

Temporal Mining Approaches for Smart Buildings Research

Huijuan Shao

Dissertation 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 and Applications

Narendran Ramakrishnan, Chair
Manish Marwah
Anil Vulkanti
Chang-Tien Lu
Bodicherla Aditya Prakash

September 20, 2016
Arlington, Virginia

Keywords: Data mining, Sustainability, Energy disaggregation, Occupancy prediction.
Copyright 2016, Huijuan Shao

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. Moreover, the rapid *urbanization* we are witnessing requires optimized energy distribution. This dissertation focuses on two sub-problems in improving energy conservation; *energy disaggregation and occupancy prediction*. Energy disaggregation attempts to separate the energy usage of each circuit or each electric device in a building using only aggregate electricity usage information from the meter for the whole house. The second problem of occupancy prediction can be accomplished using non-invasive indoor activity tracking to predict the locations of people inside a building. We cast both problems as *temporal mining problems*. We exploit motif mining with constraints to distinguish devices with multiple states, which helps tackle the energy disaggregation problem. Our results reveal that motif mining is adept at distinguishing devices with multiple power levels and at disentangling the combinatorial operation of devices. For the second problem we propose time-gap constrained episode mining to detect activity patterns followed by the use of a mixture of episode generating HMM (EGH) models to predict home occupancy. Finally, we demonstrate that the mixture EGH model can also help predict the location of a person to address non-invasive indoor activities tracking.

This work is supported in part by Virginia Tech and HP Labs.

Temporal Mining Approaches for Smart Buildings Research

Huijuan Shao

(GENERAL AUDIENCE ABSTRACT)

This dissertation uses data analytics techniques to address energy problems in commercial and residential buildings.

One topic is energy disaggregation, which is to discover electricity usage of each device by only analyzing data from existing installed power meters. This research provides the following results: what devices use majority of power; how much electricity is used for each device; when each device is turned on or off. With this technique, no extra monitor is needed thus the installing cost is low, which makes it applicable to millions of office buildings and homes. This work can benefit electricity companies and all electricity customers.

Another research topic occupancy prediction is to forecast when a house will be occupied. Since heating, venting, and air conditioner (HVAC) consumes the largest power usage at home, automatic operations of HVAC according to house occupancy are crucial for saving electricity without sacrificing the comfort of people inside the building. By analysis, I can predict whether a home will be occupied in the next thirty minutes. Then the HVAC can be turned off before its vacancy and turned on ahead of its occupancy. By integrating this technique with an automatic HVAC device, any home installed with it can lead a more energy-saving, convenient and comfortable life.

In conclusion, this research contributes to saving energy and protecting environment.

Dedication

To my parents Yuncheng Shao and Congxiu Zou, my beloved husband Yaowei Li, and my kids Elaine, Franklin and George.

Acknowledgments

First, I would like to give thanks to my advisor Dr. Naren Ramakrishnan. Under his wise direction and continuous encouragement, I have been able to persevere with this research and complete my dissertation. Dr. Ramakrishnan often gave me excellent advice on my research, which inspired me to gather courage to push my research forward. He has also provided insightful and thoughtful ideas on many things and has given me sound advice on several key problems that arose during my PhD program. It has been my honor to study under him at the Discovery Analytics Center.

Second, I want to thank Dr. Manish Marwah from HP labs. He is an expert in sustainability and has tons of experience in the research of smart buildings. He directed me on how to conduct research step by step until I achieved the correct results. During my work for this project, some of the most exciting and happiest times were while I was working with Dr. Marwah.

In addition, I appreciate much on the opportunity to cooperate with Dr. B. Aditya Prakash from the Discovery Analytics Center and Dr. Madhav Marathe, Dr. Anil Vullkanti, and Dr. Maleq Khan from VBI, Virginia Tech. Their advice on my research has been quite useful and impressive. Furthermore, I would thank my committee member Dr. Chang-tien Lu for giving me advice on energy disaggregation, especially the great help with two important exams for my PhD degree. I would not have been as successful without his help. Furthermore, I am thankful for Dr. Xinwei Deng and Dr. Hui Yi for giving me suggestions on my research from the perspective of statistics.

I am very grateful to have been able to study with my lab-mates. Their passionate discussion about research, their physical help and their spiritual encouragement during hard times has enabled me to appreciate the preciousness of friendship and the merits of human nature.

Most important of all, I have much appreciation for my husband, my parents, my dear sister Huirong Shao, my parents-in-law, and my friend Xiaoyan Yu's comprehensive support. It is only with their enthusiastic help and warm company that everything in my life was able to go smoothly so I could focus on my research and complete this dissertation.

