Variable Speed Devices. Figure 2.15 gives a comparison example of current before and after rectifier and

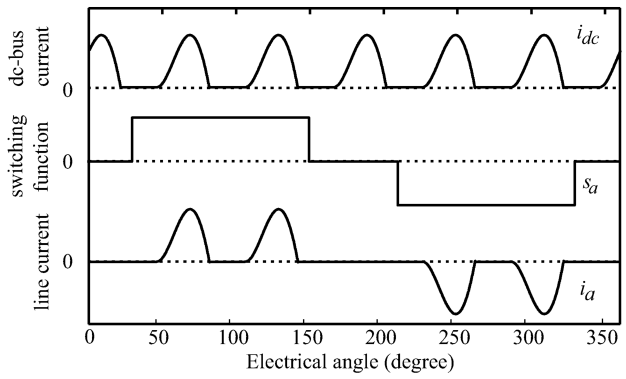


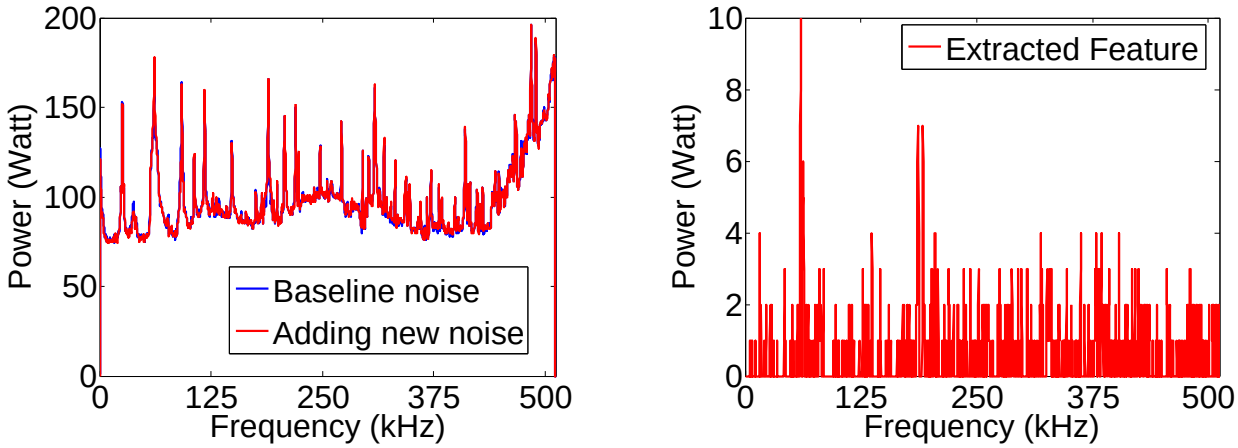
Figure 2.15: Switching-function for VSDs disaggregation (Courtesy:[131]).

inverter of VSDs. The operations of rectifier of a VSD corresponds to a switching-function. The current is initially direct current I_{dc} and after the current goes through the switching process, then it changes to I_a. The relationship between these two currents is captured as $I_a(\theta) = S_a(\theta)I_{dc}(\theta)$, where $S_a(\theta)$ is the switching function. This switching function can be represented as a linear combination of different harmonics: $S_a(\theta) = S_0 + \sum_n (S_n^p \sin n\theta - S_n^q \cos n\theta)$, where n is the harmonic number, S_0 is the DC component, and the variables S_n^p and S_n^q are the magnitudes of the in-phase and quadrature parts of nth harmonics of the switching function. By comparing the Fourier coefficients with the S_n^p and S_n^q of harmonic coefficients, these VSDs can be recognized.

Noise data: [106] and [50] recorded noise data instead of the current or voltage data. Interestingly, this noise data which occurs during switching on or off, can be used to identify devices because different devices have different noise signatures. The frequency of the noise data is also treated as a feature. [106] first detects that noise is generated when a device in a residential building turns on or off, or during the on state. With the introduction of switch mode power supplies (SMPS), the EMI generated by SMPS is also introduced as a feature [50]. Figure 2.16 (a) depicts the frequency generated by an LCD TV's on and off events in the spectrum of frequency in the Kaggle dataset [20]. Figure 2.16 (a) illustrates the noise time series background in red and the noise time series with newly added noise in blue. Figure 2.16 (b) shows that the newly added noise is segmented. By analyzing the mean value and standard deviation of the segmented noise, SMPS devices can be identified.

2.4.2 Features Beyond Current and Voltage

Duration or time of use The operation of a device may conform to some routines when it's turned on or off, and how long it's on or off. The time of a device usage involves the month of the year,



(a) (b)

Figure 2.16: (a) Baseline noise with newly added noise. (b) Noise Feature of a device.

day of the month, day of week, time of the day, and season of the year. Generally, fans and air-conditioners work in the summer and heaters work in the winter. Figure 2.17 shows that the total power usage of a commercial building [115]. Power usage is pretty high during the week day. In contrast, the power usage on weekends is pretty low.

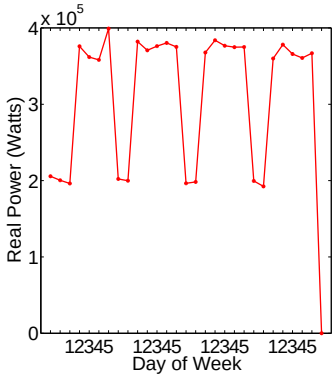


Figure 2.17: Day of Week of Feature.

The date, time, and week features have been used in previous work. For example, [65] points out that a laptop is often powered-on in the morning of workdays and TV works in the evenings and weekends. Also, [65] analyze the on/off distribution of devices and finds that they conform to Gamma distribution.

Device correlation Some devices may need to operate with other devices. For example, the TV and XBOX usually turn on and off together. The correlation between devices is first analyzed and integrated with HMM-based model [65]. This paper calculates all the correlation coefficients and shows that both the correlation between TV and stereo and the correlation between XBox and TV were as strong as 0.8. [28] combines the features of on/off time, duration, and the correlations among devices to mine for usage patterns. It is designed especially for the correlations between different types of devices other than the devices of the same type. In the continuing work, [27] develops an algorithm CoMiner to discover the correlation between devices.

EMF, sound and light The electromagnetic field (EMF) is generated by electrical devices when they are powered on. Different devices can produce different EMF. EMF can be monitored by an EMF sensor around devices. It is used in [47] to detect on/off state changes of devices.

In addition, the sound and light made by the devices can help judge the status of devices as described in [66]. This work monitors the sound by the acoustic sensor at 4Hz sampling rate to identify the compressor status of a refrigerator. Also, the light intensity installed in a refrigerator reflects whether the door is open or closed.

Weather Weather is a key factor for a commercial building's power usage because of the variation in the usage of the HVAC system in the summer and winter. The correlation between temperature and electricity usage is studied in [104]. This work shows that when the temperature is higher, the daily electricity usage increases.

Other possible features Additional sensors, such as motion sensors [32], can determine activity inside a building, and provide useful input to a disaggregation model. [121] integrates data from electricity and other sensors at home to disaggregate low-power devices such as lightbulbs. [132] proposes using plug-in low-cost outlet sensors to help capture appliance state. Also, [128] uses limited plug-level sensors to improve the disaggregation accuracy. In the future, more sensors are likely to be available and could be leveraged for energy disaggregation.

2.5 Disaggregation Algorithms

In most cases before energy disaggregation algorithms can be applied, a pre-processing phase, which comprises its own set of algorithms, is necessary. It extracts features or transforms data from one domain to another, for example time to frequency. In this section, we will first introduce pre-processing algorithms, give an overview of all disaggregation algorithms (supervised, unsupervised, semi-supervised), and explain the algorithms' advantages and disadvantages. We also show computational cost.

2.5.1 Pre-processing: Event Types and Feature Extraction Algorithms

Event types of device features

While conducting energy disaggregation, the events in a time series play an important part in identifying devices. These events can be classified into two categories viz. *point-event* or *vector event*.

1. *Point Event*. A point event is defined as an event that is determined by a single value from the input dataset and this event is used to characterize a device. For instance, examples of point events include real power, reactive power, or power noise at any particular time.

2. *Vector Event*. A vector event is defined as an event that is composed of several data points that characterize a device, instead of just one data point. For instance, the waveform of current or voltage is a vector event as it spans a period of time and cannot be captured by just one data point. When using vector events, if the raw data is transformed from the time domain to the frequency domain, the extracted features are also treated as vector events. It a is common practice for current and voltage to be transformed into harmonics and wavelets. These transformations' representations as vector events better capture the data and provide more useful information.

Feature extraction algorithms

Point events are easily obtained from the raw dataset which, for example, provides real power or reactive power time series. A commonly used event is obtained from the difference between two successive data points. If the difference is significant (above a threshold), a point event is generated. Otherwise, no event is recorded.

Since high frequency datasets provide rich information, it is quite common to extract vector events from them. It is done in the following manner. After pre-processing a time series from the high frequency data, we can obtain several features like the startup of current, current waveform, harmonics of current, current transformation, eigenvalue of current, voltage waveform, or voltage noises. Generally these pre-processing algorithms are classified into three types: *1) basic signal processing, 2) Fourier transform, and 3) wavelet transform.* Basic signal processing is used to filter, shift or amplify a time series data. Fourier transform converts the time series data from time domain to frequency domain, and harmonics are acquired from the result of the Fourier transformation. Wavelet transformation divides a time series into different scale components and each component is assigned to a frequency range. The features extracted by the wavelet transform are relatively stable. For instance, when a device is turned on, a corresponding sharp peak is generated in the time domain which is not characteristic of the device's power signature. But the use of wavelet transformation in the frequency domain will discard this transient information and be more consistent with the device's actual power signature.

Table 2.1: Pre-processing Algorithms on Vector Event Feature Extraction.

Feature-identification Algorithm	Startup of I_{AC}	Harmonics of $\|I_{AC}\|$	I_{AC}	I_{AC} transformation	eigenvalues of I_{AC}	V_{AC}	voltage noise
Signal Processing[34]						$\surd$	
Fourier Transform [131]		$\surd$	$\surd$				
Wavelet Transforms[22]			$\surd$		$\surd$		

Table.2.1 gives examples of several vector events using these three types of pre-processing algorithms. Basic signal processing steps, low-pass filter, amplification and shifting are used to extract the distorted voltage shapes in [34]. In [131], the Fourier transform of the waveform is applied to find distinct coefficients for those VSDs. The Fourier transform is also used in [59] to extract the harmonics features. Wavelet transform is employed to identify harmonics features of devices in [22].

2.5.2 Overview of Disaggregation Algorithms

As the number and complexity of features used have increased, so has the complexity of the disaggregation algorithms. From a data mining or machine learning perspective, the algorithms are sorted into three categories: supervised, unsupervised, and semi-supervised. Table 2.2 lists the three categories.

From an events perspective, algorithms can be point-based, event-based or a combination of these two. Point-based algorithms are dedicated to the processing of turning on and off events of devices. Vector-based algorithms treat the current, voltage or power value as ordered time series instead of picking up a single transition states. Point-based algorithms perform well on discrete steady-state devices, but perform poorly on the devices with vector features, such as the variable speed devices (VSDs).

Comparison of Supervised and Unsupervised Algorithms We summarize the merits and shortcomings of supervised and unsupervised learning algorithms for disaggregation from the perspective of installation cost of meters, dataset size requirement for building models, computational cost of operation process, and accuracy results of disaggregation.

Compared to unsupervised learning approaches, the *advantages* of supervised learning algorithms are as follows:

1. The disaggregation accuracy is higher when using the same dataset as input.
2. They require less data set to build a disaggregation model.
3. Once the model is trained, they usually have faster operation to obtain the output with same input.

Compared to unsupervised learning algorithm, the main *disadvantages* of supervised learning techniques lie in that the labelled data of each device is hard to get because the cost would be very high if installing meters to monitor each device.

2.5.3 Supervised Learning Algorithms

Classification-Based

Supervised learning based energy disaggregation algorithms focus on distinguishing devices from aggregated data by treating the problem as one of device classification. These classification algorithms include simple pair-wise match, rule-based algorithm, SVM, Kernel Based Subspace Classification, Bayesian classifier, neural network, genetic algorithm, dynamic bayesian network, sparse coding, AdaBoost, decision tree, a combination of SVM and AdaBoost etc.

Classification-based energy disaggregation operate under the following general assumption:

Assumption: *A classifier that can distinguish devices can be learned in the given feature space.*

Since there are more than one device inside a building or house, this is actually a *multiclass* classification problem.

Neural network A basic energy disaggregation technique utilizing a neural network is implemented in two steps. First in the training stage, a neural network is trained to learn several features of multiple devices. Second in the test stage, each feature extracted from the aggregated data is provided as an input to the neural network. If the neural network recognizes the input by associating it with one of the features learned in the training phase, then the device that generated that input is classified accordingly.

A neural network example is illustrated in Figure 2.18. There are d features in the input and M number of devices on the output. The neural network defines K hidden states.

Generally the evaluation for a neural network based classifier is to compare the relative error percentage.

Roos et al. initially proposed to adopt neural network for classification based on real power and reactive power by transforming the aggregated data into images for processing [111]. Next, [10] employs backpropagation (BP) neural networks with attributes as number of states, duration time and average energy consumption. Furthermore, the training stage of [40] and [119] is based on

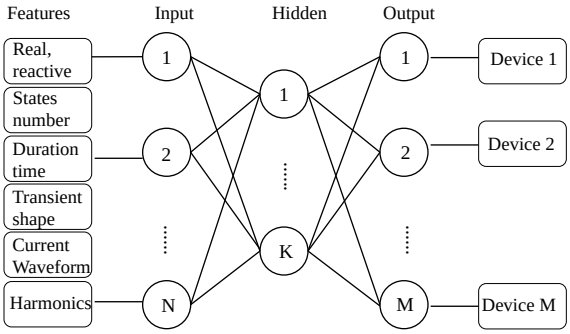


Figure 2.18: Neural Network Approach for Energy Disaggregation.

the current waveform and harmonics. The latter paper treats eight odd-numbered harmonics as a vector event feature for classification and chooses 16 hidden nodes. Note that [119] compares several neural network approaches, namely multilayer perception (MLP), radial basis function (RBF) network, support vector machine (SVM) with linear and polynomial and RBF kernels. The results indicate that these MLP and RBF-based approaches have high classification accuracy.

Chang et al. extends the back propagation approach by employing electromagnetic transient program (EMTP) with transient real power when devices start up [25]. The transient shape is a vector event rather than a point event. In order to identify devices from aggregated data adaptively, a window size w is adopted to enhance the algorithm as an adaptive neural network. Initially, the differential values $dP_{transient}$ for a period time w represent the characteristics of a class of devices. During the training process, the time period value w increases by δ from 1 to w. The δ which achieves highest recognition accuracy is retained.

Recently basic adaptive neural network (ANN) applied to energy disaggregation was presented in [87]. The paper selects backpropagation ANN (BPANN) to train a model. Based on a combination of features, such as real, reactive power, transient shapes, harmonics, eigenvalue of current waveform, voltage waveform, etc., this paper establishes a committee decision system based on three rules: most common occurrence (MCO), least unified residue (LUR), and maximum-likelihood estimation (MLE), to classify devices. As a result, the disaggregation accuracy is high.

Two variants of the basic neural network were proposed. [134] classified the transient events by way of back propagation (BP) and and [24] adopted learning vector quantization (LVQ) to recognize devices. Neural networks have also been combined with other approaches. For example, [23] combines multi-layer feed-forward neural network with genetic algorithm to to analyze the device turn-on transient signatures.

Support vector machine SVMs attack the problem using multiclass learning techniques by learning the event features only from the training data as opposed to unsupervised methods that learn features from the entire dataset. Kernels such as radial basis function (RBF) kernel are adopted to learn complex features such as harmonics. At first, SVMs were employed to classify devices by 13

odd-order harmonic current and phase angles from on and off events [103]. Later, a kernel based subspace classification (KSC) approach is used for events classification in SVM [100].

SVM was widely utilized with noisy datasets although it do not scale well for large data sets. SVM was adopted by [106] to classify transient pulses noise from various homes. In [46], SVM is applied to transient and continuous voltage noise data. Noisy voltage generally produced by devices influences the power wiring. According to Gen Marubayashi [93], there are three types of voltage noise: on-off transient noise, steady-state line voltage noise which is produced at 60 Hz or integer times of 60 Hz (e.g. harmonics), and steady-state continuous noise which is generated beyond 60 Hz. Voltage noise data are sampled with very high frequency. During pre-processing, the noisy recorded voltage data is transformed by a Fourier analysis. Then, three to five transient voltage noise signatures are labeled and a threshold is pre-defined. During the training phase, by sliding a window on the aggregated voltage noise, a part of the data with continuous voltage noise is extracted and compared with the pre-stored voltage noise data by measuring the Euclidean distance. If the distance is larger than the pre-defined threshold, then the feature vector is exerted from this window. After sliding over aggregated voltage noise data, all these feature vectors are classified by the SVM.

Besides used by themselves as a standalone classifier, SVM is also utilized in energy disaggregation by combining with other approaches. In [101], both stand-alone SVM and combination of SVM and radial basis function network (RBFN) are implemented to compare the disaggregated data with the ground truth harmonics.

Bayesian network A combination of SVM and Dynamic Bayesian Network was demonstrated in [46]. Initially a threshold value for the Euclidean distance between new data and basic noise data is predefined. Then a window slides on to determine whether the distance exceeds the threshold. According to the Euclidean distance, the feature vectors which characterize the devices are classified by the SVM. Then the Dynamic Bayesian Network is utilized to classify the devices based on prior information, such as washing machines, dryers, and HVAC.

Rule based Rule based algorithms use the different operating rules of the various devices to solve the classification problem. The training dataset comprises of various rules that describe the operation of a device. If a test event presents one of these rules, the device that produces that event is classified accordingly. Rule based techniques have been primarily used in multiple-state devices.

Closure-rules with maximal length of four for real power with transition states was used by [73] to classify devices. The principle of closure rules is that if only one device change its state, the baseline signature is the same before the occurrence of the state change. Rules of many devices can become complex for each device if the vector events feature is introduced.

Rule mining is also proposed in [110]. The first step is to identify candidate rules. For each time slot of an hour, a co-occurrence matrix is derived by detecting the device states. Through this, we know when the devices are probably turned on for each hour of a day and each day of a week. In the second step, those significant rules are chosen by a *JMeasure*. And only those rules with values greater than 0.01 are selected.

Naive Bayes classifier

Algorithms that use the Naive Bayes classifier (NBC) was proposed in [136] to distinguish devices. Based on that approach, [139] uses power and time as features to automatically disaggregate the major residential electronic devices.

AdaBoost, decision tree [16] tests with four approaches: k-nearest neighbor, Gaussian naive Bayes (GNB), decision trees (DT) and multi-class AdaBoost (MultiBoost) for high frequency data.

[103] integrates SVM with AdaBoost to classify devices based on odd-number harmonics. In this case, AdaBoost helps SVM to classify those unclear points. Suppose in support vector machine, the margin Q is defined as

$$Q = \min_{i=1,...,l} \rho(z_i, f) \tag{2.10}$$

where

$$\rho(z_i, f) = y_i f(x_i) \tag{2.11}$$

AdaBoost is used to minimize the margin $\rho(z_i, \alpha) := \rho(z_i, f_\alpha)$ on the training set

$$\mathscr{G}(a) = \sum_{i=1}^{l} exp\{-\|\alpha\|_1(\rho(z_i, \alpha) - \phi)\} \tag{2.12}$$

To achieve this goal, every example z_i is given a weight $w^t(z_i)$. Applying bootstrap on the weighted sample distribution, we can find α_t to minimize $\mathscr{G}(\alpha)$, where $t = 1, ..., T$.

Computational Complexity The computational complexity is a function of the classification approach used. Kearns [62] presents a comprehensive discussion on this matter. Generally for training, decision trees tend to be faster than techniques which requires quadratic optimization such as SVMs. The testing phase is usually very fast Real power is a uni-dimensional feature and real reactive power is a two dimensional feature. If harmonics, waveform and wavelet are introduced, the feature becomes multi-dimensional. The computation time and complexity increases with higher dimensionality. Neural network has the advantage over detecting interactions between the disaggregated features and output time series data but it works very slow.

Nearest Neighbor-Based

Several energy disaggregation algorithms have been designed using the nearest neighbor (NN) techniques have been used in energy. These techniques generally make the the following assumption:

Assumption: *Feature instances from the same device occur in dense neighborhoods, whereas different device feature instances occur further away from their nearest neighbors.*

For all these NN techniques, obviously, a distance or similarity measure between two instances must be defined in order to perform device classification. There are different ways to compute the distance (or similarity) between two data instances. For single feature disaggregation, viz.

point event or vector event, Euclidean distance is a common choice [50]. For multiple features disaggregation, that is, several point events or vector events, the distance between two instances is computed as the Euclidian distance across the dimensions of the vector event as in [116].

Nearest neighbor based energy disaggregation techniques can be grouped into two categories:

1. Techniques that use the distance of a data instance to its k^{th} nearest neighbor as the measurement.
2. Techniques that use the relative density of each data instance as the measurement.

Using distance to k^{th} nearest neighbor The basic nearest neighbor technique has been applied to detect the multiple feature such as transient power shape [16, 17, 81, 116].

[116] describes how transient shapes of power consumed by devices over time are discovered. Transient shapes exemplar for each device are summarized and recorded in form of real and reactive power P-Q by analyzing the data from each device. A pre-defined window size of 100 data points is used. As the aggregated data flow comes, consequently the data points in each window is compared with the pre-stored exemplar. If the Euclidean distance is smaller than the pre-defined threshold, an event is said to have occurred in this window and it matches a pre-stored exemplar. Based on this grouping, [117] decomposes the real power transient shape into two vectors, shape vector and time vector instead of setting the whole transient shape as a device feature. Figure 2.19 depicts an exemplar with two shape vectors s_1 and s_2.