Contents

List of Figures	vii
List of Tables	x
1 Introduction	1
1.1 Challenges in Smart Building Research	2
1.2 Goals of this Dissertation	3
1.3 Temporal Data Mining and the Probabilistic Model	5
1.4 Organization of this Dissertation	7
2 Survey of Energy Disaggregation	9
2.1 Introduction	9
2.1.1 Background	10
2.1.2 Challenges	11
2.1.3 Scope of this Survey	12
2.1.4 Organization	13
2.2 A Primer on AC power	13
2.2.1 Electricity Transmission	13
2.2.2 Circuits and Devices	14
2.2.3 Voltage and Current	15
2.2.4 Real Power and Reactive Power	16
2.2.5 Harmonics	18

2.3	Formalisms	19
2.3.1	Definition	20
2.3.2	Technology Timeline	20
2.4	Disaggregation Features	21
2.4.1	AC Power Features	22
2.4.2	Steady state	22
2.4.3	Transient state	23
2.4.4	Features beyond Current and Voltage	28
2.5	Disaggregation Algorithms	29
2.5.1	Pre-processing Stages	30
2.5.2	Overview of Disaggregation Algorithms	31
2.5.3	Supervised Learning Algorithms	32
2.5.4	Unsupervised Learning Algorithms	43
2.5.5	Semi-supervised Learning Algorithms	49
2.6	Evaluation Methodology	50
2.6.1	Evaluation based on Events	50
2.6.2	Evaluation based on Time Series	51
2.6.3	Evaluation based on Combinational Metrics	52
2.6.4	Data Collection and Public Data Sets	52
2.7	Ongoing Research	54
2.8	Conclusion	54
3	Energy Disaggregation	58
3.1	Introduction	58
3.2	Background	59
3.3	Temporal Motif Mining	62
3.4	Evaluation	66
3.5	Experiments on REDD dataset	67
3.5.1	Disaggregation experiments	68

3.5.2	Comparison of Motif Mining and AFAMAP	68
3.6	Commercial Building Dataset	70
3.7	Discussion	71
4	Recursive Multivariate Piecewise Motif Mining to Disaggregation	72
4.1	Introduction	72
4.2	Prior Work	73
4.3	Disaggregation Formalism	74
4.4	Recursive Multivariate Piecewise Motif Mining	74
4.4.1	Piecewise Motif Mining	76
4.4.2	Encoding Events From Multiple Phases	77
4.5	Evaluation	78
4.6	Experiments	79
4.6.1	Electricity Disaggregation	80
4.6.2	Water Disaggregation and Constraints	83
4.7	Conclusion	84
5	Occupancy Prediction	85
5.1	Introduction	85
5.2	Related Work	86
5.3	Problem Formulation	87
5.4	Temporal Mining Mixture Model	87
5.4.1	Time-gap Constraint Episode Mining	88
5.4.2	Mixture EGH	90
5.4.3	Predict When the Target Event Occurs	92
5.5	Experiment Results	92
5.5.1	Occupancy Prediction of Individuals	93
5.5.2	Occupancy Prediction of Residential Buildings	95
5.5.3	Limitations of Mixture EGH Model	96

5.5.4	House Occupancy Prediction 30 Minutes Ahead with Hybrid Approach . . .	96
5.6	Conclusion	97
6	Conclusion	103
	Bibliography	106

List of Figures

1.1	Smart Buildings Schema.	2
1.2	Overall Framework for Smart Building Issues.	6
2.1	(a) Aggregate power input to a disaggregation algorithm. (b) Disaggregated information about devices and their power usage patterns.	10
2.2	Electricity generation and transmission to residential and commercial buildings. . .	14
2.3	Example of a circuit in (a) residential building and (b) commercial building.	15
2.4	Three phase power waveform.	15
2.5	AC Circuit of basic loads: resistor, inductor, and capacitor (courtesy: [54]).	16
2.6	Real and reactive power for different devices (courtesy: [55]).	17
2.7	Power Triangle.	18
2.8	Circuit 4 (a) Current Waveform and (b) Harmonics.	18
2.9	Timeline of features and algorithms proposed for energy disaggregation.	21
2.10	Category of (a) AC Power Features and (b) Non-AC Power Features.	22
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. .	24
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.	25
2.13	The eigenvalue of a circuit and dining room light.	26
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.	26
2.15	Switching-function for VSDs disaggregation (Courtesy:[140]).	27

2.16	(a) Baseline noise with newly added noise. (b) Noise Feature of a device.	28
2.17	Day of Week of Feature.	28
2.18	Neural Network Approach for Energy Disaggregation.	32
2.19	Transient Shape Decomposition and KNN Search	36
2.20	Probability density functions of three appliances neighboring by power draw.	37
2.21	Graphical model with M devices. (a) FHMM and (b) Difference FHMM.	45
2.22	AFAMAP Flowchart.	45
2.23	Motif Mining Example ([122]).	48
2.24	Four Types of Meters in a Building.	53
3.1	A residential setup for data collection.	60
3.2	Steady state transitions and transient features at startup.	60
3.3	Example of energy disaggregation.	61
3.4	Temporal motif mining framework for disaggregation.	63
3.5	Mining episodes from a symbolic time series.	64
3.6	Episode constraints.	64
3.7	Illustration of motif mining. Note that there are 3 non-overlapped occurrences of Episode 3.	65
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.	69
4.1	Recursive Multivariate Motif Mining Approach.	75
4.2	Multivariate Piecewise Motif Mining.	76
4.3	Encoding Events from Multiple Phases.	78
4.4	(a) On piecewise event and (c) Off piecewise event of heatingIndoor. heatingIndoor is disaggregated by motif mining the on event (b) and off event (d).	81

4.5	Disaggregating dryer and continuous variable load heatingOutdoor with multivariate motif mining. The on event of a device and the corresponding diffs in the two phases for (a) dryer, (b) water heater, (c) heatingOutdoor. (d) Disaggregating the toilet water use end with dynamic time-warping subsequence search. This Y-axis is water flow rate in 10000*liter/minute.	82
5.1	Occupancy Prediction Framework.	87
5.2	Time-gap constraint episode mining example.	89
5.3	States Transition of Episode Generating HMM (EGH).	91
5.4	Occupancy prediction precision, recall and f-measure comparison of three approaches of person1 on 02/20/2014 on Study10.	95
5.5	ROC curve of house occupancy prediction in Study10 (02/20/2014).	97
5.6	ROC curve of house occupancy prediction in Study11 (02/04/2014).	98
5.7	ROC curve of house occupancy prediction in (c) Study14 (12/20/2013).	100

List of Tables

2.1	Pre-processing algorithms with vector event feature extraction.	31
2.2	Categories of energy disaggregation algorithms.	56
2.3	Meters Used in Experiments.	57
3.1	Comparing Motif mining against AFAMAP on the REDD dataset.	70
3.2	Evaluation measures for commercial building disaggregation.	71
4.1	Power Levels, Standard Deviation of Power Levels, On/off Duration, Connected Phases and Disaggregation Results of Electricity Devices from Study10.	80
4.2	Water Flow Rate Levels of Water End Uses.	83
5.1	Datasets summary.	93
5.2	Precision Recall F-measure Comparison of Individual and Whole House Occupancy Prediction in Study 14.	94
5.3	Precision Recall F-measure of Individual and Whole House Occupancy Prediction in Study 14.	96

Chapter 1

Introduction

Global urbanization has become a trend, and this movement continuously accelerates. It has been projected that by the year 2030, cities will grow by 590,000 square miles and add an additional 1.47 billion people, so that 6 out of every 10 people on the planet will live in a city [121]. Key issues concerning urban populations; such as public health, sustainable use of limited energy resources, emergency preparedness, and societal stability will rise to the forefront of common discussion. Epidemiological analysis of public health data to analyze infection spread, traffic flow analysis via public transport and taxi cab data, election forecasting and e-governance and the like are just a few of the many examples of how dissecting data about urbanization can provide awareness and knowledge about the society we live in.

Sustainable energy is a critical issue in this era of urbanization. Energy consumption research has long been a topic of research interest, and new requirements are continuously being raised in the progress of city development. There is more demand for sustainable energy supply to cities than ever before. The problem of supplying increasing power to cities is two-fold. The immediate impact of urbanization has presented us with logistical problems. How do we meet the needs of the rapidly growing cities? Can we design *smart buildings* and *smart neighborhoods* that optimize energy consumption? Answering these questions is critical to the economy of the country and the economy of the society at large. The larger impact of supplying power to cities is one that the energy industry has on the environment. While renewable energy sources are being investigated, fossil fuels remain the primary source of energy supply. Climate change is a huge problem that will have an impact on humanity as we know it. It is a problem that needs a several-pronged attack to find a solution, and finding solutions to problems in energy consumption, thereby reducing the carbon footprint, would be a big step forward.

One of the essential resources in today's world is power and electricity. Needless to say, 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. [2]. Furthermore,

our dependence on electricity will continue to grow as emphasis shifts away from vehicles that run on fossil fuel to electric vehicles.

Smart building research focuses on the problems associated with energy consumption and maintaining the comfort and satisfaction of people, which mainly occurs in inside residential, commercial and industrial buildings [57]. As shown in Figure 1.1, by using an energy management and control system (EMCS), which includes electricity scheduling, chilled- and hot-water reset, a thermostat controller, and exporting sensor and controller data, the buildings can work intelligently to achieve the goal of energy efficiency by reducing energy costs, maintaining high-level comfort, and improving environment friendliness by reducing carbon emissions.