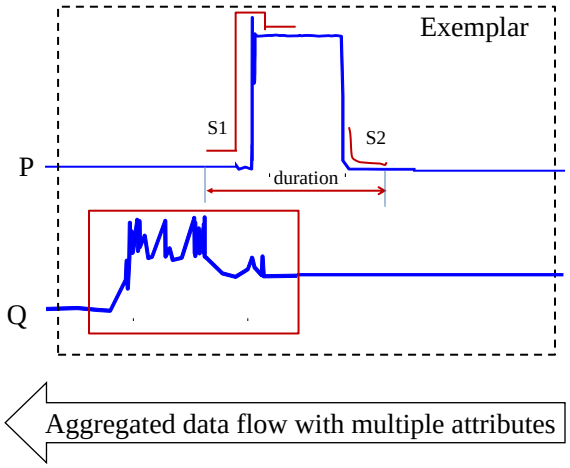


Figure 2.19: Transient Shape Decomposition and KNN Seach

To identify which device the disaggregate signature belongs to, it is to compare with this exemplar by least square criteria. After that, a similar exemplar comparison approach is applied to identify devices. The advantage of real power shape decomposition is that when comparing the transient shapes, only some characteristic parts are needed rather than entire transient shapes in the data.

This helps cut computational cost for the exemplar comparison phase. Although this paper doesn't mention K-NN algorithm explicitly, the description in this paper exactly matches what KNN algorithms do to search for closest shapes.

A variant of the KNN approach measures the Euclidean distance with inverse weighting [50]. The variant KNN is employed to identify devices with switch mode power supplies (SMPS) that have power line noise features. These power baseline noise signatures of each device are stored as vectors and 8dB is set as the power threshold above the noise baseline. In order to classify events from aggregated data into 25 noise events corresponding to 25 devices, a window is set to calculate the difference vector. After a new event is added on a particular power line, the distance between the vector of the newly-added event and baseline noise vector is calculated. If there is a peak above the pre-defined threshold, a Gaussian function is applied to calculate the mean, standard deviation of the difference vector.

Another variant KNN, discussed in [81], identifies variable speed devices (VSDs). It builds a table to store the real power, reactive power and harmonics for each device. Then the signatures extracted from the aggregated power are compared with the stored features. The disaggregated signature is assigned to the device, whose feature is most similar to the stored feature. Since this process essentially replicates the K-nearest neighbor mechanism, [81] is classified into KNN category.

Using relative density Techniques that estimate density of the neighborhood of each data instance are also popular in device classification. The classification is based on whether the instance lies in a neighborhood of high or low density. If an instance lies in a neighborhood with high density, it is declared to be in the device group corresponding to that neighborhood.

Given an instance as a center, circles with varying radii are drawn around it. The distance to its k^{th} nearest neighbor is equivalent to the radius of a hyper-sphere. In a probability density graph, this distance represents the inverse of the dataset's density [72]. Real power probability density function is used as a feature to classify two-state devices in [138]. The number of device is indexed by the power as shown in Figure 2.20.

In the training step, the real power probability density function of each device is obtained by analyzing each device's actual power consumption. In the classifying phase, the negative values are first clustered and the mth cluster represents device m. Then the positive values are clustered to match the negative clusters. The real-power probability density function is used to match the negative values to their corresponding positive counterparts.

Computational Complexity

A drawback of basic nearest neighbor approaches is that the time complexity is $O(N^2)$. If multiple attributes are employed with window size w instead of only real power, the computation cost is even higher than $O(N^2)$.

Advantages and Disadvantages of Nearest Neighbor Based Techniques

The *advantage* of nearest neighbor based devices classification is that it's straight-forward and

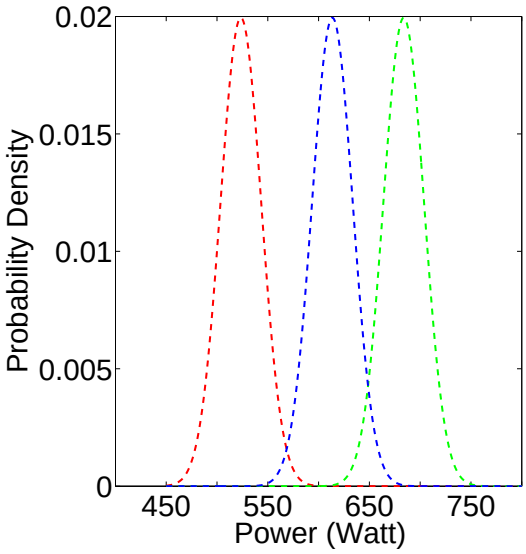


Figure 2.20: PDFs of three neighboring by power draw appliances.

primarily requires a proper distance measure for the given features.

The *disadvantages* of nearest neighbor based devices classification techniques are as follows:

1. The computational complexity at the test stage is high, especially for the high frequency data with vector features. The algorithms require comparison between all device features at each window from the aggregated data to obtain the nearest instance.
2. When multiple features are applied, the definition of distance measure becomes challenging because different features have different units of distance.

Statistical Model-Based

Statistical approaches to devices classification assume that *a device instance belongs to a high probability region of a stochastic model, while not belonging to a region at low probability.*

Statistical techniques fit a stochastic model given the event features from all devices. A statistical inference test is applied to determine whether an unseen event extracted from the aggregated data belongs to this model. Instances with low probability generated from the learnt model are declared as wrong event classification. Both parametric and non-parametric techniques are used to fit a statistical model. Parametric techniques assume that the underlying distribution of events are known whereas non-parametric techniques posits that this underlying distribution is unknown.

Parametric models As mentioned before, parametric techniques assume that the device's features follow a parametric distribution with parameter θ and probability density function $F(y, \theta)$, where

y is the observation. The score of a test instance is the inverse of the probability density function.

An alternate approach is the hypothesis test. The $null$ hypothesis (H_0) is that a test instance x has been generated using the estimated distribution with θ. If the statistical test rejects H_0, x is declared to not belong to this device's distribution.

[59, 60] use Goodness-of-fit (GOF) Chi-squared to detect the transient events generated by the first harmonics of power consumption. GOF utilizes the hypothesis approach. At first, a change point in time series data is detected. For i independent and identically distributed (iid) data points $y_t, t = 1, 2, ..., T$ drawn from a distribution $G(y)$ and the supposed distribution $F(y)$. The binary hypothesis testing problem is defined as

$$H_1 : G(y) \neq F(y) \tag{2.13}$$

$$H_0 : G(y) = F(y) \tag{2.14}$$

Then the χ^2 test for goodness-of-fit(GOF) is defined. If the χ^2 hypothesis condition is satisfied, then the feature is classified into the supposed device.

Generalized likelihood ratio is applied in [8], [18] and [89]. They use the generalized likelihood ratio to classify the events generated by different devices.

First, the mean power value before and after a time t is calculated. Given the aggregated data, the log ratio of probability distribution before and after each event is calculated as follows.

$$R = \prod_{t=j}^{k} \frac{F_{u_t}(y_t)}{F_{u_{t-1}}(y_t)} \tag{2.15}$$

where y_t is the sampled variable at time t, u_t is the mean value of the sampled sequence at time t, and $F_u(y_t)$ is the probability density function of the sampled sequence $y_t(t = j...k)$ about the mean value u. The greater the probability, the data points belongs to a specified device.

Non-parametric models The non-parametric techniques in this category does not define a prior assumption such as smoothness of density, etc. The model is driven directly by the data.

A hierarchical probabilistic is proposed in [130]. It aims to find devices with multiple states. It utilizes the device-on distribution and real power features. In the hierarchical probabilistic model, a three layered model is applied. The first layer is a feature layer, the second layer is a state layer, and the last layer is a consumption layer. The objective function estimates the Maximum-a-Posteriori (MAP) that an event belongs to a device. Since the computation cost is high, it utilizes a heuristic approach.

Computational Complexity The computational complexity of statistical techniques depends on the nature of the fitted statistical model. Fitting single parametric distributions from the exponential family, e.g. Gaussian, is linear in data size as well as number of attributes. Fitting complex distributions such as Gamma distribution [65] using iterative estimation techniques such as expectation maximization (EM) are typically linear per iteration though they might be slow in converging depending on the problem and convergence criterion.

Advantages and Disadvantages of Statistical Model-Based Techniques The *advantage* of statistical techniques is :

1. If the assumption regarding the underlying data distribution holds true, statistical techniques provide a sound device classification.

The *disadvantages* of statistical techniques are as below:

1. The device classification primarily relies on the assumption that the data is generated from a particular distribution. But this assumption often does not hold true, especially for multiple-state devices.
2. Even if the distribution assumption is true, there are several hypothesis tests for devices classification. It is difficult to choose a proper hypothesis test when dealing with a complex distribution.

Optimization-Based

There are several techniques that cast device classification as an optimization problem. In this formulation, energy disaggregation is specified as an objective function that minimizes the error.

Dynamic programming

[11] utillizes mathematical dynamic programming with the genetic algorithm to find the multiple-state device that are represented as finite state machines (FSM). In this paper, the genetic algorithm is integrated with clustering and dynamic programming as an approach to solve the devices classification problem.The whole procedure is broken down into four steps. In the initial step, a finite state machine is used to describe the real power change events for each device. The real power change events are detected from the aggregated data. All the on and off events shown in the time series are represented as $\Delta y_t = y_t - y_{t-1}$. In the second step, fuzzy clustering is used to cluster all the detected real power change events. In the third step, all finite state machines are created by a genetic algorithm. At the final stage, dynamic programming is applied to discover the shortest path in those finite state machines.

The qualification of disaggregated finite state machines is evaluated as follows. Shannons entropy is introduced to compare the shortest path to the pre-stored path of finite state machines. Assume a shortest path $\Gamma_l = S_{l1}, ..., S_{lk}$ and a device's finite state machine path $\Gamma_m = S_{m1}, ..., S_{mk}$, the Shannon entropy is calculated as Equation (2.16).

$$Q_l = -\sum_i \Delta e_i log|\Delta e_i| \tag{2.16}$$

and where $\Delta e_i = |\frac{\sigma_i(\Gamma_l) - \sigma_i(\Gamma_m)}{\sigma_i(\Gamma_m)}| + e_0$ and σ_i represents either ON duration between state changes or real power standard deviation of the state S_i. The shortest path with least entropy belongs to

device m which has the characteristics of state machine Γ_m. This genetic programming based optimization approach is applied to the features of three current and voltage features.

In [23], genetic programming is integrated with the neural network to identify devices. In [129] clustering is integrated with finite state machine and dynamic programming to disaggregate the devices in real time with low cost.

Dynamic model [38] assumes that each device has an input and an output then applies a dynamic approach to simulate the disaggregation process. Each device is represented as linear time-variant state-space model over the entire time series. The problem is formalized as Equation (2.17).

$$\begin{aligned} &\underset{\hat{y},x}{\operatorname{argmin}} \mathcal{L}(\hat{y}, y) + g(x) \\ &s.t. \hat{y_m} = h_m(x_m) \\ &\hat{y} = \sum_{m=1}^{M} \hat{y_m} \end{aligned} \tag{2.17}$$

where $m \in 1, ..., M$, M is the number of devices, x_m is the input to the device m, h_m is a function which denotes the underlying dynamics. To estimate $x[\cdot]$, blind system identification techniques [2] are used. **Integer programming** Integer programming [124] is applied to the current waveform in a supervised learning setting. Each device's waveform which spans T, where T is 1/50 or 1/60 seconds. is stored in the database, then a disaggregation process moves on to identify the devices according to the pre-stored current waveform. This paper supposes there are N kind of devices, and there are C_n appliances for each kind of device.

Suppose there is an aggregated load *current* y,

$$y_t = \sum_{m=1}^{M} c_m(s_m) x_t^{(m)} + \epsilon \tag{2.18}$$

where $c_m \in \{0, ..., C_m\}$ is integer variable for $m \in \{1, ..., M\}$, $t \in \{1, ..., T\}$, $x_t^{(m)}$ represents the current of m kinds of devices at time t, M denotes the number of device types, $c_m(s_m)$ is the operation states of one appliance c_m belong to kind m if eace device has only one operating mode.

ϵ represents noise. To estimate c_m from the aggregated y_t, then this problem is abstracted as an integer quadratic programming problem

$$\min \sum_{t=0}^{T-1} (y_t - \sum_{m=1}^{M} c_m(s) x_t^{(m)})^2 \tag{2.19}$$

subject to

$$c_m \in \mathcal{Z}, 0 \leq c_m \leq C_m, \forall m \in \{1, ..., M\}.$$

Viterbi algorithm Another paper [136] employs real power probability density functions (PDFs) by a conjunction of Semi-Markov and Viterbi-type algorithms to distinguish devices. The standard Viterbi algorithm is used to maximize the likelihood of power draws of appliance m and its neighbors.

$$\{\hat{S}_t\} = argmax_{s_t}[\{S_t\}|\{\omega_t\}] \tag{2.20}$$

Where $\{S_t\}$ is the state sequence and $\{\omega_t\}$ is the transition observations.

It adopts a similar approach to the one mentioned in [12]. The difference of this method is that they introduce the probability density function of real power of each device.

Sparse coding [70] introduces non-negative sparse coding to solve the energy disaggregation problem. It is composed of three major steps. The first step is the sparse-coding pre training step and it aims to model each source using nonnegative sparse coding by solving Equation (2.21).

$$\min_{A_m \geq 0, B_m \geq 0} \frac{1}{2}\|X_m - B_m A_m\|_F^2 + \lambda \sum_{p=1,q=1}^{r,s} E(A_m)_{pq} \tag{2.21}$$

such that $A_m \in R_+^{r \times s}$ and $B_m \in R_+^{T \times r}$, where $X_m \in R^{T \times s}$ represent the sth power level associated with device m. the columns of B_m represent r basic functions corresponding to features, the columns of A_m represent the activation, i.e. sparse codes of these basic functions set, λ represents the sparseness degree, and F denotes the Frobenius norm. This optimization is solved by a coordinate descent approach but without computing the bases of each model.

The second step is the discriminative disaggregation training step. It incorporate the aggregated Y in the bases $B_m, m = 1, .., M$.

$$\hat{A}_{1:M} = argmin_{A_{1:M}} \parallel Y - [B_1 ... B_M][A_1 ... A_M]^T \parallel_F^2 + \lambda \sum_{p=1,q=1,m=1}^{r,s,M} E(A_m)_{pq} \tag{2.22}$$

where M is the number of devices, $\hat{A}_1, ..., \hat{A}_M$ are the activations related to aggregated power. Each column of $Y \in R^{T \times s}$ represents the sth power consumption associated with the device m. The target of the sparse coding approach is to find the best $\hat{A}_m^*$. Therefore the difference between $\hat{A}_{1:M}$ and $A_{1:M}^*$ should be as small as possible.

To achieve this goal, a regularized disaggregation error is defined. $B_{1:M}$ is optimized at each iteration during discriminative training phase. Then in the same iteration, the base of $B_{1:M}$ is updated to calculate $\hat{A}_{1:M}$ again. By updating $A_{1:M}^\star$ and $B_{1:M}$ alternatively, the sparse code and the real power consumption of each device is calculated.

$$\hat{B}_{1:M} \leftarrow \hat{B}_{1:M} - \alpha((Y_{1:M} - \hat{B}_{1:M}\hat{A}_{1:M})\hat{A}_{1:M}^T - (Y_{1:M} - \hat{B}_{1:M}A_{1:M}^*){A_{1:M}^*}^T) \tag{2.23}$$

where α is the step size.

Note that sparse coding is also extended to water disaggregation [37].

Nonnegative tensor factorization [45] applies a nonnegative tensor factorization and compares it with nonnegative sparse coding. The power consumption of each device is represented as a tensor.

For each device, the power usage over a period of time T can be cast as a matrix factorization problem.

$$Y_t^{(m)} \approx \sum_{l=1}^{r} A_l S_t^{(l)} \tag{2.24}$$

where $S^{(l)}$ is the main features or power levels of each device and A_l is the corresponding activation, r is the number of bases used by sparse coding, and $t = 1, ..., T$.

Given the aggregated data, using a supervised learning approach, one can formulate the energy disaggregation as a nonnegative matrix factorization problem.

Furthermore, to solve this problem, [45] implements two solutions: one is based on nonnegative sparse coding, another is multidimensional representation and factorization method. Nonnegative sparse coding has been introduced in paper [70]. For tensor decomposition, this paper adopts the approach PARAFAC [69] with nonnegative constraints.

Computational Complexity The computational cost of dynamic programming for classifying devices is polynomial. Assume m is the number of FSMs, n is the number of "diff" data. The computational time cost of the dynamic programming step to classify devices is $O(mn)$ [30]. However, the computational cost of the whole procedure in [11] is higher because it contains the steps of fuzzy clustering and genetic programming.

[124] formulates the energy disaggregation problem as a linear integer programming problem. Therefore the computational cost is polynomial i.e. $O(TM)$, where T is the number of aggregated data in the form of current waveform and M is the number of devices. However, the total computational cost in [124] is relatively high because it utilizes the high frequency data with large data size.

The computational cost of viterbi algorithm is linear i.e. $O(T)$, where T is the number of aggregated power points [19].

The computational cost of sparse coding is high. Therefore the energy disaggregation is formulated as ℓ^1 minimization optimization problem. The computational cost decreases and becomes linear to the data points and number of devices $O(TM)$ [86], where T is the aggregated data points and M is the number of devices.

Advantages and Disadvantages of Optimization-based Techniques The *advantage* of optimization solution is as follows:

1. The device classification problem is formally proposed to minimize the error or entropy.
2. The solution for the optimization problems is straightforward.

The *disadvantage* of optimization-based technique is given below:

1. If more features are introduced such as harmonics, it's hard to formulate an optimization problem because the distance measurements of these features are non-uniform.

2.5.4 Unsupervised Learning Algorithms

When Hart initially proposed energy disaggregation, the problem was tailor made for unsupervised learning methods [52] because the exact information of individual circuits or devices is unknown. In recent times, unsupervised disaggregation has emerged as a hotbed for research. Clustering [48] is used to group similar events. Different approaches such as HMM [65, 71, 105] and temporal mining [114] have been applied. Clustering-based disaggregation algorithms are designed under the following assumption: *Events and features generated by a single device will be clustered together.* These techniques apply a known clustering algorithm to the data set and group events generated by a device. While clustering techniques have been designed with no knowledge on the number of devices, some unsupervised learning methods also assume that the number of devices is also known.

Hierarchical clustering-based

Gonccalves et al. proposed a method that disaggregates devices without a-priori knowledge of the total number of devices [48]. As the first step, in order to extract the real and reactive power features, blind source separation [83] is used. In the second step, hierarchical agglomerative clustering of real and reactive power is used to cluster the on and off events. The greedy matching pursuit (MP), which is a direct implementation of Hart's intuition, is calculated in terms of Euclidian distance $([P_t, Q_t] - [P_{closest}, Q_{closest}])$.

Computational Complexity [48] only studies disaggregating devices with on and off events. In this study, the real power and reactive power are used. The computational cost of the measurement of pair-wise distances is $O(T^2)$, where T is the number of points in aggregated time series. For agglomerative clustering, the computational cost of unsupervised disaggregation is $O(T^2)logT$ [58].

Advantages and Disadvantages of Clustering-based Unsupervised Learning Techniques

The *advantages* of clustering-based unsupervised techniques are as follows:

1. It is easy to set up the model even if the number of devices is not known.

The *disadvantages* of clustering-based techniques are as below:

1. Clustering-based technique may incorrectly group the devices with same power levels.
2. These techniques are applied to devices with two states, on and off, but not applicable to devices with multiple states.

FHMM-based

The factorial hidden semi-Markov model (FHMM) is a relatively new unsupervised energy disaggregation approach. It assumes that we know the number of devices inside a building and the power usage of the entire house is available. Kim et al proposes an FHMM technique and FHMM [65] to disaggregate devices in the manner described below. As shown in Figure 2.21 (a), FHMM uses multiple HMMs to model the status of each device. The aggregated power at a specific time is given by adding the values produced by the HMM corresponding to each device.

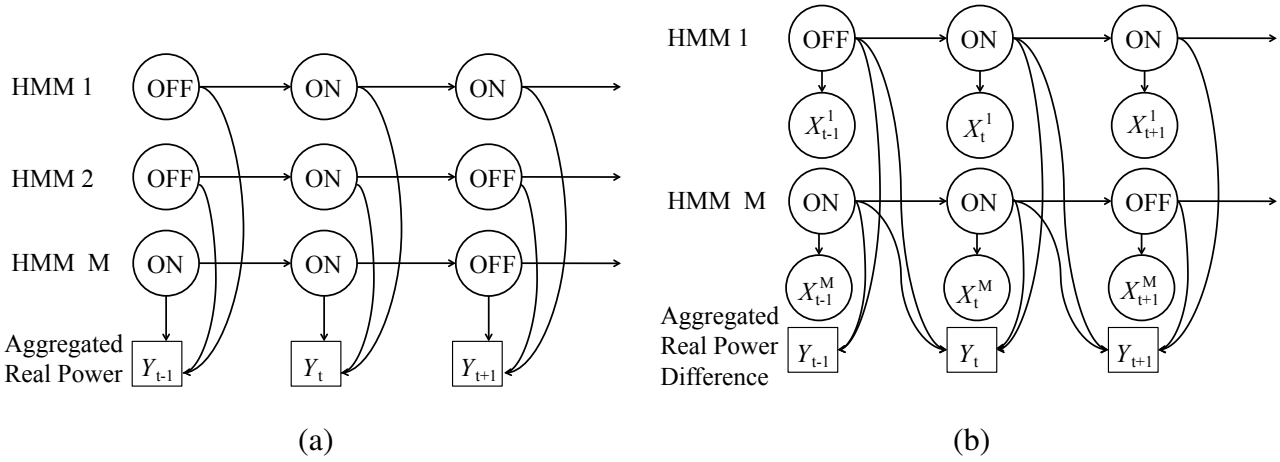


Figure 2.21: Graphical model with M devices. (a) FHMM and (b) Difference FHMM.

FHMM and constraint FHMM extend FHMM by incorporating the time duration for which the device is turned on, the correlation between various devices, and the usage time of each device.

We form the FHMM by calculating the initial probability $\phi_{in}(y, x|\Theta)$, emission probability $\phi_e(y, x|\Theta)$, and transition probability $\phi_t(y, x|\Theta)$, where Θ is the parameter set. The product of these three probability is given in Equation (2.25).

$$P(y, x|\Theta) = \phi_{in}(y, x|\Theta) \cdot \phi_e(y, x|\Theta) \cdot \phi_t(y, x|\Theta) \tag{2.25}$$

By maximizing Equation (2.26) with the EM algorithm, we can derive the HMM which represents the device.

$$\phi(\Theta, \Theta') = \sum_x P(y, x|\Theta') log P(y, x|\Theta) \tag{2.26}$$

where Θ' and Θ represent the previous and current iteration parameter set of the EM algorithm.

A variant of FHMM is the Additive Factorial Approximate Maximum a Posterior (AFAMAP) [71]. It is a mixture of the additive factorial model and difference FHMM model. The box diagram of AFAMAP is as Figure 2.22.

The disaggregation procedure comprises of the following four steps. Initially, the MAP is proposed

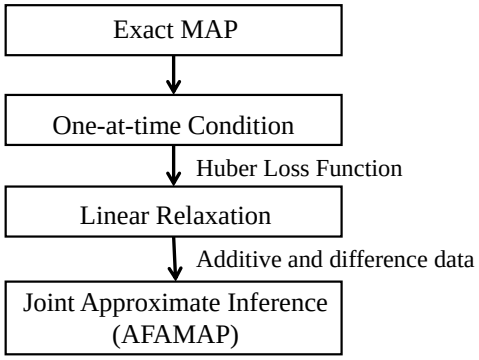


Figure 2.22: AFAMAP Flowchart.

and priors are defined as a Laplace prior given in Equation (2.27).

$$\begin{aligned} p(z_{1:T}) &= \frac{1}{Z(\theta,T)} exp\{-\theta \sum_{t=1}^{T-1} \|z_{t+1} - z_{t-1}\|_1\} \\ p(\Delta z_{1:T}) &= \frac{1}{Z(\theta,T)} exp\{-\theta \sum_{t=1}^{T} \|\Delta z_t\|_1\} \end{aligned} \tag{2.27}$$

where z_t is a introduced signal, and $\Delta z_t = z_{t+1} - z_{t-1}$. Thus the posterior of additive and difference model turns into a Gaussian distribution separately.

$$\begin{aligned} y_t | x_t^{(1:M)}, z_t &\sim \mathcal{N}(\sum_{m=1}^{M} \mu^{(m)}_{x_t^{(m)}} + \Sigma^{1/2} z_t, \Sigma) \\ \Delta y_t | x_{t-1}^{(1:M)}, \Delta z_t &\sim \mathcal{N}(\sum_{m=1}^{M} \Delta\mu^{(m)}_{x_t^{(m)}, x_{t-1}^{(m)}} + \Sigma^{1/2} \Delta z_t, \Sigma) \end{aligned} \tag{2.28}$$

where μ_j^m is the mean of the mth HMM for the state j, $x_t^{(m)} \in 1, ..., S_m$ denotes the state of the mth HMM at time t.

Then in the second step, the once-at-a-time constraints are added as in Equation (2.29) to limit that at any given time, only one device is turned on or off.

$$\mathcal{O} = \mathcal{Q} : \sum_{m,j,k \neq j} Q(x_{t-1}^{(m)}, x_t^{(m)})_{j,k} \leq 1 \tag{2.29}$$

Till this step, to solve the MAP, the computation cost is very high. In order to get a resolved solution, in the third step, the Huber loss function is employed to perform optimization by linear relaxation.

$$\begin{aligned} D(y,\theta) &= \min_z \{\|y - z\|_2^2 + \theta \|z\|_1\} \\ &= \sum_{\ell=1}^{n} \min\{\frac{1}{2} y_\ell^2, max\{\theta |y_\ell| - \frac{\theta^2}{2}, \frac{\theta^2}{2}\}\} \end{aligned} \tag{2.30}$$

Thus disaggregation is converted to a joint approximate inference AFAMAP problem. It's a convex quadratic program which can be solved by classical optimization algorithms. Then with aggregated data as input, we can get the M number of HMMs corresponding to M devices.

Another variant of FHMM was proposed in [105]. The difference FHMM is shown as Figure 2.21 (b). This method assumes that we know the labels of each device, thus meaning that the number of devices and device names are known. However, the power usage of each device is unknown. In the first step, the aggregated data is trained to get the features of each device. Since this training process only uses the aggregated data, we classify this approach into unsupervised disaggregation. During the procedure, the features are repeatedly deleted. Then more device features are gradually identified. In the next step, the appliance behavior like peaks arising from device being turned on or the power demand of the device, obtained from the previous step is used as a prior for the difference FHMM. Then the EM algorithm is used to evaluate the likelihood of whether the profile is of a certain device type.

$$accept(y_t, ..., y_{t+w}|\hat{\theta}) = \begin{cases} true & \text{if } \ln \mathcal{L} > \mathcal{D} \\ false & \text{otherwise} \end{cases}$$

where $y_t, ..., y_{t+w}$ represents the data in a window size w beginning from index t to $t+w$, $\mathcal{L}$ denotes the likelihood given the prior parameter $\hat{\theta}$, $\mathcal{D}$ is the predefined likelihood threshold. In the final step, all these devices are disaggregated by an extended viterbi algorithm.

Further, [55] uses HMM for electric heat usage disaggregation. HMM and AFAMAP are also run by additional applications [88].

Computational Complexity The computational cost varies for these two kinds of unsupervised learning approaches. Generally the computational cost of FHMM and its variants is exponential in the number of latent chains. Theoretically, the computational complexity is $O(MS^{2K})$, where M devices correspond to M chains, each device has S states, and K latent variables [19]. It's hard to obtain the direct solution theoretically. Therefore Gibbs sampling is applied to the first FHMM solution [65]. Later in the AFAMAP, QP problem techniques are used in the solution. In another variant of difference FHMM [105], the viterbi algorithm is applied.

Advantages and Disadvantages of FHMM-based Unsupervised Techniques

The *advantages* of FHMM-based unsupervised learning techniques are as follows:

1. It's the first formally proposed unsupervised learning approach.
2. It's solvable by introducing MCMC or converting it to an optimization problem.

The *disadvantages* of FHMM-based techniques are as below:

1. The computational cost is high.
2. The parameters obtained from the MCMC approach are not easy to estimate.

Temporal mining-based

A lightweight time series motif mining method [114] is proposed to identify devices rapidly. In this approach, a motif which represents a multiple-state device, is discovered in a time series of aggregated real power. Figure 2.23 illustrates how a motif is found. Non-overlap search for a single episode explains multiple-state changes for a device. A device turns on, then its state changes to another state, until it turns off. This episode corresponds to a complete running cycle of a device. A device may include multiple episodes. Between any two episodes, overlap does exist. For example, the second instance of Episode1 overlaps with the first instance of Episode 2. The overlap between episodes explains the operations of several devices. We regard Episode 1 as device A and Episode 2 as device B. When device B turns on for the first time, before it turns off, device A turns on for the second time then turns off, then device B turns off. Also, it can integrate with AFAMAP [71].

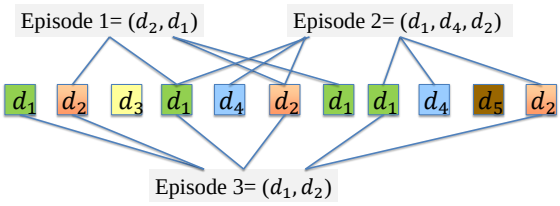


Figure 2.23: Motif Mining Example ([114]).

The output of motif mining can be used as the input of AFAMAP.

Computational Complexity Assume m is the number of power levels in the 'diffs' data. Then the computational complexity of DPGMM is $O(mnd^2 + md^3)$, where n is the number of points in diffs data, and d is the number of feature dimensions (e.g., time, date). The computational complexity for the episode generation step is $(p-1)O(m^2)$, where p is the maximal episodes length. Since p, which is 3, and m, which is 14 or 27, are small, we apply a brute force approach. The worst-case time complexity of the motif mining algorithm is $O(msq)$, where q is number of candidate episodes, and s is the size of the episode.

Advantages and Disadvantages of Temporal Mining-based Unsupervised Techniques

The *advantages* of temporal mining-based techniques are as follows:

1. It's a lightweight approach.
2. The disaggregation results are comparable to the results from complex models.
3. It's applied to multi-state devices.
4. It can capture device disaggregation even from commercial buildings.

The *disadvantages* of temporal mining-based techniques are as below:

1. The smoothing parameter is not adjusted automatically.
2. The problem is not formally proposed.

Probabilistic graph-based

Besides HMM, another probabilistic graph model was proposed by [63]. The model is composed of three layers. The component layer forms the the bottom most layer, the second layer comprises of a probabilistic graph model that captures appliances, and finally the top-most later us an inter-appliance layer. So far, this approach has not implemented in detail.

2.5.5 Semi-supervised Learning Algorithms

Semi-supervised algorithms assume that the feature for each device, such as the power levels of the device is already known. Instead of extracting features from the training data, it utilizes the features from the aggregated data using unsupervised algorithms. Then these features are used to predict devices from the test data.

Assumption: *The features are clustered based on the device i.e. all the known features that characterize a device are grouped together.*

Clustering-based

Lam et al. initially propose to utilize voltage-current (V-I) trajectory of appliance as a feature to perform clustering [75]. Hierarchical clustering are exploited to cluster the appliances by analyzing these V-I trajectories. When hierarchical clustering is employed, pairwise differences between V-I's shape features are calculated. Then a dendrogram is created to show the relationship between devices.

HMM-based

When FHMM is proposed by [65], it applies a semi-supervised learning model by integrating the duration when a device is turned on and off. Based on these durations, a semi-Markov model variant hierarchical dirichlet process hidden semi Markov model (HDP-HSMM) [61] is adopted by extending a Bayesian nonparametric approach to capture the duration distribution of each device.

Optimization-based

[133] proposes a contextual supervision approach to solve the single-channel source separation problem as an optimization problem. It uses the power levels and time of turning on and off for each device as features. then

$$\min_{x_1,\ldots,x_M,\theta_1,\ldots,\theta_M} \sum_{m=1}^{M} \{\ell_m(x_m, Z_m\theta_m) + g_m(x_m)\} \quad s.t. \sum_{m=1}^{M} x_m = y \tag{2.31}$$

Where ℓ_m and g_m are loss function and regularization term related to a device m. Choose these two as convex functions then the disaggregation problem transforms into an optimization problem. Note that different ℓ functions are chosen for different types of device. ℓ_1 norm is proper for sharp transition devices such as air conditioning. ℓ_2 loss is appropriate for groups of devices with smoother dynamics. When we use mean average error to evaluate the performance of the methods, the results show that contextually supervised approach performs better than the nonnegative sparse coding.

Advantages and Disadvantages of Semi-unsupervised Techniques The advantages and disadvantages of semi-supervised learning techniques are as follows.

The *advantages* of semi-supervised learning techniques are as follows:

1. It either learns features of each device by learning from some period's data or the feature of each device is given directly.
2. It can disaggregate the devices more accurately than unsupervised learning, which knows nothing about the exact features of each device.

The *disadvantages* of semi-supervised learning techniques are given below:

1. The existing features of each device are hard to be obtained.
2. The non-parametric approach works but the computational cost is still high.

2.6 Evaluation Metric

To the best of our knowledge, there is no unified evaluation metric to evaluate the disaggregated results of energy disaggregation. Suppose we know the true power consumption value of each device, then there are two metrics to evaluate the device disaggregation results - event based and

time series based. In the event based metric, we check whether the disaggregated turning on and off events are correctly classified for the target device. The time series metric gauges whether the disaggregated power values of each device is in the range of ground truth over a period of time.

For both event-based metric and time-series-based metric, the disaggregation results can be measured through the confusion matrix, F-measure, or simple error rate.

2.6.1 Evaluation Based on Events

Event-based evaluation metric, primarily on and off events, has been widely used in previous research work. They are identified by real power and reactive power as stated in [16] and [23], by real power and transient shapes [46], by real power, reactive power and voltage-current trajectory [134], by just transient shapes [24], by comparing the waveforms as shown in [124] and [17], or by analyzing the voltage noise [106].

Generally, the events classification rate is calculated as follows. For each device, suppose there are totally N on or off events $\{E_1, ..., E_i, ..., E_N\}$ during a period of time, the corresponding predicted events are $\{\hat{E_1}, ..., \hat{E_i}, ..., \hat{E_N}\}$, and the the coverage range of these on or off events is given in Equation (2.32) in [101].

$$E_{coverage} = \frac{\sum_{i=1}^{N}(E_i - \hat{E_i})}{\sum E_i} \tag{2.32}$$

Higher coverage for a device implies better prediction results.

Secondly, the disaggregation accuracy rate can be calculated by judging whether the disaggregated devices are classified as the right device or not. [48] evaluates the classification results based on this criteria as given in Equation (2.33).

$$purity_m(\Omega, C) = \frac{1}{1/N_m} \sum_k \max_m |\omega_k \subset c_m| \tag{2.33}$$

where Ω is the set of all ground truth device labels, C is the disaggregated device labels set. Suppose there are M number of clusters corresponding to M devices, N_m is the number of elements in cluster m. ω_k is the subset with the highest frequency in each $c_j = m$ cluster. Other than the aforementioned classification accuracy rate, F-measure is also employed in [136] to get the disaggregation result.

2.6.2 Evaluation Based on Time Series

The time series metric compares the disaggregated power values with the ground truth power values at each point over a period of time.

Using the time series, [71] compares the disaggregated time series with the ground truth of each device given in Equation2.34.

$$\sqrt{(\sum_{t,m}\|y_t^{(m)} - \hat{y}_t^{(m)}\|_2^2)/(\sum_{t,m}\|y_t^{(m)}\|_2^2)} \tag{2.34}$$

where y_t is the true real power value at time t, $\hat{y}$ is the disaggregated real power value. Note that, the disaggregate error rate can be calculated over a specific time range [71]. In addition, [105] uses the square root of error rate of all devices over a period of time to calculate the disaggregation accuracy as in Equation (2.35).

$$\sqrt{\frac{1}{T}\sum_t (y_t^{(m)} - \hat{y}_{\mu_t^m}^{(m)})^2} \tag{2.35}$$

where m is the number of devices, y_t^m is the power value of device m at time t, μ_t^m is the average true power value of device m at time t.

The third method to measure time series data is F-measure. [65] is evaluated based on F-measure of time series.

The fourth method is to evaluate by accumulating the total energy over a period of time as [114]. Once again, F-measure is used to evaluate the performance of the algorithm.

2.6.3 Evaluation Based on Combinational Metrics

Note that some papers propose several approaches to evaluate the experimental results. [87] proposes three evaluation metrics, namely detection accuracy, disaggregation accuracy, and overall accuracy. The first one is the events classification accuracy. The last two metrics is similar to the standard F-measure metric that is commonly used in machine learning algorithms.

Summary of Evaluation Metric

In conclusion, the evaluation in energy disaggregation is not standardized, which makes it harder to compare different works even with if the same data set is used. If the research community were to agree on one or two evaluation metrics, a fair comparison between several algorithms can be performed. Moreover, researchers can work on improving the accuracy of their algorithms for the standardized evaluation metrics.

2.6.4 Data Collection and Public Data Sets

The usefulness of energy disaggregation algorithms is a function of the aggregated datasets' availability. Therefore we need to collect this data by installing the corresponding meters and sensors and setting up the necessary experiments.

Meters

Generally speaking, there are two types of data that can be used to disaggregate devices: AC power and non-AC power information. To obtain AC power data, real power meter, reactive power meter, ammeter, voltage meter (usually consolidated into one meter) can be installed to record different power values, current or voltage values, and noises generated by power line. Sensors are installed to collect non-AC power data like electromagnetic fields (EMF) around devices [47], light, and sound [66].

Figure 2.24 illustrates how four types of meters/sensors are installed in a building. After 2-phase power is delivered into the home, three power meters which record real power, current, and voltage are installed on these three entry power lines separately. On each circuit, such as a refrigerator, a power meter is installed to monitor the true status of the devices (for validating results). On the outlet, a sensor is plugged in to monitor the voltage noise data. Besides these AC-power meters, an electromagnetic field sensor, a sound sensor , and a light sensor may also be installed around devices, such as refrigerator to capture its electrical magnetic field, sound-related, and light-related operations.

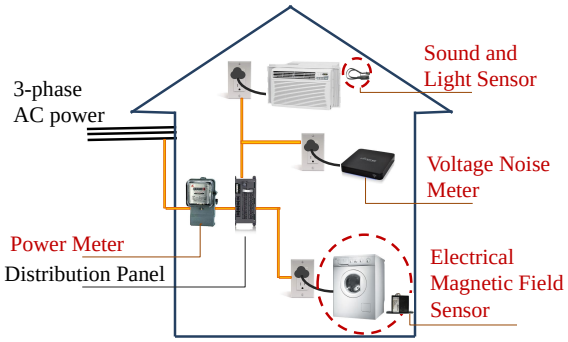


Figure 2.24: Four Types of Meters in a Building.

The meters or sensors that have been used in the experiments are listed in Table 2.3. Devices like ammeter to gauge the current value, voltage meter to record the voltage, wattmeter to log the real power, and reactive power meter to record the reactive values are easily available.

Low frequency data and high frequency data

When meters are installed to monitor the voltage and current, generally two kinds of data are collected: low frequency data and high frequency data.

In North America, the basic frequency of voltage or current is 60 Hz. If the interval between successive data points is larger than 1/60s, the data recorded by the meter is low frequency data, otherwise it is high frequency data. High frequency data can recover the waveform as illustrated

in Figure 2.4. In practice, only the apparent power or real power is measured for low frequency data. For high frequency data, to facilitate the capture of different device characteristics, normally current and voltage are monitored separately. In order to capture high order harmonics, the sampling frequency to record the data should be at least twice as much as the highest frequency. If targeting to capture harmonics with the highest order $N_{highest}$, the desired harmonics frequency is $N_{highest} \times 60Hz$, then the sampling frequency of recorded data f should meet the criteria that $f \geq N_{highest} \times 1/60Hz$.

There are many meters that can record the low frequency power. However, high frequency data must be monitored by special devices as TED [126]. Some examples of aggregated energy data collection are mentioned here. [17] install voltage/current meter in a residential home in Pittsburg, PA. This experiment chooses 17 on-off devices and installs plug-level meter for these 17 devices. It records the data at a high frequency (100 kHz). Then features such as real and reactive power, harmonics are extracted at the frequency of 20 Hz. Voltage noise data [106] is obtained in high frequency sampling rate by plugging meters into an outlet. [10] install an optical sensor to collect the real power. Then the on-off events are extracted from the real power. A detailed comparison of whole-house meter, circuit meters and plug-meters is given in [17].

Public datasets

Although there are many data sets especially in power industry for energy disaggregation, majority of them are not open to the public. So far there are a few public data sets REDD [72], BLUED [9], Smart [13], AMPds [92], CASAS [33], iAWE [14], and GREEND [99]. The first open data set REDD, when introduced, opened the doors for several researchers to attack the energy disaggregation problem.

So far, a majority of the data is stored as plain text. Some work proposes to store these datasets in database [74] or builds a metadata as a standard [64].

2.7 Ongoing Research

Current data mining research in energy disaggregation focusses on two areas: feature discovery and developing learning algorithms for disaggregation. Feature discovery is mainly explored by electrical engineering specialists who have a better understanding of device or electricity features. There is a lot of scope for data scientists to extend the research in developing learning techniques as this part of research is still in its nascent stages. For learning algorithms, unsupervised ones have a distinct advantage since labelled data is not required. The introduction of more new statistical models will improve the disaggregation accuracy.

As more electric companies join this field, software tools have been developed to analyze data on power consumption. Smart![13] provides an interface tool so that users can monitor their power

consumption. A database has been built for REDD [74]. These tools benefit both the developers and customers.

Non-intrusive load monitoring paves the way for many other research problems. One of them is occupancy research, which infers whether there are people living in the home [26]. The second one is demand response. Inferring activities of daily life [118], personal energy usage [82], efficient energy management [31] are all topics of research.

2.8 Conclusion

Significant increase in energy usage worldwide and the consequent impact on the environment has pushed energy disaggregation research to the forefront in recent years. While energy disaggregation primarily refers to electricity disaggregation, similar algorithms are being explored for natural gas and water disaggregation. We have surveyed features, algorithms, evaluation metrics, and instrumentation required for energy disaggregation from the perspective of data mining. Initially, disaggregation algorithms focused on features of real power and reactive power, which can be easily obtained from low frequency data. With decreasing cost of meters to record data, high frequency consumption data can be recorded these days. Therefore, rich features such as harmonics, transient shapes, noise data, and electromagnetic fields are available which increase accuracy. While supervised algorithms were first used in energy disaggregation, it is becoming more common to use unsupervised algorithms. Although there is no unified evaluation metric for energy disaggregation so far, there are two types: event-based and time-series based. An important need for the research community is to agree on a standardized evaluation metric. This will assist researchers in comparing and improving their algorithms' performance. In addition, we describe the setup of experiments on how to record data. In the near future, more data mining algorithms will be designed and invented in the energy disaggregation area, thus improving the disaggregation accuracy and scalability, and enabling its widespread use.