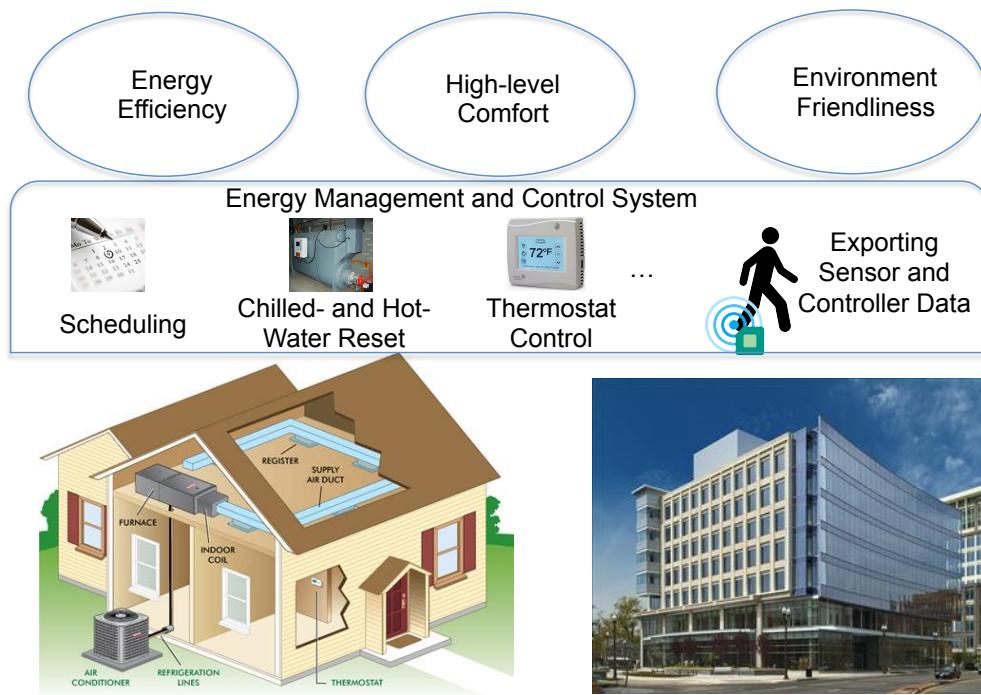


Figure 1.1: Smart Buildings Schema.

1.1 Challenges in Smart Building Research

Smart building aims to achieve significant energy saving by utilizing advanced energy technology. The intelligent energy management system monitors devices, controls appliances and conserves energy. To design an energy-saving building, we need to conduct several procedures to achieve this goal. Each change to existing procedures faces different problems.

- Collecting data. Determining how to collect accurate energy data using state-of-the-art tech-

nology is a fundamental problem. With the advent of sensor technology, wireless sensor nodes and active RFID tags are installed on devices or worn by people to gather comprehensive information [138]. Generally, energy consumption, water usage, gas expenditure, air temperature or noise status are monitored by meters or wireless sensors, then the collected data is transmitted to the computer and to storage. Identifying how to measure and collect this data precisely with low cost and minimal intrusion has always been a problem.

- Quantify opportunities. Analyzing data and discovering opportunities for energy optimization are another concern. First we need to identify the devices which consume a large amount of power or water or gas, when they are operated, and the regular patterns of energy consumption. Next, we need to figure out the possible opportunities to decrease energy consumption by evaluating the usage time and energy cost.
- Target and Schedule. Determining how to set a reachable target by scheduling the devices inside the building is an optimization subject. When to turn devices on or off, how to set the temperature inside the building without sacrificing the comfort of people, and how to make rules for device automation have been popular research problems.
- Track progress. To evaluate whether a scheduling plan is effective is an important issue. How to introduce new technology and update models in time is a challenge.

In this dissertation, I focus on quantifying opportunities in energy management. Other issues, such as how to collect energy consumption data, how to schedule the electrical devices or other water use ends, and how to refurbish models to optimize energy usage, are outside the scope of this work.

1.2 Goals of this Dissertation

In this dissertation, I endeavor to clarify the problems of how to qualify energy usage inside buildings. Then I implement existing approaches or enhance current methods to pinpoint energy usage as accurately as possible. This introductory section describes the research motivations, then explains our data-driven models for energy measurement, and concludes with a list of our contributions to smart building research.

Research motivation

Analytics has transformed the perception of the world we live in. Big Data is the lingua franca of the twenty-first century, and data science has become an essential lens through which decision making is seen. The massive amounts of data collected via several active and passive instruments positions us truly in an *age of information*. Data via Web 2.0, social media, the internet of things, traffic flow, gene sequencing, etc., in conjunction with the advancement of using data-mining and machine-learning to glean actionable insights, have influenced the world in designing policy, pricing products, launching political campaigns, and many other applications. The power

of *data analytics*, thus, is in its diversity. With respect to energy quantification, I introduce data-driven models to address several problems that I study in this thesis.

Topic 1: Energy Disaggregation

While people generally agree on the importance of conservation and usage curtailment, they often find it difficult 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 appliance/device, 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 [2].

One obvious approach to determining the breakdown of consumption is to install power meters in every circuit (and sub-circuit) to capture the consumption of individual devices in homes and offices. Such installation is costly and intrusive, making this option non-viable in practice. An alternate solution, called energy disaggregation or non-intrusive load monitoring (NILM), first proposed by Hart [55], is using 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 meter. 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 both offering challenging problems for data analytics and having practical relevance in a number of areas including sensor networks and building analytics.

Topic 2: Disaggregation to Electricity and Water

Water disaggregation [24] is similar to electricity disaggregation. By measuring the water flow rate, vibration or pressure in the hot and cold water, the flow trace of each water use end is inferred. This breakdown process is somewhat different from the electricity disaggregation because of the characteristics of water use ends.

Topic 3: Occupancy Prediction

An effective approach to save energy in homes is to efficiently use electric devices. In residential buildings, the biggest consumer of electricity is the HVAC (heating, ventilation, and air conditioning) system, which generally accounts for 54% of the building's electricity consumption [2]. Determining how to automatically start up and shut down the HVAC unit is thus a key problem. One solution is to predict the occupancy of a home by analyzing the activities of daily life inside the building. Based on the occupancy information, an automatic control system can then be installed to operate the HVAC.