Table 2.2: Energy Disaggregation Algorithms Categories.

Category	Sub-category	Algorithm Name	Example	Features Adopted
supervised	Classification	Pair-wise match	[52]	real reactive power
		Neural Network	[111]	real power, reactive power
		SVM	[106]	startup of I_{AC} and voltage noise
		SVM, AdaBoost, RBF, NN	[103]	startup of I_{AC}
		SVM, KSC	[100]	startup of I_{AC}
		SVM, RBFN	[101]	real power, harmonics of I_{AC}
		Bayesian Classifier	[17]	real power
		Genetic algorithm	[12]	startup of I_{AC}, on duration
		Rule-based	[109]	real power
		Dynamic Bayesian network	[46]	real power
		Decision Tree	[16]	startup of I_{AC}
		AdaBoost	[16]	startup of I_{AC}
	Nearest Neighbor	KNN	[116]	startup of I_{AC}
		Duration PDF	[138]	real power, on duration
	Statistical Model	General likelihood ratio	[8]	real reactive power, duration
	Optimization	Dynamic Programming	[11]	I_{AC}
		Dynamic Model	[38]	real power
		Integer Programming	[124]	I_{AC}
		Sparse Coding	[70]	real power
		Nonnegative Tensor Factorization	[45]	real power
unsupervised	Clustering	Hierarchical Clustering	[48]	real power, reactive power
	HMM-based	FHMM	[65]	real power, time, duration
		AFAMAP	[71]	real power, startup of I_{AC}
		Difference FHMM	[105]	real power
	Temporal mining	Motif Mining	[115]	real power
semisupervised	Clustering	Hierarchical Clustering	[75]	startup of I_{AC}, I_{AC}, V_{AC}
	HMM	HDP-HSMM	[61]	real power
	Optimization	Contextually supervised	[133]	real power

Table 2.3: Meters Used in Experiments.

Meter Types	Meter Name	Meter Example	Recorded Features
AC power	ammeter	TED, LEM LA55-P [126]	AC waveform, harmonics
	voltage meter	Pico TA041 [125]	voltage waveform, voltage
	real power meter	National Instruments USB-9215A [57]	real power
	reactive power meter	TrendPoints EnerSure [127]	reactive power
	voltage noise meter	Build by author	voltage noise
Non-AC power	electromagnetic field meter	Trifield [5]	electromagnetic field
	sound sensor	mindstorms [41]	sound strongness
	light sensor	extech [56]	light strongness
	temperature meter	amprobe [7]	temperature

Chapter 3

Energy Disaggregation

3.1 Abstract

Non-intrusive appliance load monitoring has emerged as an attractive approach to study energy consumption patterns without instrumenting every device in a building. The ensuing computational problem is to disaggregate total energy usage into usage by specific devices, to gain insight into consumption patterns. We exploit the temporal ordering implicit in on/off events of devices to uncover motifs (episodes) corresponding to the operation of individual devices. Extracted motifs are then subjected to a sequence of constraint checks to ensure that the resulting episodes are interpretable. Our results reveal that motif mining is adept at distinguishing devices with multiple power levels and at disentangling the combinatorial operation of devices. With suitably configured processing steps, we demonstrate the applicability of our method to both residential and commercial buildings.

3.2 Introduction

As the saying goes, sustainability begins at home. Greater than ever before, there is now a significant interest in reducing household energy footprints by providing consumers with detailed feedback on their energy consumption patterns. By contrasting such 'drill-down' data with neighborhood profiles, consumers can make better informed decisions about how their daily activities impact the environment as well as their bottom line.

A key step in this endeavor is energy disaggregation. This is the task of, non-intrusively, monitoring aggregate energy usage (electricity, water) at a home/unit and separating it out into individual appliances, subunits, and other spatial dimensions automatically, using machine learning methods. A variety of methods have been proposed, e.g., factorial HMMs [65] and sparse coding [71] but the increasing diversity of appliances to be accommodated and the spatio-temporal coherence

properties that must be modeled provides continuing opportunities for algorithm innovation.

Here we propose a temporal motif mining approach (see [29, 135] for background) to energy disaggregation. We specifically focus on low-frequency measurements since those can be obtained from smart meters and aim to characterize stable power consumption events, in contrast to transients. The basic idea is to discover the minimal episode which corresponds to a complete state-change cycle by a device or part of a device. Unlike state-of-the-art probabilistic methods that posit detailed temporal relationships and involve complex inference steps, we argue that our method is lightweight and, at the same time, capable of accuracy levels better than or comparable to these more complex methods. Using this approach, we conduct a thorough experimental investigation of our method on a residential dataset (REDD [72] as well as a commercial dataset, demonstrating the ability of our approach to disaggregate different classes of electrical loads.

3.3 Background

Residential vs commercial buildings.

There are significant differences between residential and commercial disaggregation problems. First, the number of devices is one to two orders of magnitude larger in commercial buildings. Although disaggregation of *all* devices is not feasible in commercial buildings, we can disaggregate branches of the electrical infrastructure resulting in a drastic reduction in the number of meters required to monitor loads. The electrical infrastructure in residences and commercial buildings also differs. The former have low voltage levels (e.g., 110V or 220V) and two phase circuits while the latter have three-phase, high voltage lines coming from the utility which feed a hierarchical electrical infrastructure in the building. Heavy duty equipment such as chillers, blowers, pumps, elevators, etc., use three-phase power, which is then split into two phases and stepped down for lighting and plug loads. Residences typically receive two-phase power from the utility, as shown in Figure 3.1. Each phase connects to many circuits and in turn each circuit has one or more devices that draw power from it. Devices in residences usually consist of microwaves, refrigerators, ovens, lights, washers/dryers, and air conditioners. Some devices such as washers/dryers typically connect to both phases. Compared to residences, there is more automation in commercial buildings, e.g., blowers, pumps, lights and other devices are controlled by a building management system (BMS) and turn on/off at scheduled times. Most of the past research in disaggregation pertains to residential buildings.

High frequency vs low frequency sampling.

High frequency sampling, typically at the rate of hundreds to thousands of Hz, can reveal transients in the electrical signal which can then be used as features for disaggregation. However, customized HW usually needs to be installed to sample at such high rates. Low frequency sampling, typically

at rates of 1Hz or below, can be obtained from smart meters, which are being deployed in increasing numbers by utilities worldwide.

Multiple states and transients.

The device to power state mapping is not one-to-one. A given device might involve multiple power states as shown in Figure 3.2 (left). For instance, a washer/dryer might function at a fixed power level of 1700W but later change levels based on its workload. Further, as shown in Figure 3.2 (right), before the refrigerator reaches a stable state, a transient is observed and, after a period of time, the power consumption stabilizes to a certain level.

Energy disaggregation.

Energy disaggregation, initially proposed by [52], records only the power at the main entry or several points of a building, and aims to deduce the power consumption of devices in the building over a period of time through analysis of the aggregate. Figure 3.3 gives an example of energy disaggregation where a total power time series is disaggregated into fourteen devices over a period of time (here, 8am to 12 noon). For instance, note that it has been deduced that the refrigerator (in purple) is switched on for three periods of time, namely, 8:50am to 9:05am, 10:15am to 10:40am, and 11:50am to 12:05pm.

Challenges.

The field of disaggregation has over the last twenty years developed many practical solutions drawing primarily from the field of electrical engineering. However, many challenges remain, including lack of knowledge about the number of power levels of each device, uncertainty about the number of steady states for a given device (e.g., a microwave oven can operate in states of defrost, heat

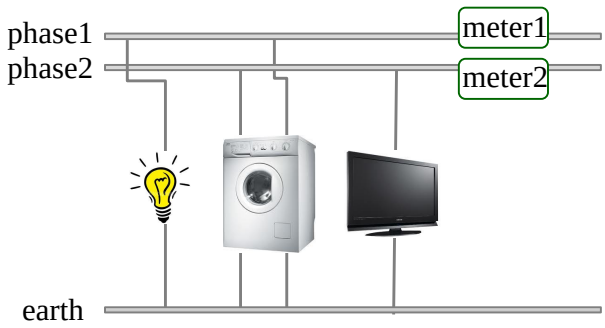


Figure 3.1: A residential setup for data collection.

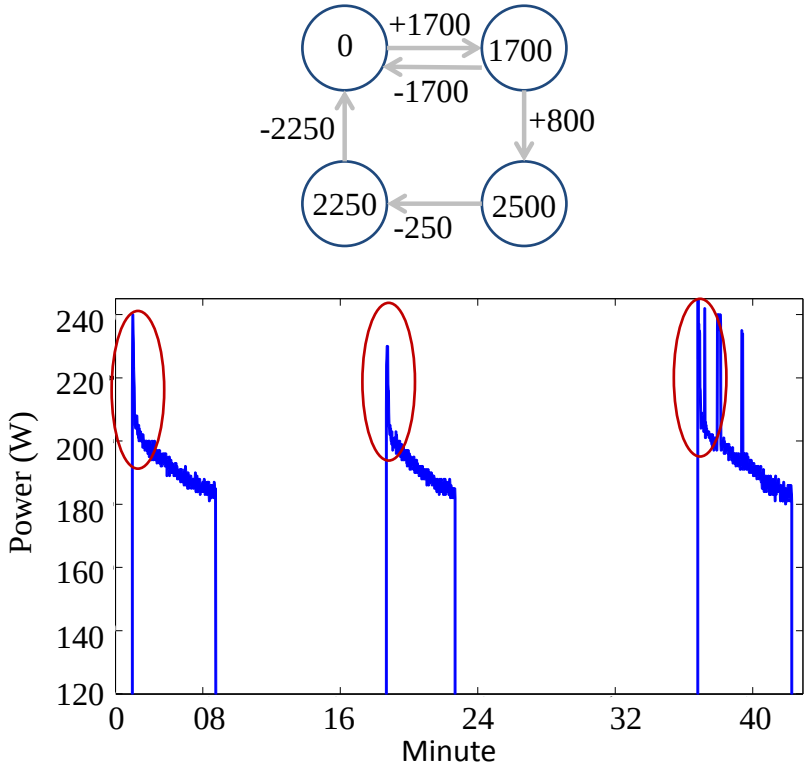


Figure 3.2: Steady state transitions and transient features at startup.

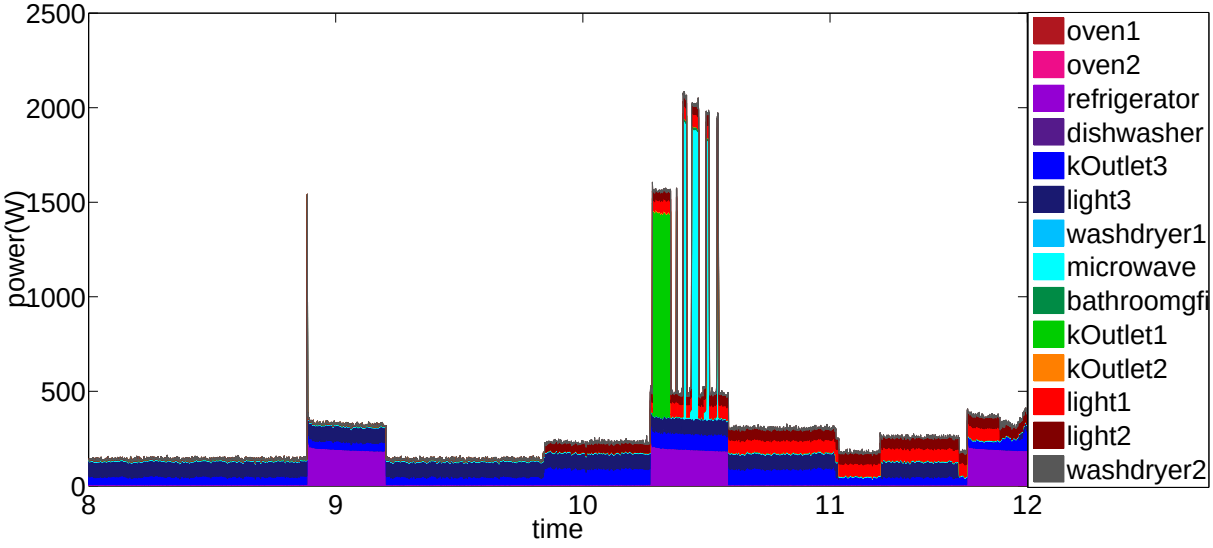


Figure 3.3: Example of energy disaggregation.

with low power, or with high power), multiple devices exhibiting the same power level (e.g., lights and monitors), concurrent switchings on/off of multiple devices (e.g., printers and PCs), distinguishing start up transients from steady state levels (the former could persist for significant periods in time in commercial buildings), variable speed devices that show continuous power levels, and rare operation of some devices (because they are seldom operated by humans). These challenges are aggravated in commercial buildings [102] compared to residential buildings.

Features from meters.

Let us first review the type of features discernible from metered usage data. From low frequency measurements, it is possible to infer features such as steady states, real power, reactive power, low-order harmonics, and the time of day. From high frequency measurements, in addition, we will be able to discern characteristics such as higher-order harmonics and the current or voltage waveform. In addition, from high frequency data, it is possible to discern transient states.

Prior approaches to disaggregation.

Initial research focused on using simple device features such as real power and reactive power [52]. With the development of automated meters, transient states generated when devices turn on have been employed to identify devices [116]. Raw current waveforms [119], and voltage waveforms [75], and transforms of the current waveform [22] have also been adopted as characteristics. In particular, harmonics of non-linear devices have been utilized in prior work [22]. Further, non-AC power features such as power line noises [106], time of day and device correlations [65], can be combined with AC power features to aid disaggregation. The underlying algorithms have been drawn from a variety of domains: supervised learning [101], data mining, optimization, and signal processing, e.g., kNN [116], SVM [106], sparse coding [70]. Recent research has placed a great emphasis on building in unsupervised learning features, including hierarchical clustering [75], semi-supervised approaches [105], factorial HMMs [65], and AFAMAP [71].

3.4 Temporal Motif Mining

Early approaches to disaggregation (e.g., Hart[52]) assume that only the aggregated current and voltage information is known whereas later work assumes that the number of devices, possible steady states of devices are also known, so that the problem reduces to minimizing the error between the combination of disaggregated devices and the ground truth devices. Here, we assume that the number of devices/number of circuits is known, a reasonable assumption since such information is obtainable from a top-level circuit map of the building.

Our framework (see Figure 3.4) unifies clustering and temporal data mining to discover power levels, forms episodes from power levels corresponding to devices, and models the underlying time

series as a mixture model whose components correspond to the device episodes. The framework has six key stages, viz. baseline removal, steady states extraction, episode mining and selection, probabilistic sequential mining, motif mining or time-based motif mining, and device recovery. Gray box in Figure 3.4 denotes that the step can be neglected (and are typically used when disaggregating for commercial buildings).

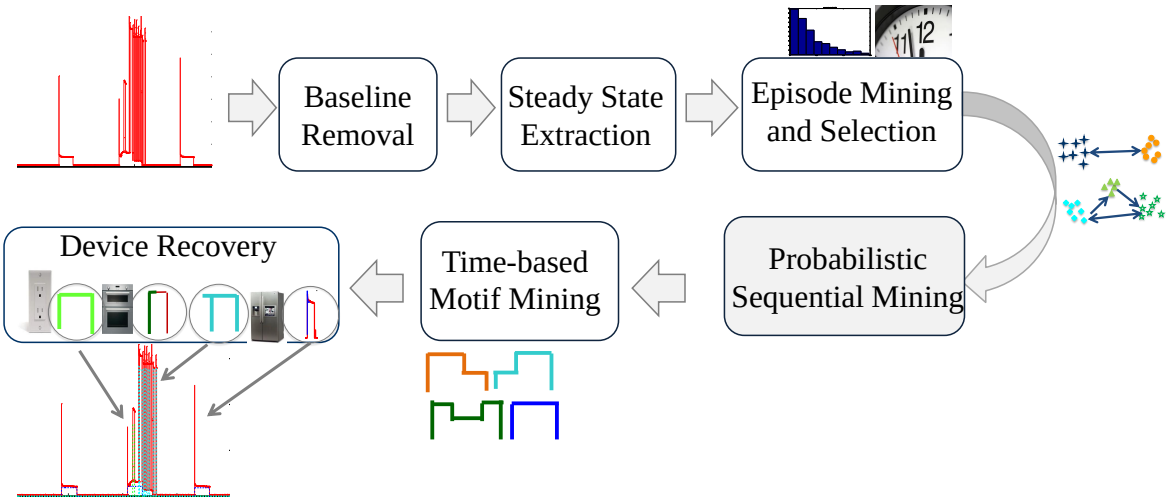


Figure 3.4: Temporal motif mining framework for disaggregation.

Baseline extraction.

Baseline removal aims to separate devices that are always on. Given the aggregated (input) power series $P(t)$ over time period T, the baseline power P_{base} is defined such that $P_{\text{base}} \geq \min_t P(t)$ and where $f(P_{\text{base}}) \geq \alpha T$ (a minimum support threshold).

Steady state extraction.

Two basic approaches here involve a heuristic method (window-sized filtering) and the more systematic Dirichlet process Gaussian mixture models (DPGMMs) [49]. In the former, a mean filter smoothing is typically applied whose window size is adjusted to correspond to the mean or maximal start time duration in the given collection of devices (e.g., this could be just a second in the case of lighting, but higher for say a refrigerator). A DPGMM can be viewed as an infinite-mixture extension of a traditional Gaussian mixture model (GMM). Recall that in a traditional GMM, $\mathbf{y} = \Sigma_{i=1}^{k} \alpha_i N(\mu_i, \Sigma_i)$ where $\Sigma_i \alpha_i = 1$, and each component has a mean μ_i and covariance matrix Σ_i. A DPGMM defines Gaussian priors for all the component means μ_j:

$$p(\mu_j|\lambda, r) \sim N(\lambda, r^{-1})$$

The distribution of λ is set to be a Gaussian prior and the distribution of r is set to have a Gamma prior, so that the number of points in each component i conforms to a multinomial distribution with an unknown number of components. After modeling all the power levels in this manner, we replace all values with their representative (nearest centroid) power levels, record only the differences in successive power levels, and use this 'diffs' time series for further modeling.

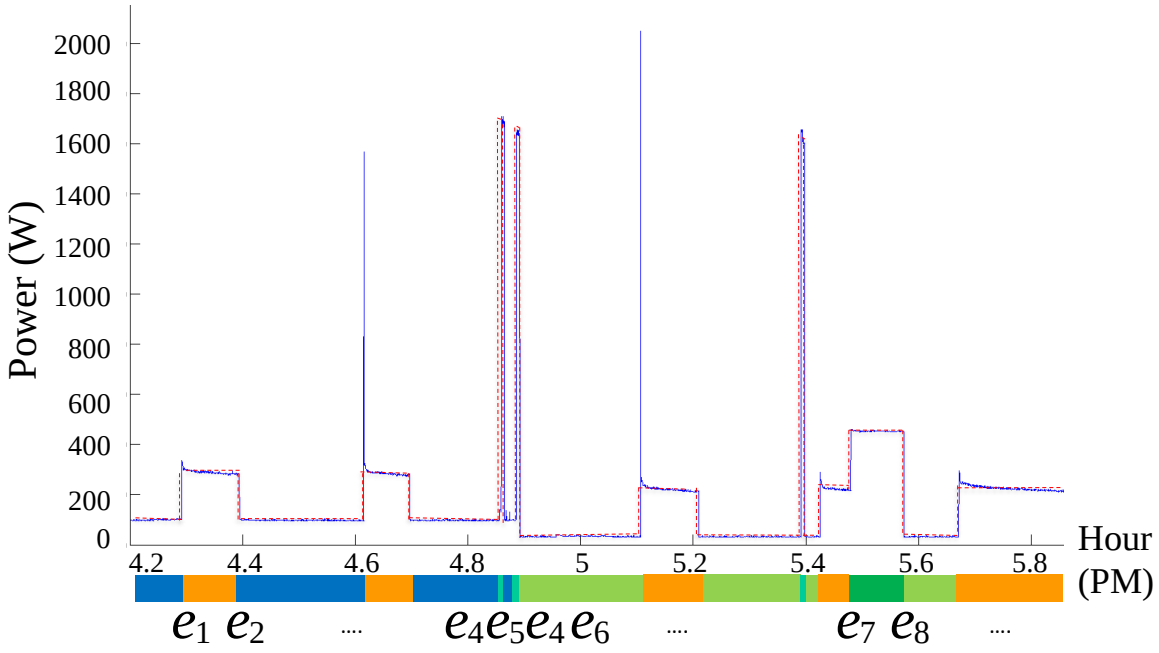


Figure 3.5: Mining episodes from a symbolic time series.

Episode mining and selection.

The goal of episode mining [107] is to identify repetitive sequences of power level changes and, further, to isolate (select) those episodes that potentially correspond to the operation of a single device. Recall that at this point, we have generated a symbolized time series from the 'diffs' data. Let the set of symbols be S. From the diffs sequence, the transitions between symbols are recorded to help constitute episodes. We set the max episode length to be N, corresponding to the N-1 states of a device. Then all the symbols in the symbol set are permuted with length from 2 to N. As a result, all possible episodes with length from 2 to N are generated. To select valid episodes, some constraints checks are performed.