1.3 Temporal Data Mining and the Probabilistic Model

Temporal datasets display a character of time-dependency. They are recorded frequently in smart buildings and build scenarios to infer the energy usage of people. Temporal data mining revolves around the techniques (algorithms) that enumerate structures, patterns, and signatures over temporal data (time series, for instance). A survey [82] has investigated several efficient techniques to discover the patterns in ordered data streams. The techniques used to discover significant patterns vary according to the dataset. One of the compelling patterns in temporal data mining is frequent episodes [99].

Frequent Episode Discovery

Frequent episode discovery is proposed in [99]. Given a sequence of events $< (E_1, t_1), \dots, (E_n, t_n) >$, where E_i denotes the i^{th} event at the time of t_i , the aim is to find temporal patterns (called *episodes*) that occur frequently in the long sequence. This episode is an ordered event collections. For instance an episode $(A \rightarrow B \rightarrow C)$ represents that event type A comes before event type B , which occurs before event type C . The occurring time of these events are unnecessary to be consequent. The frequency threshold is decided by a user. Several data mining algorithms have been researched to discover the frequent episodes [81, 99].

Motif Mining in Multi-variate Time Series Data

Motif mining is a temporal data mining technique that was initially proposed in [32] and [144] and extensively studied in [103, 113, 132]. The fundamental idea behind *motif mining* is that it symbolically encodes the numerical time series data. After this encoding, the symbols combine to form episodes in the data, resulting in patterns that can be mined. Furthermore, by combining domain-specific information and pattern mining techniques, we extract frequent, meaningful episodes from the symbolized time series.

Furthermore, when there are time series that describe the data, we employ *multi-variate motif mining* to find meaningful patterns. The algorithms for multi-variate temporal motif mining are similar to the univariate case, except that the symbolic encoding is represented as a vector. Therefore each time point in the data is represented as a vector of symbols, with each symbol corresponding to one of the several time series that represents the data. Now, the combination of these vector symbols forms episodes that can be mined from the multi-variate time series data. Again, by combining domain specific knowledge, we extract meaningful episodes from the data.

Episode Generating Hidden Markov Model (EGH)

A hidden Markov model is a ubiquitous construct to model time series data. It is used to represent probability distributions over a sequence of data. The hidden Markov model gets its name from two important properties. The observation (data point) is generated by a process whose state is *hidden* from the observer. Second, the state of this hidden process satisfies the *Markov* property that the current state is independent of all prior states. The *Episode Generating HMM*, researched in [81], connects the episodes with an HMM model and has a parameter to evaluate whether an episode

is frequent or not. A mixture of EGH describes a situation in which several frequent episodes are embedded into a time series, and can be used to predict whether a target symbol will be the next symbol in a time series.

Applications of Motif Mining and EGH

In our work we apply motif mining techniques, both univariate and multivariate cases, to energy disaggregation. By correlating episodic information with the switching on and off of appliances from time-series represented energy data, we are able to successfully determine the one-to-one mapping between a certain appliance and its usage patterns and time.

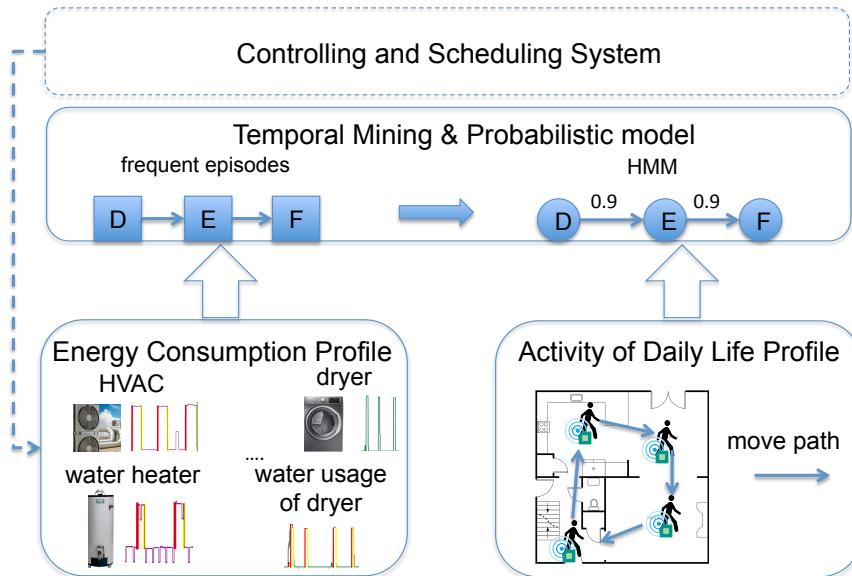


Figure 1.2: Overall Framework for Smart Building Issues.

Figure 1.2 gives a overall framework on the issues of the above smart building issues. By mainly collecting the energy consumption of profiles of electrical devices or water use ends, and the activities of daily life profiles of people inside buildings, we analyze the frequent episodes from these profiles with temporal mining and probabilistic models. Based on this analysis, a controlling and scheduling system can be developed to control individual devices.