First, steady state values extracted from the previous step are clustered into a discrete symbol time series and transitions between symbols are recorded to identify episodes. Figure 3.5 describes how transition events are generated, resulting in the event series: $(e_1, e_2, e_1, e_2, e_4, e_5, e_4, e_6, e_1, e_2, e_4, e_5, e_1, e_7, e_8, e_1)$. An episode of length N, $E = (e_1 \rightarrow e_2 \rightarrow ...e_N)$, denotes an ordered sequence of (not necessarily

consecutive) symbols. To select those episodes that correspond to characteristics of an electrical device, several constraints are introduced:

1. *The sum of the power level changes corresponding to the events of a episode is nearly zero.* Figure 3.2 (left) shows an example, where there are two complete episodes for a washer-dryer: (+1700, -1700) and (+1700, +800, -2250, -250).

2. *The sum of the power level changes corresponding to any prefix of a episode is positive.* This constraint is particularly geared toward multiple state devices. Fig 3.6 shows two examples of episode selection based on this constraint The episode (+100,-100) is retained but the episode (-100,+100) is discarded. As another example, episode (+600, -400, +1000) is chosen and episode (+600, -1000, +400) is discarded. Note that this assumes there are no always on devices.

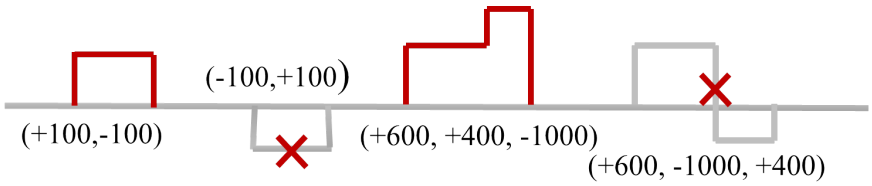


Figure 3.6: Episode constraints.

3. *The absolute value of the power level of any event in an episode related to a device must be higher than a support threshold over the maximum level in the episode.* In other words, power state changes in a device are assumed to be greater than a support threshold. This condition is intended to exclude cases where the low power consumption of one device inadvertently forms part of the episode of a high power consumption device. For instance, using a support threshold of 0.1, the episode $(1000, -850, -90)$ will get disqualified (because $90 < 100$) since this episode is likely generated by more than one device, rather than a single device.

Probabilistic sequential mining.

This step aims to discover devices that exhibit several power levels sequentially and which operate frequently within a very short period of time. We use sequential mining [3], a levelwise framework, with duration constraints to discover such devices. We begin by seeking episodes that satisfy the above three checks and which can be systematically grown into longer chains of power level changes within a user-specified window.

Devices in commercial buildings are often scheduled to turn on/off at fixed time. Therefore, we cluster power levels according to time of day and day of week. We apply hierarchical clustering with Ward Euclidean distance to diffs of power levels. As a result, each set of power level diffs that

qualifies the three constraints are chosen. For example, a cluster can identify a power level diff set $S = \{e_1, e_2, ...e_n\}$ belonging to a single device.

Regarding probabilistic sequential mining, a coverage probability θ, say 0.9, is introduced to determine what percent of power levels should be covered for each device. Probabilistic sequential mining only considers the coverage of power levels rather than the sequence of power levels as motif mining.

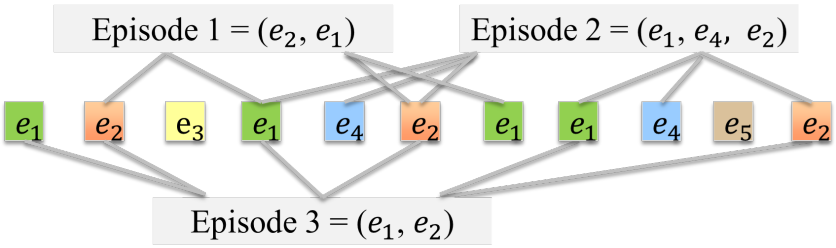


Figure 3.7: Illustration of motif mining. Note that there are 3 non-overlapped occurrences of Episode 3.

Motif mining.

Motif mining aims to find repetitive episodes in a time series using the technique of non-overlapped occurrences [78]. Assume there are five power event change symbols $\{e_1, ..., e_5\}$ and a time series $(e_1, e_2, e_3, e_1, e_4, e_2, e_1, e_1, e_4, e_5, e_2)$ which is produced by these five symbols as shown in Figure 3.7. Consider the episode Episode 3, composed of two ordered events (e_1, e_2). In this time series, there are four e_1 and three e_2 occurrences, and three instance of Episode 3. The first e_1 and first e_2 comprise the first instance of Episode 3. The second e_1 and second e_2 make up of the second instance of Episode 3. The third instance is composed of the fourth e_1 and the third e_2. Other possible instances of Episode 3 which may cause overlaps with the above instances are not considered in the non-overlapped count measure, such as (e_1, e_2) which consists of the first e_1 and the second e_2. With this count measure, all episodes that have support greater than the specified threshold are discovered by motif mining. For commercial buildings that have scheduled on/off devices, we adopt a time-constrained version of non-overlapped count, where the episode growth is restricted to events that fall within a specified time window.

Computational complexity

Assume m is the number of power levels in the 'diffs' data. Then the computational complexity of DPGMM is $O(mnd^2 + md^3)$, where n is the number of points in diffs data, and d is the number of feature dimensions (e.g., time, date). The computational complexity for the episode generation step is $(p-1)O(m^2)$, where p is the maximal episodes length. Since p, which is 3, and m, which

is 14 or 27, are small, we apply a brute force approach. The worst-case time complexity of the motif mining algorithm is $O(msq)$, where q is number of candidate episodes, and s is the size of the episode.

Parameters

There are three kind of parameters used: (1) those pertaining to power level generation, (2) threshold for moitif mining, and (3) window size for median filtering. For each of these, a range of values were tried and their values were set based on performance on a test set.

3.5 Evaluation

We use precision, recall and F-measures in our evaluation. The standard definition of these metrics are: precision $= \frac{TP}{TP+FP}$, recall $= \frac{TP}{TP+FN}$, F-measure $= \frac{1}{\frac{1}{\text{precision}}+\frac{1}{\text{recall}}}$

We need to define the notions of true/false positives and negatives in the context of disaggregation.

Now suppose there is a ground truth time series X with length T; denote the corresponding disaggregated time series by X^*. For any time $t \in (0, T)$, there are two values: the ground truth value $X_i(t)$ and the disaggregated value $X_i^*(t)$. We define a parameter ρ for the range of true values $X_i(t)$ and another parameter θ as the noise. For any given measurement, there are four total power values at each point: true positive Ψ_{TPi}, false negative Ψ_{FNi}, true negative Ψ_{TNi}, and false positive Ψ_{FPi}.

1. When $X_i(t) > \theta$ and $X_i^*(t) > \theta$, at this point the disaggregation is a true positive. There are three situations in turn:

1.1. When $X_i(t) \times (1-\rho) < X_i^*(t) < X_i(t) \times (1+\rho)$, then

$$\begin{aligned} \Psi_{TPi} &= X_i^*(t) \\ \Psi_{FNi} &= \Psi_{FPi} = \Psi_{TNi} = 0 \end{aligned}$$

1.2. When $X_i^*(t) < X_i(t) \times (1-\rho)$, then only the disaggregated power is considered as true positive and the power that is not disaggregated is regarded as a false negative:

$$\begin{aligned} \Psi_{TPi} &= X_i^*(t) \\ \Psi_{FNi} &= X_i(t) - X_i^*(t) \\ \Psi_{FPi} &= \Psi_{TNi} = 0 \end{aligned}$$

1.3 When $X_i^*(t) > X_i(t) \times (1+\rho)$, then the disaggregated power is a true positive, and those

values which are greater than the truth values are treated as false positive.

$$\begin{aligned}\Psi_{TPi} &= X_i^*(t)\\ \Psi_{FPi} &= X_i^*(t) - X_i(t)\\ \Psi_{FNi} &= \Psi_{TNi} = 0\end{aligned}$$

2. When $X_i(t) > \theta$ and $X_i^*(t) < \theta$, at this point the disaggregation is a false positive. Then,

$$\begin{aligned}\Psi_{FPi} &= X_i(t)\\ \Psi_{TPi} &= \Psi_{FNi} = \Psi_{TNi} = 0\end{aligned}$$

3. When $X_i(t) < \theta$ and $X_i^*(t) > \theta$, at this point the disaggregation is a false negative. Then,

$$\begin{aligned}\Psi_{FNi} &= X_i(t)\\ \Psi_{TPi} &= \Psi_{FPi} = \Psi_{TNi} = 0\end{aligned}$$

4. When $X_i(t) < \theta$ and $X_i^*(t) < \theta$, at this point the disaggregation is a true negative. Then,

$$\Psi_{TPi} = \Psi_{FNi} = \Psi_{FPi} = \Psi_{TNi} = 0$$

For the REDD dataset which features a maximal power level of 4000W, we use $\theta = 30$ and $\rho = 0.2$.

3.6 Experiments on REDD dataset

We conduct experiments on the low frequency data from the REDD [72] dataset. We focus on 'House 1' since it has the most complete information (for validation purposes) and because it features 18 devices, providing a good test for our algorithm. The sampling frequency of both the mains is 1s and that of each circuit is 3s. The power consumption for devices in this dataset ranges from 50W to 4000W.

3.6.1 Disaggregation experiments

Knowing the ground truth, we synthesize aggregate data with different combinations of devices/circuits and evaluate our algorithm by disaggregating the combined data into the constituent devices. Figure Figure3.8 (a),(b),(c) show the plots of precision, recall, and F-measure values for 14 devices. For each device the number of aggregate devices was increased from 2 to 11. Since for k devices, there are ${}^{14}C_{k-1}$ possible combinations for each device, the results show the average over all the combinations. In cases where number of such combinations exceeded 100, 100 combinations were randomly sampled and averaged. Figure 3.8(d) plots the power-weighted precision, recall and F-measure for these cases.

From Figure 3.8(a), we can see that devices that are used frequently (both consuming low and high power), such as oven2 (4000W), microwave (1527W), kitchen outlet1 (1076W) (kOutlet1), washdryer2 (2712W), refrigerator(193W) and light1(64W) exhibit a stable precision level (above 0.7) even with increase in number of devices.

In contrast, devices such as kOutlet2 (1535W) (kitchen-outlet2), that share similar power levels with microwave (1527W) and bathroomgfi (1605W) show greater precision drops with increase in number of synthesized devices. However, the more frequently such devices are used, the greater the precision level.

As Figure 3.8 (c) shows, devices with higher power or frequent use can be disaggregated well by motif mining. If a low power consumption device is prone to be influenced by high power devices, identification depends on the devices masking it; ultimately frequency of use helps disambiguate such situations. Finally, as Figure 3.8 (d) shows, precision, recall and F-measure decrease only slightly with increase in the synthesized number of devices. This shows that power levels of devices play a key role in determining accurate disaggregation. When true power levels are supplied, the average precision, recall and F-measure of motif mining fare slightly better than AFAMAP.

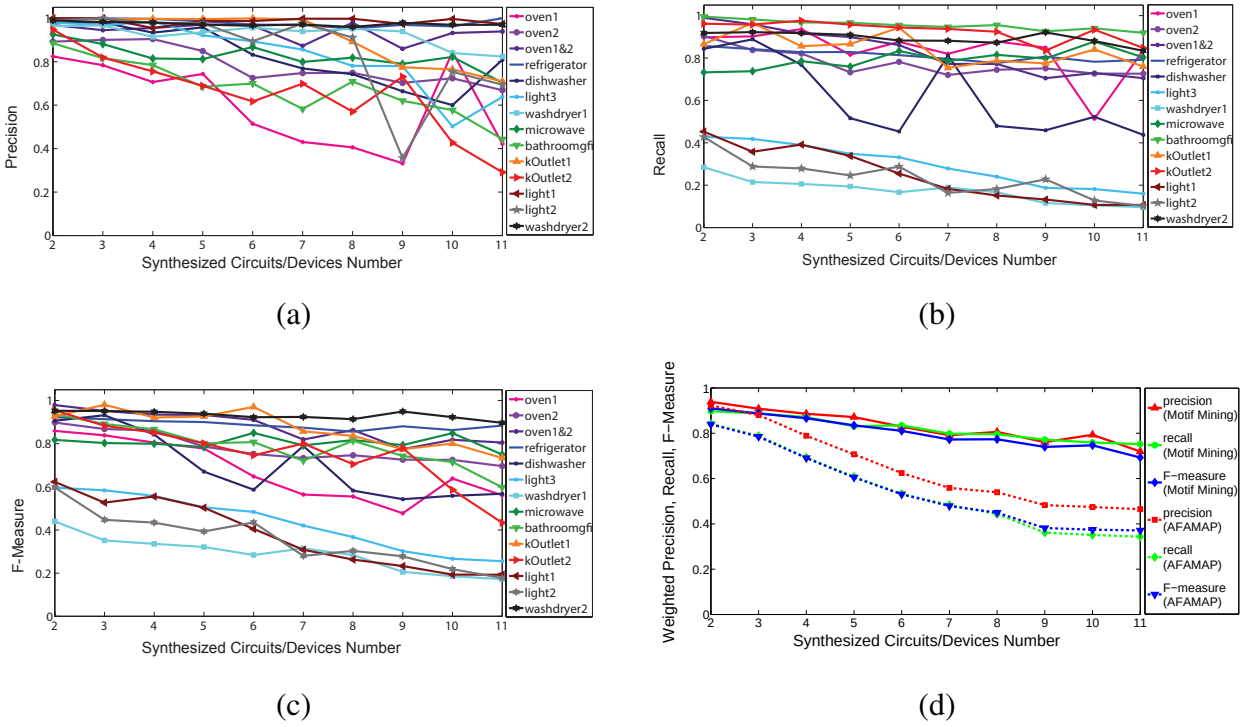


Figure 3.8: We increase the number of synthesized circuits from 2 to 11 and calculate performance measures for disaggregation of each device. (a) Precision (b) Recall (c) F-measure (d) The precision, recall and F-measure of all the devices are combined weighed by their average power levels.

Table 3.1: Comparing Motif mining against AFAMAP on the REDD dataset.

device	True Power (W)	Motif mining			AFAMAP (true power levels supplied)			Motif mining & AFAMAP		
		Precision	Recall	F-Measure	Precision	Recalll	F-Measure	Precision	Recall	FMeasure
oven1&2	4000	0.9297	0.5209	0.6677	0.4902	0.6750	0.5680	0.4008	0.6708	0.5018
refrigerator	193	0.9759	0.7368	0.8396	0.8825	0.3329	0.4834	0.7791	0.5433	0.6402
dishwasher	1113; 900; 400; 200	0.9786	0.2858	0.4423	0.062	0.4104	0.1077	0.5337	0.7431	0.6213
kOutlets3	100; 60	0.1487	0.0318	0.0524	0.6928	0.0439	0.0825	0.467	0.2892	0.3572
light3	282; 90	0.5768	0.1349	0.2187	0.4396	0.023	0.043	0.5519	0.1973	0.2907
washdryer1	466; 50	0.1789	0.1236	0.1462	0.3621	0.5401	0.4336	0.1703	0.6349	0.2686
microwave	1527	0.8035	0.3799	0.5158	0.5909	0.2907	0.3897	0.4512	0.3741	0.4090
bathroomgfi	1605	0.5199	0.6815	0.5898	0.2642	0.7551	0.3915	0.1075	0.406	0.1700
kOutlet1	1076	0.9320	0.6997	0.7993	0.21	0.7313	0.3264	0.2636	0.6394	0.3733
kOutlet 2	1535	0.2233	0.6261	0.3292	0.1153	0.2821	0.1637	0.0234	0.0826	0.0365
light1	64	0.6199	0.1963	0.2981	0.7972	0.0796	0.1447	0.667	0.1759	0.2784
light2	53	0.2603	0.1404	0.1824	0.6658	0.0817	0.1455	0.446	0.2776	0.3422
washdryer2	2711	0.9563	0.8305	0.889	0.7516	0.4237	0.5419	0.6427	0.3301	0.4361

3.6.2 Comparison of Motif Mining and AFAMAP

Next, we conduct experiments comparing our approach with the AFAMAP algorithm [71], and also develop a method that combines motif mining and AFAMAP. Unlike motif mining, AFAMAP requires the power levels of each device; when running AFAMAP separately, we use the ground truth power levels for each device. When using AFAMAP in conjunction with motif mining, we use the power levels from generated episodes as an input to AFAMAP. Table 3.1 lists the results of the comparison.

In all, there are 18 devices but 4 of them are seldom used; and, thus the remaining 14 devices can be disaggregated by these three methods. For high power consumption devices, such as oven1&2, bathroom_gfi, kitchen_outlet1, kitchen_outlet2 and washdryer2, motif mining performs much better than AFAMAP even when AFAMAP is supplied with the ground truth power levels. For some of the low power consumption devices (such as light1), AFAMAP performs better. For high frequency devices, such as the refrigerator, motif mining performs much better.

Furthermore, by integrating motif mining and AFAMAP, we see the performance is much better than the individual algorithms on multiple state devices such as dishwasher and light3. Since the

Table 3.2: Evaluation measures for commercial building disaggregation.

Device	Precision	Recall	F–measure
Pump and blower	0.99	0.99	0.99
Fan	0.99	0.99	0.99
Elevator	0.75	0.52	0.61

power level of light3 is low, the performance of the integrated method is better than using only motif mining.

3.7 Commercial Building Dataset

We applied our framework to a dataset from a commercial building (from HP Labs' campus in Palo Alto, CA). Data was collected from a branch in the electrical infrastructure of a large building and is composed of a root (aggregate) node and seven child nodes. Although all the nodes are instrumented with meters, we assume only the root and two of the child nodes, a transformer and a sub-panel, are available. The remaining five child nodes are devices that need to be disaggregated. These are: a pump, a fan, an exhaust fan, a blower, and an elevator. The real power of all nodes are logged at intervals of 10 seconds. Using ground truth data, we combine all five to synthesize the aggregated data.

After the processing steps as described in our framework, we find five power levels that often occur in a range of just around 1 minute. Therefore we set the window size to 60 seconds and apply probabilistic sequential mining using a probability of 0.8 (as described earlier). The precision and recall for extracting individual devices is shown in Table 3.2.

In analyzing these results, we discover that the baseline power is constituted of two devices, namely, the pump and the blower. The elevator shows a sequential episode involving six power levels. The scheduled device is a fan. The only un-disaggregated device in our experiments is the exhaust fan which has very low power consumption compared to others and thus can be disregarded.

3.8 Discussion

We have described an intuitive motif-based approach to disaggregation that performs well relative to more complex algorithms that perform detailed modeling of temporal profiles. More importantly, we have demonstrated how our approach is not just an aid to disaggregation but, as a byproduct, also extracts temporal episodic relationships that shed insight into consumption patterns. In this sense, our work goes further than past work into addressing the real goal of disaggregation research, namely, to understand systematic trends in consumption patterns with a view

toward identifying opportunities for savings.

Chapter 4

Proposal: Occupancy Prediction

4.1 Abstract

Conserving energy and optimizing its use has been a long standing challenge. Apart from the monetary benefits associated with tackling these problems, saving energy has significant positive environmental impact. For instance, can the HVAC of residential buildings be adjusted automatically based on occupancy? In our work, we mine the people's energy activity profile to predict the occupancy of residential buildings. We show that our method, which uses episode mining for target event detection and a mixture of episode generating HMM (EGH), generally performs better than the standard kNN approaches used for this problem.

4.2 Introduction

Modeling activity of daily life (ADL) has become a burgeoning research topic since people demand a comfortable life lifestyle at home at a lower cost. Since HVAC consumes $\sim$53% of the total electrical usage by heating and cooling spacesof an average household, automating the operation HVAC devices to save energy is important. One of the crucial components required to achieve this goal is to model and predict the occupancy of a home. Supervised learning approaches on the analysis of indoor temperature[67], smart phone's GPS data[68], electricity consumption[43] and sensor data by tracking the indoor activities[6, 113] are effective ways to approach this prediction problem. Prediction with sensor data is broadly researched. By capturing daily activities like room occupancy of the house, usage of electrical devices, usage of water system, etc. using sensors, researchers have modeled occupancy [15, 42, 90, 91] and used these results to automate the control of HVAC system.

Although the supervised learning kNN[113], neural network[90] and Markov model[42] are effective, the detailed household activities represented as a time series is not fully utilized. Daily

activities such as waking up, cooking, washing, commuting to work/school and back, etc. have different patterns based on the day. For instance, the schedule on a working day is significantly different from that of a weekend or a holiday. Thus, this scenario leads itself to an episode mining analysis, which can be used to predict household occupancy. Following this strategy of episode mining for occupancy prediction has three advantages. First, episode mining, a temporal mining approach, mines according to the time distribution for each type of activity. Second, it builds the activity scenario and connects the episode with a probabilistic hidden Markov model (HMM). As opposed to previous models, the time and order of each kind of activity are fully utilized. Third, the algorithm predicts according to the scenarios-based probabilistic model episode generative HMM (EGH). The prediction accuracy improves compares to the existing models.

Our contributions can be highlighted in the form of the three questions below.

1. How can we mine for meaningful scenarios? Episode mining can mine many frequent episodes, but not all the episodes are useful for occupancy prediction. By narrowing the episodes according to the start state, end time, event dwelling time and gap between two activities, we can interpret these episodes and provide insight as to which episodes are informative.

2. How can we predict the occupancy more accurately? Our dataset comprises of detailed information of the various activities of a household tracked as a time series on a daily basis. Thus our episodes have rich detailed information based on occupancy and un-occupancy of the household. Since we are mining episodes from this data, the accuracy of occupancy prediction improves significantly.

3. Can it help save electric usage at home? The prediction occurs at least 15-minutes ahead of a person leaving or coming back. By connecting this prediction result to an automatic controlling system over HVAC, the HVAC can be operated ahead. Since the HVAC doesn't work during occupancy, it saves electric usage.

4.3 Related Work

Most of the approaches that model and predict occupancy primarily use sensor data such as room occupancy, use of electrical appliances, water usage, etc. Several supervised learning approaches like kNN, neural networks, rule-based, and Markov chain models have been used to model and predict building occupancy [6, 15, 42, 90, 91, 113]. Using the kNN supervised learning algorithm and monitoring sensor data for portion of the day, [113] predicts the entire day's occupancy. A neural network approach using binary time series based on occupancy/unoccupancy along with exogenous input network (NARX) was proposed in [90, 91]. Mahmoud et al. tackle the problem by presenting a non-linear autoregressive with exogenous input (NARX) network. Several Markov chain models like blended Markov chain, closest distance Markov chain, and moving-window Markov chains were presented in [42]. A mixture of multi-lag Markov chains was used to predict

occupancy of single person offices [94]. In that work, the authors also compare their model with Input Output Hidden Markov Model, First Order Markov Chain and the NARX neural network.

Our work differs from previous research based on the main contributions listed below:

1. We formulate the problem as one of temporal mining: the activities inside the building are abstracted as episodes, and each episode is connected with an episode generative HMM model.
2. We mine the activity patterns according to the time and gap: both the duration of each type of activity, and the gap between two consecutive events are limited in a proper range. This range is extracted from the historical data according to the weekday and holidays.
3. Our prediction solution performs better than the kNN approach mentioned before, which is generally considered a benchmark in occupancy prediction problem.

4.4 Problem Formulation

Before formalizing the problem statement, we introduce several concepts and notations first.

Episode An episode is a pattern which can be explained meaningfully. For instance, we represent 'S' as sleep, 'K' as kitchen, and 'Z' as going out. If an episode $S \rightarrow K \rightarrow Z$ is found, the story is described as a person getting up, going to the kitchen for breakfast, and then leaving the house. An episode α is composed of a series of ordered events $\alpha = \langle E_1, .., E_t, ...E_n \rangle$, where E_t denotes that E occurs at time t. These events may be the point event or dwelling event. The dwelling event has a start time $E.start$ and end time $E.end$. In this paper E denotes a dwelling event and represents which room a person stays inside the building i.e is *occupied*. For example we if Z is used to show a room to be unoccupied. $Z.start$ means when the person goes out of home and $Z.end$ means when the person comes back.

EGH Episode generative HMM model is a type of HMM model which connects each episode α with a special HMM model Λ_α. The uniqueness of the EGH is that the transition matrix and emission matrix is only decided by a noise parameter η. The value of η is computed based on the frequency of the corresponding episode α.

We formulate the occupancy prediction problem as one of episode mining and event type time prediction problem.

Problem Statement Given a sequence of the room occupation events stream *s* with finite events symbols ε, and the target un-occupancy event *Z*, can we find the frequent patterns $\{\alpha_1, ...\alpha_n\}$ and can the corresponding episode generative HMM $\{\Lambda_{\alpha_1}, ..., \Lambda_{\alpha_n}\}$, predict when the person leaves $Z.start$, and when the person comes back $Z.end$.?

Next, we first discuss the time-gap constraint episode mining model and the mixture model, and

how to predict the target event in section 4. Then, in section 5 we will show the experimental results.

4.5 Constraint Episode Mining and Mixture EGH

We use a two-pronged approach to tackle the problem of mining and predicting unoccupancy. First, we use an episode mining algorithm to discover frequent events before a person leaves a room. Then we use a mixture HMM model, EGH, to predict whether the room is unoccupied and when the person leaves and comes back.

4.5.1 Time-gap Constraint Episode Mining

Episode mining has been studied in previous research [95] and [77]. Assume there is an event stream $s = ACBDEDEAABBA$, and the target episode is AB. To mine for AB, we can can use non-overlap mining approach to find the target episode. Non-overlap mining can be used where any two instances of the target episode has no intersection. It is to be noted that, non-overlap episode mining may result in different instances. For instance in the above example, if we align it to the left, the episode mining results are $\langle A_1, B_3\rangle, \langle A_8, B_{10}\rangle$. If aligned to the right, the results become as $\langle A_1, B_3\rangle, \langle A_9, B_{11}\rangle$. A variant of episode mining is event gap constraint episode mining as proposed in [108].

In this application, the events dwell at an event for a period of time. Therefore, we combine the above two episode mining algorithms and enforce more constraints. The first change is to use the right alignment for the first element in the episode. In the example of AB, the mined second instances is $\langle A_9, B_{11}\rangle$. The second modification is to check the time constraints and apply gap duration constraints between two consecutive events inside an episode. Figure 4.1 shows an example of time-gap constraint episode. Assume we have the frequent episode $S \rightarrow B \rightarrow K \rightarrow$

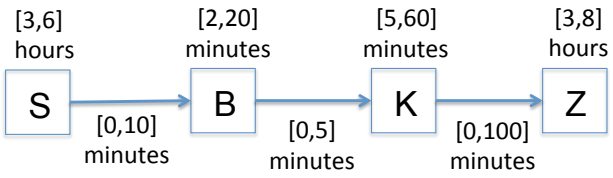


Figure 4.1: Example of Duration-gap Constraint Episode.

Z, we add the time constraints to each event $\{S, B, K, Z\}$. The dwelling duration of S is 3 to 6 hours, of B is 2 to 20 minutes, of K is 5 to 60 minutes, and of Z is 3-9 hours. Also, we set gap duration between any two consecutive events. The gap duration of SB is calculated as

$\Delta SB = B.start - A.end$. We set the maxim gap time between SB, BK, and KZ as 10 minutes, 5 minutes and 100 minutes; the minimal gap time is 0. Figure 4.2 is a time-gap constraint episode mining example. We have a stream composed of a sequence of dwelling events and the target episode is the same as in Figure 4.1. The unit of the figures is in minutes. Initially, a $waits$

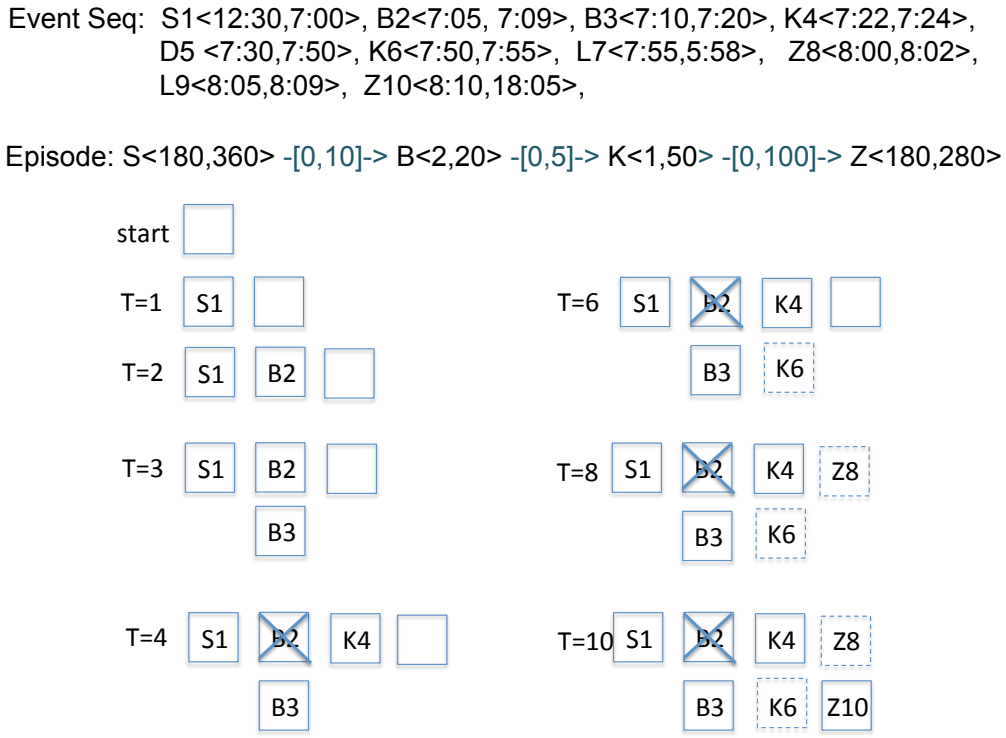


Figure 4.2: Time-gap constraint episode mining example.

structure related to this episode is created. Each $waits$ structure has the same length of episode structure $node$. A $node$ structure related to S is created and it waits for the first element of the episode $S\langle 180, 360\rangle$. When $T = 1$, the duration of $S1$ is checked. Since it's in the range of $3 - 6$ hours, $S1$ passes and is put into the node structure $node$ related to S. Then a new $node$ structure is created to wait for $B\langle 2, 20\rangle$. When $T = 2$ and $T = 3$, both of the $B2$ and $B3$ are qualified in terms of the time constraints and the gap constraints, e.g. the gap between S and B ΔSB should be between 0 to 10 minutes. Then $B2$ and $B3$ are input into the $node$ B structure in the $waits$ structure. At the same time, a new $node$ structure is created for $K\langle 5, 60\rangle$. When $T = 4$, the gap between $\langle B3, K4\rangle$ is satisfied with the distance condition between B and K 0-5 minutes. But the gap between $\langle B2, K4\rangle$ is longer than the constraint gap. Therefore, $B2$ is cancelled off. Now a new $node$ waits for the symbol $Z\langle 180, 540\rangle$. When $T = 6$, the gap from $B2$ and $K6$ is too far. Therefore, $K6$ is not added into the $node$ K structure in $waits$. When $T = 8$, the time duration of $Z8$ is not qualified the condition between 3-9 hours. $Z8$ is not added. When $T = 10$, the

duration of $Z10$ meets the requirement 3-9 hours and its distance to $K4$ meet the requirement of $\Delta KZ \in [0, 100]$ minutes. Thus $Z10$ is added into the $node\ Z$ structure in $waits$. Therefore, a complete episode mining is done.

This complete gap-constraint episode mining on dwelling events is described in detail in Algorithm 2 in the appendix section.

4.5.2 Episode Generating HMM

Each frequent episode is connected with an HMM. According to EGH [77], each episode generated HMM only has a noise parameter η. The noise parameter η of frequent episode α is calculated as $\eta = \frac{T - N f_\alpha}{T}$ [77], where T is the training data stream length, α is the frequent episode, N is the length of frequent episode α, f_α is the frequency over the time T.

Figure 4.3 gives an example of the transition matrix in EGH. Assume we have a N-node frequent episode $S \rightarrow B \rightarrow Z$ and $N = 3$ here. We define 2N number of hidden states, N for episode states, and N for noise states. The noise states are $W \rightarrow X \rightarrow Y$. An episode state transfers to another episode state at the probability of $1 - \eta$. An episode state transfers to a noise state at a probability of η. A noise state transfers to another noise state at a probability of $1 - \eta$. The emission matrix is calculated as following. Let M denote the totally number of symbols in the event stream. For any hidden states in the episode, it has a delta function emission. Whenever it is visited (right alignment of the first element in the episode, left alignment for the left elements in the episode), it will generate the same observation symbol. For any noise hidden states, it emits any of the symbols from the M observation symbols with a uniform distribution at probability $\frac{1}{M}$.

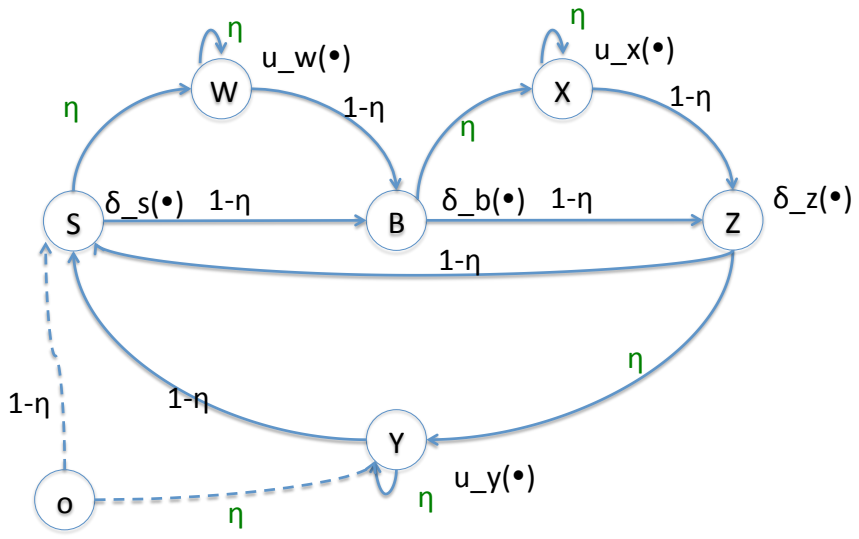


Figure 4.3: States Transition of Episode Generating HMM (EGH).

Theorem 1[77] is very important. It guarantees that the more frequent an episode inside the sequence, the probability of most likely sequence is larger. Its proof is explained in detail in [77].

THEOREM 1. *Let $D_Z = X_1, ..., X_K$ is the given sequence data, ε is the symbol set, and the size of these symbols is M. Given two frequent N-node episodes α and β with frequency f_α and f_β. Their corresponding EGH is Λ_α and Λ_β. The most likely state sequence for episode α and β are q^*_α and q^*_β. The noise parameters of these two EGH are η_α and η_β. Assume both of these noise parameters are less than $\frac{M}{M+1}$, we have (1) if $f_\alpha > f_\beta$, the $P(D_Z, q^*_\alpha|\Lambda) > P(D_Z, q^*_\beta|\Lambda)$ (2) if $P(D_Z, q^*_\alpha|\Lambda) > P(D_Z, q^*_\beta|\Lambda)$, $f_\alpha > f_\beta$*

In this occupancy prediction application, we change the calculation of frequency. In our work, the frequency of episode is based on day. If an episode happens more than once, then we calculate it as only once.

4.5.3 Mixture EGH

Mixture EGH has been discussed fully in previous work [77]. The mixture EGH model has the advantage of giving different weight for each EGH model for prediction. From the previous subsection, we obtain whether an episode occurs in a certain day. Let $D_Z = \{X_1, ..., X_K\}$ denote the K days data set. $F = \{\alpha_1, ...\alpha_J\}$ denote the frequent episodes in the dataset D_Z. EGH Λ_{α_j} corresponds to one of the frequent episode α_j. Let Λ_Z denote the mixture model. The likelihood of D_Z under the mixture model is written as Equation 4.1.

$$Pr(\Lambda|Z) = \prod_{i=1}^{K} P[X_i|\Lambda_Z] \tag{4.1}$$

$$= \prod_{i=1}^{K} (\sum_{j=1}^{J} \theta_j P[X_i|\Lambda_{\alpha_j}]) \tag{4.2}$$

where θ_j is the mixture coefficient of Λ_{α_j} and it subjects to $\sum_{j=0}^{J} \theta_j = 1$

The parts inside Equation 4.1 are additive, the coefficients θ are computed by EM algorithm. The detailed description is algorithm in the appendix section. During the initialization part of the EM algorithm, the episode frequency over the times series T is calculated. Specific frequent episodes ended with target event 'Z' are selected. Optionally, we could add special constraints on episodes starting with certain event type 'S'. In the expectation step, one key part is the likelihood value of each episode α_j in time series X_i. The likelihood value is computed as Equation 4.3. Then Bayes rules is applied to compute the new coefficient θ_{new}.

$$Pr(X_i|\Lambda_{\alpha_j}) = (\frac{\eta_{\alpha_j}}{M})^{|X_i|} (\frac{1-\eta_{\alpha_j}}{\eta_{\alpha_j}/M})^{|\alpha_j| f_{\alpha_j}(X_i)} \tag{4.3}$$

In the step of maximization, we can update the objective value based on Equation 4.1. and until it converges, i.e., the difference of two consecutive objective values is smaller than a threshold.

4.5.4 Predict When the Target Event Occurs

Target event prediction has been studied in [79]. But it only predicts whether a target event will occur or not. It never considers when the target event will happen. The occupancy prediction problem involves three sub-problems: 1) whether the target event un-occupancy Z will appear; 2) when the target event Z starts; 3) when the target event Z ends.

For stream prediction for target event, refer to algorithm in [79]. This section emphasizes on the last two sub-problems, predicting when the person leaves Z_{leave} or comes back Z_{back} after we already know target event Z will surely happen. Note that Z_{leave} corresponds to $Z.start$, when the Z event starts. Z_{leave} corresponds to $Z.end$, when the Z event ends. This prediction algorithm is described in algorithm 1.

After running episode mining and mixture EGH model, we have obtained all the frequent episodes $F = \langle \alpha_1, ..., \alpha_J \rangle$, the corresponding EGH $\Lambda_{\alpha_j}, j = 1...J$ with noise parameter η_j, and the mixture models Λ_Z with coefficients θ_j. We use the coefficient of these mixture models for leave time and back time prediction. The algorithm 1 uses partial day of test data s, all the frequent episodes $epis$, the frequent episodes F in partial day of s, the mixture model Λ_Z, to predict when the person leaves or comes back home. The partial day is cut by partial index $pIndex$. Each day is cut into three phases, before the person gets up; after the person gets up but before the person leaves home; after the person comes back home.

In the first two phases, before the person leaves, the PDF leave time and back time are calculated from line 2-4. Usually before a person gets up, there is only one frequent episode named 'SZ'. After the person gets up, he/she has a lot of activities at home, there are several frequent episodes mined before the person leaves home. In case there are several frequent episodes in lines 6-12, the leave time and back time of each episode is checked whether they are in a range of PDF value in the past. If yes, the mean value of these episodes are recorded from line 14-17. If there are several frequent episodes with corresponding EGH, the leave time Z_{leave} and back time Z_{back} is the weighted mean leave time and back time of each episode in lines 18-21.

In the third phase, after the person leaves home, we already know when the person leaves home Z_{leave} but needs to predict when the person comes back Z_{back}. If the person has come back, that means Z_{back} is not equal to Z_{leave}. We don't need to do anything. If the person hasn't come back, that means Z_{back} equals to Z_{leave}. The back time is the past weighted back time from line 28-34.

4.6 Experiment Results

We have conducted experiments on three datasets. The data is obtained by monitoring two adults' 24-hour activities at home through RFID. The monitored events denote when the person is at which room. For instance, we may know at 7:00am person 1 is in the kitchen.

There are totally three datasets, Study 10 from 02/10/2014 to 02/21/2014, Study 11 from 01/29/2014

Algorithm 1 Target Event Occurs Time Prediction Algorithm

Input: partial day cut point $pIndex$, $s[1 : pIndex]$ is known, and $s[pIndex : 96]$ for prediction; partial day event stream $s = \langle E_1, ..., E_{pIndex} \rangle$; all the episodes $epis$, $\forall E \in \alpha$ and $\forall \alpha \in epis$, E has $E.start$ and $E.end$; frequent episodes $F = \langle \alpha_1, ..., \alpha_J \rangle$ inside $s[1 : pIndex]$; EGH model Λ_{α_j} with noise parameter η_j where $j = 1...J$ EGH mixture model coefficients $\Theta = \langle \theta_1, ..., \theta_J \rangle$; the slot number noise parameter ϵ ; target event Z

Output: Predict target event leaving time Z_{leave} and back time Z_{back}

if $Z \notin s$ **then**
  choose episodes $\alpha.ev = 'SZ'$
  $Z_{leave} = \frac{\sum_1^K Z.start}{|\alpha|}$
  $Z_{back} = \frac{\sum_1^K Z.end}{|\alpha|}$
  if $len(F) > 1$ **then**
    for $\alpha_j \langle e_1, ..., e_j \rangle \in F$ **do**
      if $pIndex \in (\alpha_j.ev[-2].start - \epsilon, \alpha_j.ev[-2].start + \epsilon)$ **then**
        $leaveMap[\alpha_j.ev] = \alpha_j.leave$
      end if
      if $pIndex \in (\alpha_j.ev[-2].end - \epsilon, \alpha_j.ev[-2].end + \epsilon)$ **then**
        $backMap[\alpha_j.ev] = \alpha_j.back$
      end if
    end for
    if $len(leavingMap)! = 0$ **then**
      $leaveSlotMap[\alpha_j.ev] = \frac{\sum_1^K leavingMap.get(k)}{K}$
      $backSlotMap[\alpha_j.ev] = \frac{\sum_1^K backMap.get(k)}{K}$
    end if
    if $K = len(leaveSlotMap) > 1$ **then**
      $Z_{leave} = \frac{\sum_{k=1}^K leaveSlotMap.get(k) * \theta_k}{K}$
      $Z_{back} = \frac{\sum_{k=1}^K backSlotMap.get(k) * \theta_k}{K}$
    end if
  end if
end if
if $Z \in s$ **then**
  $Z_{leave} = s.firstindex[Z]$
  $Z_{back} = s.lastindex[Z]$
  if $Z_{leave} = Z_{back}$ **then**
    for $\alpha_j \langle e_1, ..., e_j \rangle \in F$ **do**
      if $Z_{leave} \in [\alpha_j.ev[-2].end - \epsilon, \alpha_j.ev[-2].end + \epsilon]$ **then**
        $backSlotMap[\alpha_j.ev] = \alpha_j.leave$
      end if
    end for
    $Z_{back} = \frac{\sum_{k=1}^K backSlotMap.get(k) * \theta_k}{K}$
  end if
end if
Output the slot number when the person leaves Z_{leave} and comes back Z_{back}

to 02/07/2014, and Study 14 from 12/09/2013 to 12/21/2013.

Firstly, we define the occupancy at home. Sometimes, the person goes out for several minutes then comes back. This type of un-occupancy event is *not* what we concern. Note that our goal is to turn on or off the HVAC 30 minutes earlier. We define the unoccupied as follows: 1). the person leaves the outside-front or outside-back for more than 30 minutes. 2). the person stays in the living room or dining room for more than 9 hours without any other activities. 3). the gap between any two events is more than 30 minutes.