The formulation of the EGH model lends itself to predicting occupancy in a residence or any building. We develop an EGH model based on occupancy prediction that can assist in automated turning on or off of the HVAC system, which can single-handedly reduce a good portion of the energy consumption footprint. We demonstrate that our algorithm can effectively forecast occupancy.

Contributions

In this work I make efforts to resolve some of the problems arising in smart building research with temporal mining approaches.

1. I present a survey on energy disaggregation from the perspective of data mining features and supervised, semi-supervised and unsupervised algorithms.
2. The temporal mining approach motif mining works effectively for energy disaggregation.
3. I utilize a multivariate piecewise motif mining algorithm for both energy disaggregation and water disaggregation.
4. The episode-based model Episode Generating HMM (EGH) and a mixture of EGHs performs well for event prediction in occupancy prediction.

1.4 Organization of this Dissertation

Smart buildings mainly commit to saving energy in the whole building automatically by introducing cutting-edge technologies. The challenges include issues such as minimizing electricity consumption, decreasing water usage, and automating the heating, ventilation and air conditioning (HVAC) without sacrificing comfort. In this dissertation, I aim to tackle these issues with data analytics approaches. The rest of this report is organized as described below.

Chapter 2 Survey of Energy Disaggregation: In Chapter 2, I survey the energy disaggregation in the past decade from the time it was first proposed, and summarize all the features utilized in the energy disaggregation research. In addition, I categorize the algorithms into classic data mining types. I opened the thesis by defining energy disaggregation formally. Briefly, the goal of energy disaggregation is to effectively break down appliance-level power consumption. I then conduct a complete survey of several mechanisms and techniques that employ data mining and machine learning to tackle energy disaggregation. While surveys have been conducted in the past, most of them are presented from an electrical engineering perspective. I survey works that use supervised and unsupervised learning. I compare and contrast a range of algorithms that have been used for energy disaggregation. The self-contained survey used here introduces the electrical engineering concepts that are required for data scientists to conduct research in the space, describe necessary tools and datasets to develop and test algorithms. I describe how experimental testbeds should be setup and how to record data from the necessary sensors and meters. I also present some promising directions for research in energy disaggregation from a data mining perspective. Essentially, we provide a *one-stop-shop* starting point for data mining practitioners to understand the problem scope of energy disaggregation, expose themselves to the problems, and then conduct research in the space.

Chapter 3 Energy Disaggregation: In order to curtail electricity consumption, we need to capture the usage patterns of the primary electrical devices inside a building. In this chapter, I focus on temporal mining algorithms using basic power features to separate the dominant electricity consumption devices. I performed experiments on a public residential data set and a commercial data set from HP Labs.

Chapter 4 Energy and Water Disaggregation: In addition to electricity, water is a source of energy spent in buildings. For the sake of water conservation, analyzing water usage patterns in terms of the leading ends of water use is necessary. Identifying the major ends of water use is extremely similar to electricity disaggregation. This chapter aims to make the temporal mining models applicable to both electricity and water disaggregation. I conduct an experiment on a dataset which incorporates both electricity and water usage, supplied by the University of Virginia. The results are presented in Chapter 4.

Chapter 5 Occupancy Prediction: HVAC automation is widely used in commercial and residential buildings. Determining how to maximize comfort and minimize electricity usage is a tough issue. In this chapter, I target to solve this problem by predicting the occupancy of a room or house. I run my temporal mining model as a predictive model to forecast the occupancy status of a given day based on the occupancy status of the previous days and the current day. I present my analysis in Chapter 5.

Chapter 6 Conclusion: This chapter summarizes my work on temporal pattern mining and the associated probabilistic model, and proposes paths for future smart building research.

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 2013, 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 savings 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 [55], 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 Background

Hart [55] 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.

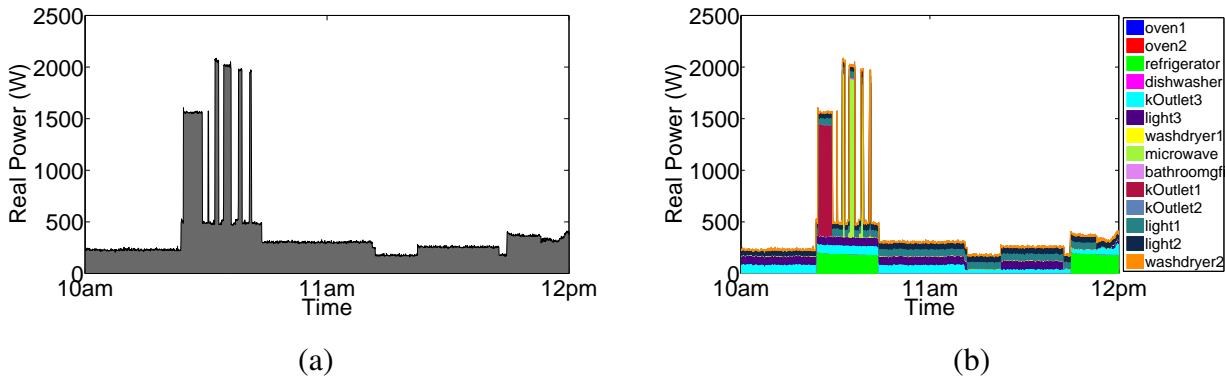


Figure 2.1: (a) Aggregate power input to a disaggregation algorithm. (b) Disaggregated information about devices and their power usage patterns.