Secondly we preprocess the data by deleting those invalid events, where the end time happens earlier than the start time. Then we sort the events according to the start time. Another data cleaning is that we delete the events whose duration is less than 2 minutes. This is very important preprocessing. A person may walk in the hallway or between the dining room and the kitchen very frequently. What we are concerned with is that where the person really is. If the person dwells in an exact room for less than 2 minutes, that means he/she just passes by the room, but does not to stay there to do something for long time.

After cleaning all the data, we apply three approaches, kNN, PDF based and mixture EGH time prediction model. Similar to [113], we organize one day's date into 96 15-minutes intervals with mixture EGH model. Then we split the test date into three phases: (1) before getting up, (2) after getting up and before going out, (3) after going out and before coming back. Thus our problem becomes to predict when the person going out and when the person comes back. Corresponding to these four phases, we adopt three different approaches. For stage (1), the probability density function of going out and combing back event is calculated. For stage (2), a duration-constraint episode mining and episode generative HMM is applied. For stage (3), the probability density function of backing time based on the time-constrains going out time is computed.

The results are shown in Figure 4.4, Figure 4.5, and Figure 4.6. Each of these figures include four sub-figures. Each sub-figure describes the occupancy or un-occupancy precision, recall, and f-measure results of a person on a certain day. The blue represents the EGH mixture model. The green represents the PDF model. The red means the kNN model. The x-axis means the number of known 15-minutes chunks of partial day. For instance, at $x = 20$, we already know the $20 * 15$ minutes' data, we need to predict whether the left 76 15-minutes the person is inside home or not. The y-axis denotes the precision value, recall value and f-measure value from top to down.

Figure 4.4 (a) and (b) in study 10 shows the person1 occupancy and un-occupancy results. mixEGH has the highest precision, recall and measure on the test day 02/20/2014 for both occupancy and un-occupancy. The other two approaches are competitive and kNN performs better than PDF after the person comes back home after slot 72. Looking into the original data, we find that person1 actually comes later than usual in the training dataset. In such case, EGH mixture model performs best from the perspective of precision, recall and f-measure.

In Study 11, mixtureEGH performs better if adjusting coefficient of mixture EGH model. Figure 4.5 (a) and (b) in study 11 shows the occupancy and un-occupancy comparison results of person2 on 02/04/2014. kNN performs the best from the perspective of precision. The precision

of mixture EGH is very interesting. It's very low before the person2 gets up ,and grows high after the person2 leaves home, and becomes competitive with kNN after the person2 comes back. The reason lies in that, person2 slept late that day after 12:00am. Before sleep, person2 stayed in the kitchen for some time. The frequent episode KZ usually happens in the morning time instead of midnight. The mixture EGH performance is not good before person2 gets up. kNN performs best because kNN approach doesn't consider the actual place inside the room and it averages the occupancy status in the past most similar 5 days. Even if the person slept late, he/she still got up regularly. EGH mixture model performs competitive with kNN.

In Study 14, Figure 4.6 (a) and (b) describe the case of person1 on 12/18/2013. Similar to Figure 4.5 (a) and (b), mixture EGH model doesn't perform well before the person1 gets up and the reason keeps the same. The person1 slept late and some confused episodes are generated.

Generally speaking, the mixtureEGH helps predict when a person leaves home during the period of sleeping and leaving home and its performance is competitive to the kNN approach.

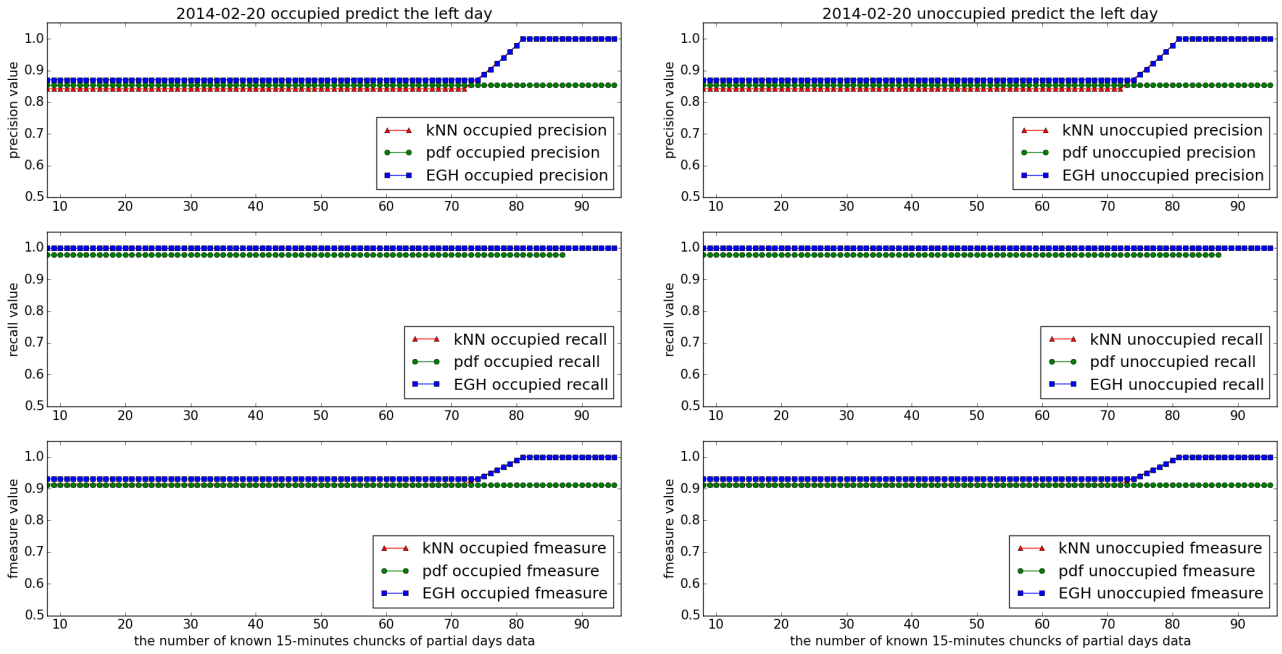
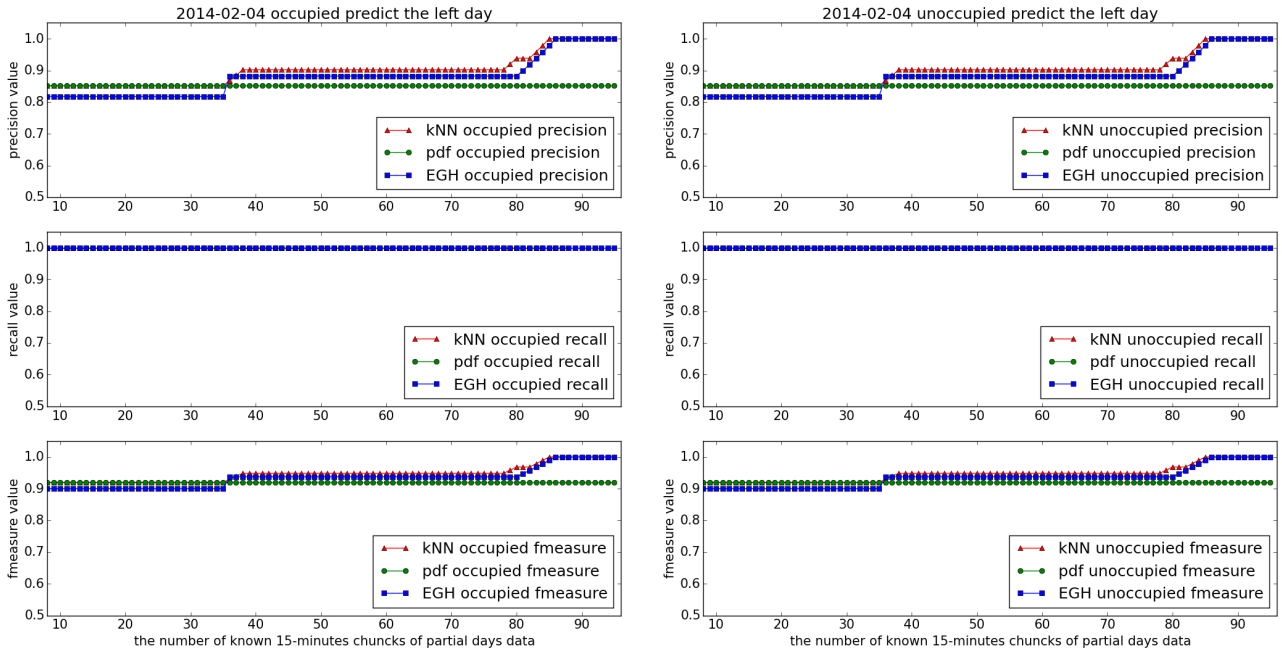


Figure 4.4: Study 10 Precision recall and f-measure comparison of three approaches. (a) person1 occupied 02/20/2014 (b) person1 unoccupied 02/20/2014 (c) person2 occupied 02/17/2014 (b) person2 unoccupied 02/17/2014



(a) (b)

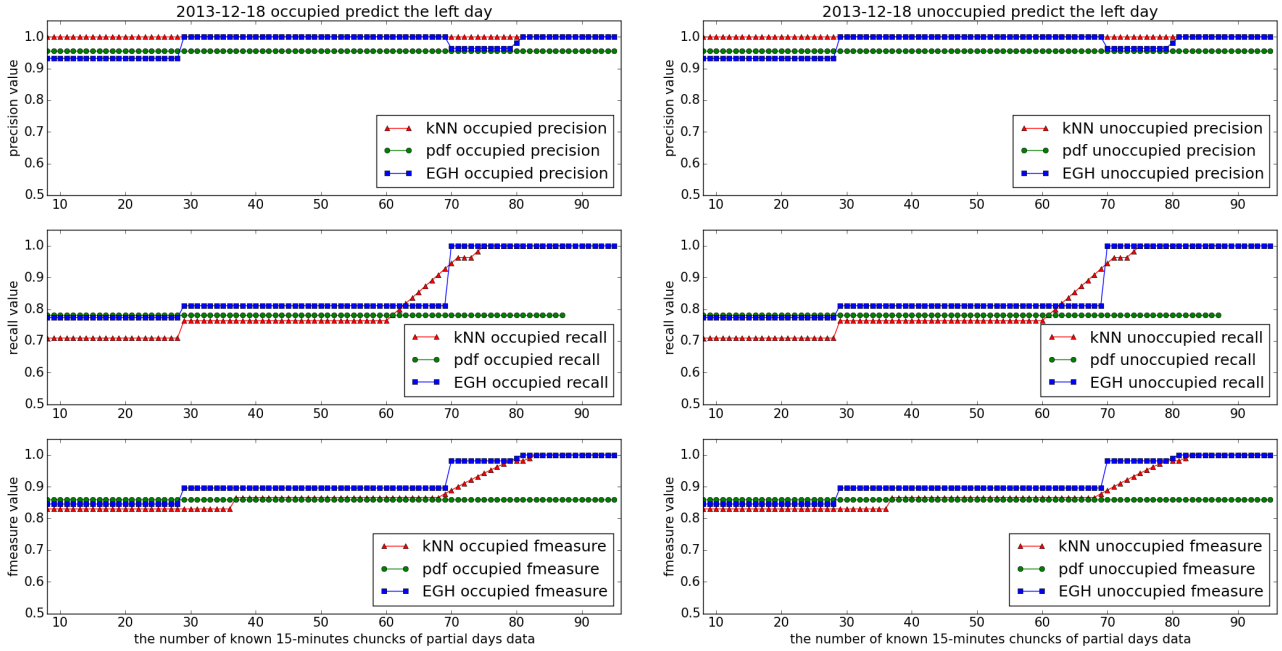
Figure 4.5: Study 11 Precision recall and f-measure comparison of three approaches. (a) person2 occupied 02/04/2014 (b) person2 unoccupied 02/04/2013

4.7 Conclusion

Residential occupancy prediction is a hot research topic on controlling the HVAC. The accuracy of occupancy prediction influences the comfortability of persons inside the home and energy saving.

In this paper, we propose the mixture EGH model and compare it with two other benchmark models, probability density function and kNN approach. The results show that it generally performs better than kNN to predict the occupancy and un-occupancy states in the workdays. The mixture model predicts well for the period of after person getting up and before person going out. The coefficient of the episode generative HMM models helps predict the exact leaving time. However in the case of abnormal events, kNN performs good because it can average the historical data. Even if there is an abnormal day, kNN can leverage it.

In the future work, we will improve the occupancy accuracy by exploring generating the mixture EGH prediction parameters automatically. So far, the hand adjust of parameters is needed for better performance of EGH. Further, we will continue on the holiday occupancy prediction. The occupant and un-occupancy patterns for these days are completely different. For some days, the person never goes out. Therefore the occupancy prediction can only be obtained from the date information.



(a) (b)

Figure 4.6: Study 14 Precision recall and f-measure comparison of three approaches. (a) person1 occupied 12/18/2014 (b) person1 unoccupied 12/18/2013

4.8 Appendix

Algorithm 2 Gap Constraint Episode Mining on Dwelling Events

for all event type A **do**
  Initialize $waits(A) = \phi$
end for
for all $\alpha \in C$ **do**
  $prev = \phi$
  for $1 \leq i \leq N$ **do**
    Create $node$ with $node.visited = False$; $node.episode = \alpha$; $node.index = i$; $node.pre = pre$;$node.next = \phi$
    if $i = 1$ **then**
      Add $node$ to $wait(\alpha[1])$
    end if
    if $prev! = \phi$ **then**
      $prev.next = node$
    end if
  end for
end for
for $i = 1 : n$ **do**
  for all $node \in waits(E_i)$ **do**
    set $accpeted = false$
    set $\alpha = node.episode$
    set $j = node.index$
    set $tlist = node.list$
    if $j < N$ **then**
      for all $tval \in tlist$ **do**
        if $(t_i - tval.init) > \alpha.t_{high}[j]$ **then**
          remove $tval$ from $tlist$
        end if

Algorithm 3 Gap Constraint Episode Mining on Dwelling Events - Part 2

```
            if j = 1 then
              update accepted = true
              tval.init = t_i
              add tval to tlist
              if node.visited = false then
                update node.visited = true
                add node.next to waits(α[j + 1])
              end if
            else
              for all prev_tval ∈ node.pre.tlist do
                if t_i − pre_tval ∈ (α.t_low[j − 1], α_high[j − 1]) then
                  Update accepted = true
                  update tval.init = t_i
                  add tval to tlist
                  if node.visited = false then
                    update node.visited = true
                    if node.index ≤ N − 1 then
                      add node.next to waits(α[j + 1])
                    end if
                  end if
                else
                  if t_i − prev_tval > α.t_high[j − 1] then
                    remove prev_tval from node.prev.tlist
                  end if
                end if
              end for
            end if
          end for
        end if
        if accepted = true and node.index = N then
          update α.freq = α.freq + 1
          set temp = node
          while temp! = φ do
            update temp.visited = false
            if temp.index! = 1 then
              remove temp from waits(α[temp.index])
            end if
            update temp = temp.next
          end while
        end if
      end for
    end for
```

Algorithm 4 EM Algorithm for mixture EGH

Input: day episode matrix, each element e_{ij} records for each day whether an episode j happens in day i; frequent episodes $F = \{\alpha_1, ..., \alpha_J\}$; symbol set ε; threshold γ

Output: the parameters for mixture EGH $\Lambda_Z = \{(\Lambda_{\alpha_j}, \theta_j), j = 1, ..., J\}$

calculate the number of episodes J, and number of days K

calculate all ηs' threshold value $mThreshold = \frac{M}{M+1}$

{initialize all the thetas to be $\frac{1}{J}$} {calculate the total frequency for each episode over training time series} { calculate the eta value}

for $0 \leq j \leq J$ **do**

  $\theta[j] = 1/J$

  $episodeFreq[j] = \sum_i^K e_{ij}$

end for

select those frequent episodes starting with 'S' and ending with 'Z' and separate these episodes by workday or holiday

{calculate eta for each episode}

for $0 \leq j \leq J$ **do**

  $\eta[j] = 1 - episodeLen[j] * episodeLen/T$

end for

{likelihood prediction of each episode j in the kth day}

for $0 \leq i \leq K$ **do**

  for $0 \leq j \leq J$ **do**

    $likelihood_{ij} = \frac{1-\eta[j]}{\eta[j]/M}^{episodeLen[j]*e_{ij}}$

  end for

end for

{calculate the obj value based on J, K, $likelihood_{ij}$ and θ}

while $newObj - obj > \gamma$ **do**

  $\theta_{new} = []$

  for $0 \leq l \leq J$ **do**

    $temp = 0$

    for $0 \leq j \leq K$ **do**

      $temp = temp + \frac{\theta_l * likelihood_{il}}{\sum_0^J \theta_j * likelihood_{ij}}$

    end for

    $\theta_{new}[l] = temp/K$

  end for

  calculate the newObj

  if $newObj - obj > \gamma$ **then**

    $obj = newObj$

    $\theta_{new} = \theta$

  end if

end while

Output $\Lambda_Z = \{(\Lambda_{\alpha_j}, \theta_j), j = 1, ..., J\}$

Chapter 5

Proposal: Indoor Activities Tracking

5.1 Indoor Activities Tracking

In the previous work, episode mining was applied in energy disaggregation and time-gap constrained episode mining and a corresponding EGH model is utilized to predict occupancy. Episode mining and its extension can be widely used in the area of sustainability for mining patterns and special prediction target.

Non-invasive indoor activities tracking is an interesting topic. It's a basis for smart home applications, such as monitoring the patient or elderly people, controlling the electrical devices based on the room occupancy. Most of such tracking system reply on installed cameras or microphones, or carried powered devices, such as ultrasonic RF transceiver. These devices are considered by some people to be invasive on the privacy. Non-invasive resident identification and indoor activity tracking was studied in the papers of[54, 120]. In the paper of [120], the authors install a ultrasonic distance sensor on the beam of the door to capture the height of a person when he/she walking through the door. According to the difference of height, each person can be identified at which room when staying at home. Based on it, [54] installs ultrasonic sensors above each doorway to estimate which room the person will go. The author utilizes the height of a person and walking direction. The challenges lies in two problems. The person identification is based on the height. Therefore the height difference among people at home is supposed to be greater than 7cm. The walking direction estimation based on the gait is not so accurate because there are maybe multiple paths instead of one.

So far, the research on noninvasive indoor activities tracking just begins. There are still a lot of questions that need to be answered. First is on how to estimate the room occupancy of each person at home more effectively? Second, how to improve the room occupancy prediction accuracy? Third, how to model a probabilistic time series model for prediction?

5.2 Approach

To solve the above indoor activities tracking problem, we propose to use the episode mining approach for the room location prediction. Each person inside the home is identified by one feature height. And he/she generates a time series with walking direction and doorway ID. Then we can apply the time or gap episode mining by constraints on the walking direction events. The order of mined episodes can be defined by us. For instance, if the door is closed. The door should be opened before a person goes through it. After that, the episode mining approach is applied to mine the frequent episodes. These frequent patterns with time give us the training data for target event prediction.

Unlike the occupancy prediction problem, there are several target events instead of one. We apply the mixture episode generative HMM model and calculate the probability of staying in each room. Since a person can only be in one room at home, we can select the room with the highest probability as the result.

We can briefly describe the problem as below. Each doorway has an ID. On it, an ultrasonic distance sensor is installed. Therefore for each doorway over a period of time, a time series data on height measurement and walking direction is recorded. The events in the doorjamb are of four types $e_i \in E = \{t_i, d_i, h_i, v_i\}$, where t_i is the timestamp, d_i is the doorway ID, h_j is the height measurement and v_i is the walking direction at the time of t. The target is to estimate the room occupancy of each person inside the home over a period of time $s_i \in S = (r1_i, r2_i)$ where $r1_i$ denotes that person 1 is at the room location at time i.

5.3 Evaluation

We already know the ground truth, which is when a person stays in each room in the building. We can compare the results of our approach and other approaches presented in [54] with the ground truth data. After calculating the precision, recall, and f-measure values of these two approaches, we can judge which method performs better.

Bibliography

[1] Buildings energy data book. *Energy Efficiency and Renewable Energy*, 2014.

[2] K. Abed-Meraim, W. Qiu, and Y. Hua. Blind system identification. *Proceedings of the IEEE*, 1997.

[3] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, 1995.

[4] M. Akbar and D. Khan. Modified nonintrusive appliance load monitoring for nonlinear devices. In *Proceedings of the IEEE International Conference on Multitopic*, 2007.

[5] AlphaLab. Emf meter.

[6] A. Alrazgan, A. Nagarajan, A. Brodsky, and N. Egge. Learning occupancy prediction models with decision-guidance query language. In *Proceedings of the 44th Hawaii International Conference on System Sciences (HICSS)*, 2011.

[7] Amprobe. Temperature meter.

[8] K. Anderson, M. Berges, A. Ocneanu, D. Benitez, and J. Moura. Event detection for non intrusive load monitoring. In *Proceedings of the IEEE Industrial Electronics Convferene(IECON)*, 2012.

[9] K. Anderson, A. Ocneanu, D. Benitez, D. Carlson, A. Rowe, and M. Berges. BLUED: a fully labeled public dataset for event-based non-intrusive load monitoring research. In *Proceedings of the 2nd KDD Workshop on Data Mining Applications in Sustainability (SustKDD)*, 2012.

[10] M. Baranski and J. Voss. Nonintrusive appliance load monitoring based on an optical sensor. In *Proceedings of the IEEE Power Tech Conference*, 2003.

[11] M. Baranski and J. Voss. Detecting patterns of appliances from total load data using a dynamic programming approach. In *Proceedings of the 4th IEEE International Conference on Data Mining*, 2004.

[12] M. Baranski and J. Voss. Genetic algorithm for pattern detection in nialm systems. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 2004.

[13] S. Barker, A. Mishra, D. Irwin, E. Cecchet, and P. Shenoy. Smart an open data set and tools for enabling research in sustainable homes. In *Proceedings of the 2nd KDD Workshop on Data Mining Applications in Sustainability (SustKDD)*, 2012.

[14] N. Batra, M. Gulati, A. Singh, and M. B. Srivastava. It's different: Insights into home energy consumption in india. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, 2013.

[15] A. Beltran and A. Cerpa. Optimal hvac building control with occupancy prediction. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, 2014.

[16] M. Berges, E. Goldman, H. Matthews, and L. Soibelman. Learning systems for electric consumption of buildings. In *ASCI International Workshop on Computing in Civil Engineering*, 2009.

[17] M. Berges, E. Goldman, H. Matthews, and L. Soibelman. Enhancing electricity audits in residential buildings with nonintrusive load monitoring. *Journal of Industrial Ecology*, 2010.

[18] M. Berges, E. Goldman, L. Soibelman, and K. Anderson. User-centric non-intrusive electricity load monitoring for residential buildings. *Journal of Industrial Ecology*, 2010.

[19] C. Bishop and N. Nasrabadi. *Pattern recognition and machine learning*. 2006.

[20] Blekin. 2013.

[21] T. Blumensath and M. Davies. Shift-invariant sparse coding for single channel blind source separation. *SPARS*, 2005.

[22] W. Chan, A. So, and L. Lai. Harmonics load signature recognition by wavelets transforms. In *Proceedings of the IEEE International Conference on Electric Utility Deregulation and Restructuring and Power Technologies*, 2000.

[23] C. Chang, H.and Lin. A new method for load identification of nonintrusive energy management system in smart home. In *Proceedings of the IEEE 7th International Conference on e-Business Engineering (ICEBE)*, 2010.

[24] H. Chang, C. Lin, and H. Yang. Load recognition for different loads with the same real power and reactive power in a non-intrusive load-monitoring system. In *Proceedings of the 12th International Conference on Computer Supported Cooperative Work in Design*, 2008.

[25] H. Chang, H. Yang, and C. Lin. Load identification in neural networks for a non-intrusive monitoring of industrial electrical loads. *Computer Supported Cooperative Work in Design IV*, 2008.

[26] D. Chen, S. Barker, A. Subbaswamy, D. Irwin, and P. Shenoy. Non-intrusive occupancy monitoring using smart meters. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, 2013.

[27] Y. Chen, C. Chen, W. Peng, and W. Lee. Mining correlation patterns among appliances in smart home environment. In *Advances in Knowledge Discovery and Data Mining*. 2014.

[28] Y. Chen, W. Peng, and W. Lee. A novel system for extracting useful correlation in smart home environment. In *Proceedings of the IEEE 13th International Conference on Data Mining Workshops (ICDMW)*, 2013.

[29] B. Chiu, E. Keogh, and S. Lonardi. Probabilistic discovery of time series motifs. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge discovery and Data Mining*, 2003.

[30] C. Chow and J. N. Tsitsiklis. The complexity of dynamic programming. *Journal of Complexity*, 1989.

[31] K. Collins, M. Mallick, G. Volpe, and W. Morsi. Smart energy monitoring and management system for industrial applications. In *Proceedings of the IEEE Electrical Power and Energy Conference (EPEC)*, 2012.

[32] D. Cook and L. Holder. Sensor selection to support practical use of health-monitoring smart environments. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2011.

[33] D. J. Cook, A. S. Crandall, B. L. Thomas, and N. Krishnan. Casas: a smart home in a box. *Computer*, 2013.

[34] R. Cox, S. Leeb, S. Shaw, and L. Norford. Transient event detection for nonintrusive load monitoring and demand side management using voltage distortion. In *Proceedings of the 21st Annual IEEE Applied Power Electronics Conference and Exposition*, 2006.

[35] J. Dai, M. Li, S. Sahu, M. Naphade, and F. Chen. Multi-granular demand forecasting in smarter water. In *Proceedings of the 13th International Conference on Ubiquitous Computing*, 2011.

[36] M. E. Davies and C. J. James. Source separation using single channel ica. *Signal Processing*, 2007.

[37] H. Dong, B. Wang, and C. Lu. Deep sparse coding based recursive disaggregation model for water conservation. In *Proceedings of the 23rd international Joint Conference on Artificial Intelligence*, 2013.

[38] R. Dong, L. Ratliff, H. Ohlsson, and S. Sastry. A dynamical systems approach to energy disaggregation. In *Proceedings of the IEEE 52nd Annual Conference on Decision and Control (CDC)*, 2013.

[39] S. Drenker and A. Kader. Nonintrusive monitoring of electric loads. *IEEE Computer Applications in Power*, 1999.

[40] J. Duan, D. Czarkowski, and Z. Zabar. Neural network approach for estimation of load composition. In *Proceedings of the International Symposium on Circuits and Systems(ISCAS)*, 2004.

[41] L. Engineering. Mindstorms.

[42] V. Erickson, M. Carreira-Perpiñán, and A. Cerpa. Occupancy modeling and prediction for building energy management. *ACM Transactions on Sensor Networks (TOSN)*, 2014.

[43] V. Erickson and A. Cerpa. Occupancy based demand response hvac control strategy. In *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, 2010.

[44] L. Farinaccio and R. Zmeureanu. Using a pattern recognition approach to disaggregate the total electricity consumption in a house into the major end-uses. *Energy and Buildings*, 1999.

[45] M. Figueiredo, B. Ribeiro, and A. de Almeida. Electrical signal source separation via nonnegative tensor factorization using on site measurements in a smart home. *IEEE Transactions on Instrumentation and Measurement*, 2014.

[46] J. Froehlich, E. Larson, S. Gupta, G. Cohn, M. Reynolds, and S. Patel. Disaggregated end-use energy sensing for the smart grid. *IEEE Pervasive Computing*, 2011.

[47] S. Giri and M. Berges. A study on the feasibility of automated data labeling and training using an EMF sensor in NILM platforms. In *Proceedings of the 2012 International EG-ICE Workshop on Intelligent Computing*, 2012.

[48] H. Gonçalves, A. Ocneanu, and M. Bergés. Unsupervised disaggregation of appliances using aggregated consumption data. 2011.

[49] D. Görür and C. Rasmussen. Dirichlet process Gaussian mixture models: choice of the base distribution. *Journal of Computer Science and Technology*, 2010.

[50] S. Gupta, M. Reynolds, and S. Patel. Electrisense: single-point sensing using emi for electrical event detection and classification in the home. In *Proceedings of the 12th ACM International Conference on Ubiquitous computing*, 2010.

[51] A. Hambley. *Electrical engineering principles and applications*. 2006.

[52] G. Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 1992.

[53] T. Hassan, F. Javed, and N. Arshad. An empirical investigation of vi trajectory based load signatures for non-intrusive load monitoring. *IEEE Transactions on Smart Grid*, 2014.

[54] T. Hnat, E. Griffiths, R. Dawson, and K. Whitehouse. Doorjamb: unobtrusive room-level tracking of people in homes using doorway sensors. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, 2012.

[55] D. Huang, M. Thottan, and F. Feather. Designing customized energy services based on disaggregation of heating usage. In *Proceedings of the IEEE Innovative Smart Grid Technologies (ISGT)*, 2013.

[56] E. Instruments. Light sensor.

[57] N. Instruments. Ni-9239.

[58] A. Jain, M. Murty, and P. Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 1999.

[59] Y. Jin, E. Tebekaemi, M. Berges, and L. Soibelman. Robust adaptive event detection in non-intrusive load monitoring for energy aware smart facilities. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

[60] Y. Jin, E. Tebekaemi, M. Berges, and L. Soibelman. A time-frequency approach for event detection in non-intrusive load monitoring. In *Proceedings of SPIE*, 2011.

[61] M. Johnson and A.S. Bayesian nonparametric hidden semi-markov models. *CoRR*, 2012.

[62] M. Kearns. Computational complexity of machine learning. 1990.

[63] J. Kelly and W. Knottenbelt. Disaggregating multi-state appliances from smart meter data. *SIGMETRICS*, 2012.

[64] J. Kelly and W. Knottenbelt. Metadata for energy disaggregation. 2014.

[65] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han. Unsupervised disaggregation of low frequency power measurements. In *Proceedings of the SIAM International Conference on Data Mining*, pages 747–758, 2011.

[66] Y. Kim, T. Schmid, Z. Charbiwala, and M. Srivastava. Viridiscope: design and implementation of a fine grained power monitoring system for homes. In *Proceedings of the 11th International Conference on Ubiquitous computing*, 2009.

[67] W. Kleiminger, S. Santini, and F. Mattern. Smart heating control with occupancy prediction: how much can one save? In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, 2014.

[68] C. Koehler, B. Ziebart, J. Mankoff, and A. Dey. Therml: occupancy prediction for thermostat control. In *Proceedings of the 2013 ACM international Joint Conference on Pervasive and Ubiquitous Computing*, 2013.

[69] T. Kolda and B. Bader. Tensor decompositions and applications. *SIAM review*, 2009.

[70] J. Kolter, S. Batra, and A. Ng. Energy disaggregation via discriminative sparse coding. In *Proceedings of Neural Information Processing Systems*, 2010.

[71] J. Kolter and T. Jaakkola. Approximate inference in additive factorial hmms with application to energy disaggregation. In *International Conference on Artificial Intelligence and Statistics*, 2012.

[72] J. Kolter and M. Johnson. Redd: a public data set for energy disaggregation research. In *Proceedings of the Workshop on Data Mining Applications in Sustainability (SIGKDD)*, 2011.

[73] H. Kuhns, M. Roberts, and B. Bastami. Closure rules for energy load disaggregation. Pittsburg, U.S.A., 2012.

[74] P. Lai, M. Trayer, S. Ramakrishna, and Y. Li. Database establishment for machine learning in nilm. Pittsburg, U.S.A., 2012.

[75] H. Lam, G. Fung, and W. Lee. A novel method to construct taxonomy electrical appliances based on load signatures. *IEEE Transactions on Consumer Electronics*, 2007.

[76] C. Laughman, K. Lee, R. Cox, S. Shaw, S. Leeb, L. Norford, and P. Armstrong. Power signature analysis. *IEEE Power and Energy Magazine*, 2003.

[77] S. Laxman, P. Sastry, and K. Unnikrishnan. Discovering frequent episodes and learning hidden markov models: A formal connection. *IEEE Transactions on Knowledge and Data Engineering*, 2005.

[78] S. Laxman, V. Tankasali, and R. White. Stream prediction using a generative model based on frequent episodes in event sequences. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008.

[79] S. Laxman, V. Tankasali, and R. White. Stream prediction using a generative model based on frequent episodes in event sequences. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008.

[80] K. Lee. *Electric load information system based on non-intrusive power monitoring*. PhD thesis, Massachusetts Institute of Technology, 2003.

[81] K. Lee, S. Leeb, L. Norford, P. Armstrong, J. Holloway, and S. Shaw. Estimation of variable-speed-drive power consumption from harmonic content. *IEEE Transactions on Energy Conversion*, 2005.

[82] S. Lee, D. Ahn, S. Lee, R. Ha, and H. Cha. Personalized energy auditor: Estimating personal electricity usage. In *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2014.

[83] T. Lee, M. Lewicki, M. Girolami, and T. Sejnowski. Blind source separation of more sources than mixtures using overcomplete representations. *IEEE Signal Processing Letters*, 1999.

[84] S. Leeb, S. Shaw, and J. Kirtley Jr. Transient event detection in spectral envelope estimates for nonintrusive load monitoring. *IEEE Transactions on Power Delivery*, 1995.

[85] M. Lewicki and T. Sejnowski. Learning overcomplete representations. *Neural computation*, 2000.

[86] Y. Li and S. Osher. Coordinate descent optimization for l1 minimization with application to compressed sensing; a greedy algorithm. *Inverse Problem Imaging*, 2009.

[87] J. Liang, S. Ng, G. Kendall, and J. Cheng. Load signature study part i: Basic concept, structure, and methodology. *IEEE Transactions on Power Delivery*, 2010.

[88] R. Lukaszewski, K. Liszewski, and W. Winiecki. Methods of electrical appliances identification in systems monitoring electrical energy consumption. In *Proceedings of the IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)*, 2013.

[89] D. Luo, L. Norford, S. Leeb, and S. Shaw. Monitoring hvac equipment electrical loads from a centralized location- methods and field test results. 2002.

[90] S. Mahmoud, A. Lotfi, and C. Langensiepen. Occupancy pattern extraction and prediction in an inhabited intelligent environment using narx networks. In *Intelligent Environments*, 2010.

[91] S. Mahmoud, A. Lotfi, and C. Langensiepen. Behavioural pattern identification and prediction in intelligent environments. *Applied Soft Computing*, 2013.

[92] S. Makonin, F. Popowich, L. Bartram, B. Gill, and I. Bajic. AMPds: A public dataset for load disaggregation and eco-feedback research. In *Proceedings of the IEEE Electrical Power and Energy Conference (EPEC)*, 2013.

[93] G. Mambayashi. Noise measurements of the residential power line. In *Proceedings of the IEEE International Symposium on Power Line Communications and Its Applications*, 1997.

[94] C. Manna, D. Fay, K. Brown, and N. Wilson. Learning occupancy in single person offices with mixtures of multi-lag markov chains. In *Proceedings of the IEEE 25th International Conference on Tools with Artificial Intelligence (ICTAI)*, 2013.

[95] H. Mannila, H. Toivonen, and A. Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1997.

[96] M. Marceau and R. Zmeureanu. Nonintrusive load disaggregation computer program to estimate the energy consumption of major end uses in residential buildings. *Energy Conversion and Management*, 2000.

[97] H. Matthews, L. Soibelman, M. Berges, and E. Goldman. Automatically disaggregating the total electrical load in residential buildings: a profile of the required solution. *Proceedings of the Intelligent Computing in Engineering (ICE08)*, 2008.

[98] A. Milioudis, G. Andreou, V. Katsanou, K. Sgouras, and D. Labridis. Event detection for load disaggregation in smart metering. In *Proceedings of the IEEE/PES Innovative Smart Grid Technologies Europe (ISGT EUROPE)*, 2013.

[99] A. Monacchi, D. Egarter, W. Elmenreich, S. D'Alessandro, and A. Tonello. Greend: an energy consumption dataset of households in italy and austria. In *Proceedings of the IEEE International Conference on Smart Grid Communications*, 2014.

[100] H. Murata and T. Onoda. Applying kernel based subspace classification to a non-intrusive monitoring for household electric appliances. In *Artificial Neural Networks?ICANN*. 2001.

[101] Y. Nakano and H. Murata. Non-intrusive electric appliances load monitoring system using harmonic pattern recognition-trial application to commercial building. In *Proceedings of the International Conference on Electrical Engineering*, 2007.

[102] L. Norford and S. Leeb. Non-intrusive electrical load monitoring in commercial buildings based on steady-state and transient load-detection algorithms. *Energy and Buildings*, 1996.

[103] T. Onoda, G. Rätsch, and K. Müller. Applying support vector machines and boosting to a non-intrusive monitoring system for household electric appliances with inverters. 2000.

[104] A. Pardo, V. Meneu, and E. Valor. Temperature and seasonality influences on spanish electricity load. *Energy Economics*, 2002.

[105] O. Parson, S. Ghosh, M. Weal, and A. Rogers. Non-intrusive load monitoring using prior models of general appliance types. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, Toronto, Canada, 2012.

[106] S. Patel, T. Robertson, J. Kientz, M. Reynolds, and G. Abowd. At the flick of a switch: detecting and classifying unique electrical events on the residential power line. In *Proceedings of the 9th International Conference on Ubiquitous computing*, 2007.

[107] D. Patnaik, M. Marwah, R. Sharma, and N. Ramakrishnan. Sustainable operation and ,anagement of data center chillers using temporal data mining. In *Proceedings of the 15th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining*, 2009.

[108] D. Patnaik, P. Sastry, and K. Unnikrishnan. Inferring neuronal network connectivity from spike data: A temporal data mining approach. *Scientific Programming*, 2008.

[109] J. Powers, B. Margossian, and B. Smith. Using a rule-based algorithm to disaggregate end use load profiles from premise-level data. *Computer Applications in Power*, 1991.

[110] S. Rollins and N. Banerjee. Using rule mining to understand appliance energy consumption patterns. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2014.

[111] J. Roos, I. Lane, E. Botha, and G. Hancke. Using neural networks for non-intrusive monitoring of industrial electrical loads. In *Proceedings of the IEEE 10th Anniversary Advanced Technologies in Instrumentation and Measurement Technology*, 1994.

[112] M. Schmidt and M. Mørup. Nonnegative matrix factor 2-d deconvolution for blind single channel source separation. In *Independent Component Analysis and Blind Signal Separation*. 2006.

[113] J. Scott, A. Brush, J. Krumm, B. Meyers, M. Hazas, S. Hodges, and N. Villar. Preheat: controlling home heating using occupancy prediction. In *Proceedings of the 13th International Conference on Ubiquitous Computing*, 2011.

[114] H. Shao, M. Marwah, and N. Ramakrishnan. A temporal motif mining approach to unsupervised energy disaggregation. In *Proceedings of the 1st International Workshop on Non-Intrusive Load Monitoring*, 2012.

[115] H. Shao, M. Marwah, and N. Ramakrishnan. A temporal motif mining approach to unsupervised energy disaggregation: Applications to residential and commercial buildings. Bellevue, U.S.A., 2013.

[116] S. Shaw. *System identification techniques and modeling for nonintrusive load diagnositics*. PhD thesis, 2000.

[117] S. Shaw, S. Leeb, L. Norford, and R. Cox. Nonintrusive load monitoring and diagnostics in power systems. *IEEE Transactions on Instrumentation and Measurement*, 2008.

[118] H. Song, G. Kalogridis, and Z. Fan. Short paper: Time-dependent power load disaggregation with applications to daily activity monitoring. In *Proceedings of the IEEE World Forum on the Internet of Things (WF-IoT)*, 2014.

[119] D. Srinivasan, W. Ng, and A. Liew. Neural-network-based signature recognition for harmonic source identification. *IEEE Transactions on Power Delivery*, 2006.

[120] V. Srinivasan, J. Stankovic, and K. Whitehouse. Using height sensors for biometric identification in multi-resident homes. In *Pervasive Computing*. 2010.

[121] V. Srinivasan, J. Stankovic, and K. Whitehouse. Fixturefinder: Discovering the existence of electrical and water fixtures. In *Proceedings of the 12th International Conference on Information Processing in Sensor Networks*, 2013.

[122] Y. Su, K. Lian, and H. Chang. Feature selection of non-intrusive load monitoring system using stft and wavelet transform. *Proceedings of the IEEE 8th International Conference on e-Business Engineering*, 2011.

[123] F. Sultanem. Using appliance signatures for monitoring residential loads at meter panel level. *IEEE Transactions on Power Delivery*, 1991.

[124] K. Suzuki, S. Inagaki, T. Suzuki, H. Nakamura, and K. Ito. Nonintrusive appliance load monitoring based on integer programming. In *Proceedings of the IEEE SICE Annual Conference*, 2008.

[125] P. Technology. Ta041.

[126] TED. Ct products, 2010.

[127] TrendPoint. Power meter.

[128] S. Uttama-Nambi, T. Papaioannou, D. Chakraborty, K. Aberer, et al. Sustainable energy consumption monitoring in residential settings. In *Proceedings of the 2nd IEEE INFOCOM Workshop on Communications and Control for Smart Energy Systems (CCSES)*, 2013.

[129] E. Vogiatzis, G. Kalogridis, and S. Denic. Real-time and low cost energy disaggregation of coarse meter data. In *Proceedings of the 4th IEEE/PES Innovative Smart Grid Technologies Europe*, 2013.

[130] B. Wang, H. Dong, A. Boedihardjo, and C. Lu. A hierarchical probabilistic model for low sample rate home-use energy disaggregation. 2013.

[131] W. Wichakool, A. Avestruz, R. Cox, and S. Leeb. Modeling and estimating current harmonics of variable electronic loads. *IEEE Transactions on Power Electronics*, 2009.

[132] T. Wu and M. Srivastava. Low-cost appliance state sensing for energy disaggregation. In *Proceedings of the 4th ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, 2012.

[133] M. Wytock and J. Kolter. Contextually supervised source separation with application to energy disaggregation. 2014.

[134] H. Yang, H. Chang, and C. Lin. Design a neural network for features selection in non-intrusive monitoring of industrial electrical loads. In *Proceedings of the 11th International Conference on Computer Supported Cooperative Work in Design*, 2007.

[135] D. Yankov, E. Keogh, J. Medina, B. Chiu, and V. Zordan. Detecting time series motifs under uniform scaling. In *Proceedings of the 13th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining*, 2007.

[136] M. Zeifman. Disaggregation of home energy display data using probabilistic approach. *IEEE Transactions on Consumer Electronics*, 2012.

[137] M. Zeifman and K. Roth. Nonintrusive appliance load monitoring: review and outlook. *IEEE Transactions on Consumer Electronics*, 2011.

[138] M. Zeifman and K. Roth. Viterbi algorithm with sparse transitions (vast) for nonintrusive load monitoring. In *IEEE Symposium on Computational Intelligence Applications In Smart Grid (CIASG)*, 2011.

[139] M. Zeifman, K. Roth, and J. Stefan. Automatic recognition of major end-uses in disaggregation of home energy display data. In *IEEE International Conference on Consumer Electronics (ICCE)*, 2013.

[140] A. Zoha, A. Gluhak, M. Imran, and S. Rajasegarar. Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. *Sensors*, 2012.