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.

Another way to distinguish features is as transient vs steady state features. 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.

Yet 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 [55]. 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 [124]. Further, the raw current waveform [127], voltage waveform [79], and the transform of the current waveform [25] can also be exploited as features. Harmonics of non-linear devices have also been studied [25]. Non-AC power features, such as power line noises [112], time of day, and device correlations [67] 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 [124], support vector machines [112], neural networks [118], genetic programming [14], sparse coding [73], as well as combinations of supervised learning algorithms [106]. Optimization algorithms used in the area of energy disaggregation have been drawn from integer programming [131], dynamic programming [13], and the Viterbi algorithm [147]. Unsupervised learning algorithms have only been recently used in the last few years, and include hierarchical clustering [51], factorial hidden Markov models (FHMMs) [67], additive factorial approximate MAP (AFAMAP) [74], difference FHMM [111], and motif mining [123]. Semi-supervised learning algorithms [64, 79] 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 profile.
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, banks 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 [108]. 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., [146], [92], and [149], 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 [38], natural gas [49], and music [119]. 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 [23, 39, 90] but are quite distinct from disaggregation. In the case of blind source sep-

aration, the goal is to separate sources from at least as many observations whereas in the case of disaggregation, only one aggregate signal is provided. We do not cover these areas here.

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.

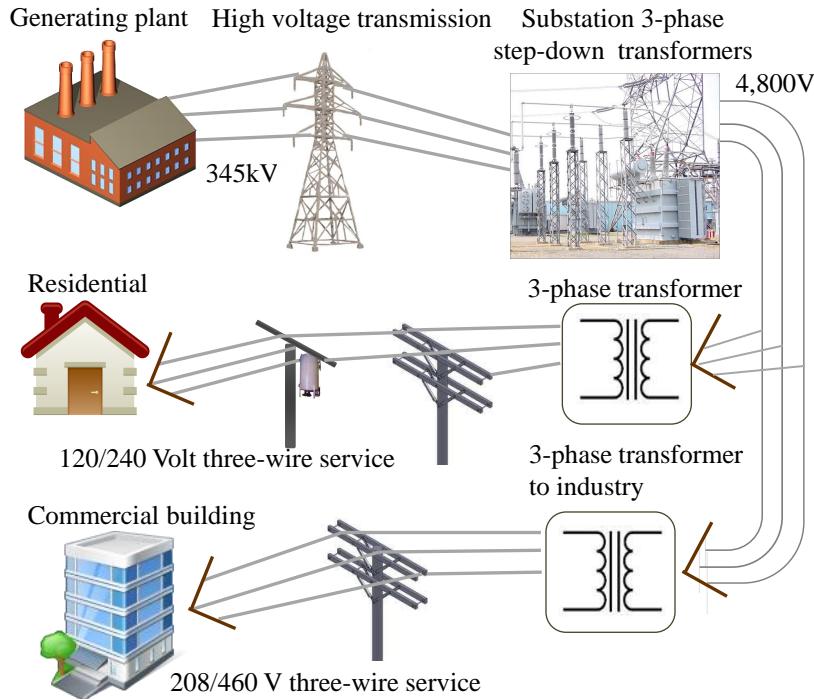


Figure 2.2: Electricity generation and transmission to residential and commercial buildings.

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: phase1 and phase2. Three circuits connect into these two phases. In the first circuit, two lights connect in series to phase1 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 (the lights thus power 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.

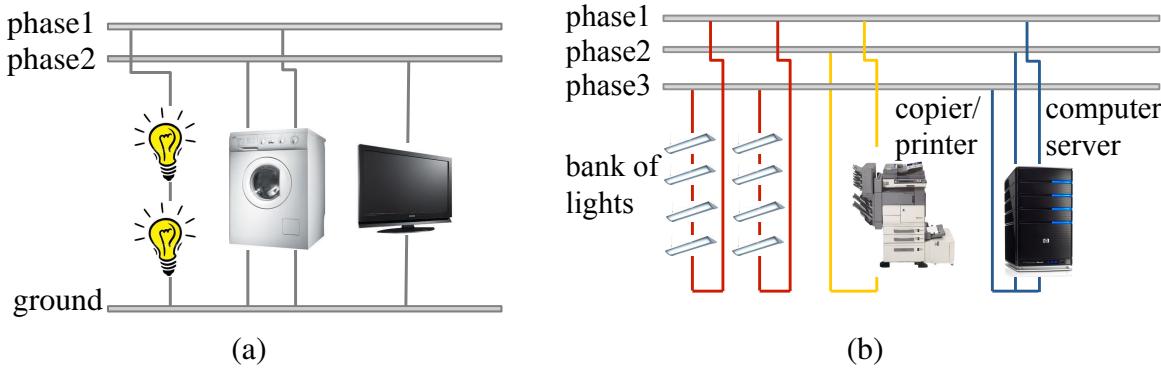


Figure 2.3: Example of a circuit in (a) residential building and (b) commercial building.

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 $\pi/3$.

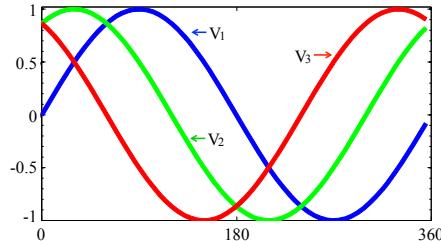


Figure 2.4: Three phase power waveform.

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.)

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

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 farad (F) where $V_s(t)$ and current $i(t)$ are AC voltage and current.

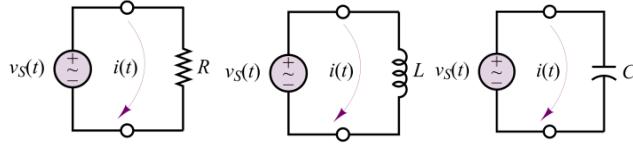


Figure 2.5: AC Circuit of basic loads: resistor, inductor, and capacitor (courtesy: [54]).

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 $V = IR$,

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

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

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

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

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

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

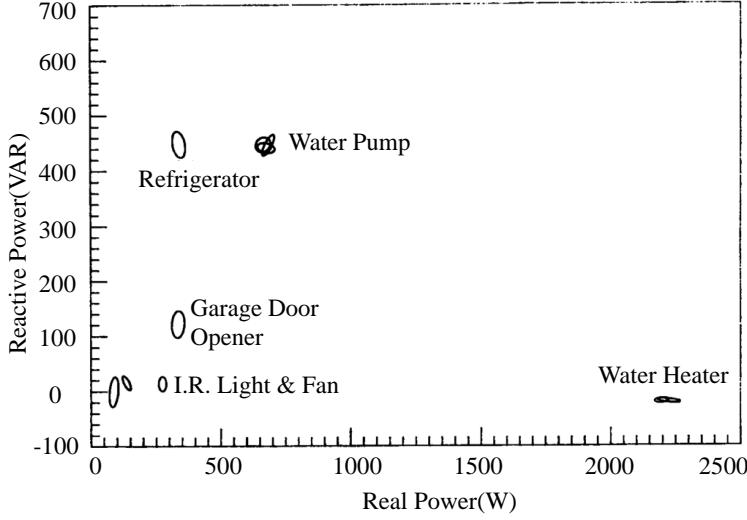


Figure 2.6: Real and reactive power for different devices (courtesy: [55]).

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, $v(t) = V \cos(\omega t - \theta_V)$ and current, $i(t) = I \cos(\omega t - \theta_I)$, 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. Then, the instantaneous real power at time t is given by:

$$P(t) = v(t) \cdot i(t) \quad (2.4)$$

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 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).

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

$$S = P_{av} + jQ \quad (2.5b)$$

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

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

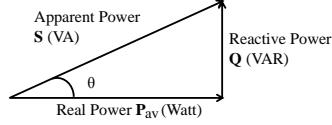


Figure 2.7: Power Triangle.

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).

$$P_{av} = R \cdot I^2 \quad (2.6a)$$

$$Q = (X_L - X_C) \cdot I^2 \quad (2.6b)$$

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 [11]. This (or any) type of wave-

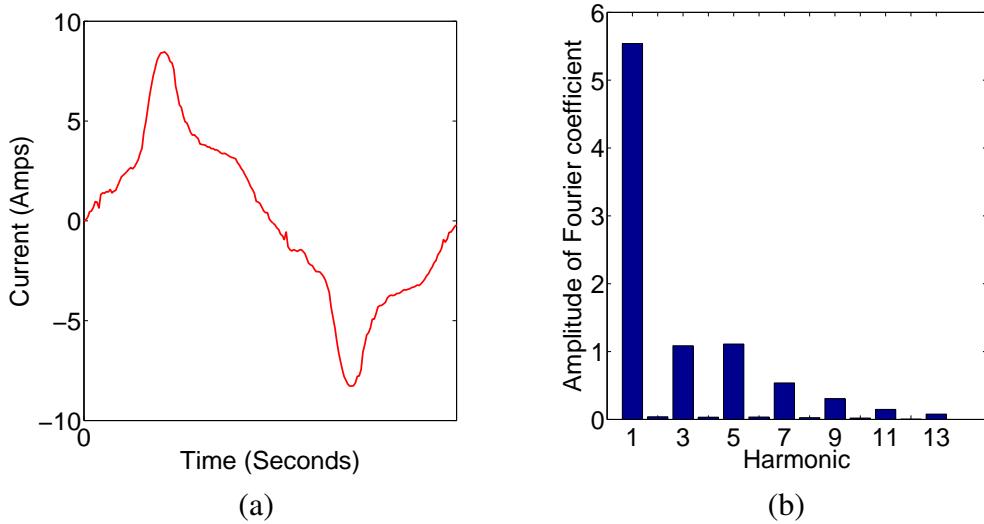


Figure 2.8: Circuit 4 (a) Current Waveform and (b) Harmonics.

form 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

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

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 Formalisms

Since Hart first defined the problem of energy disaggregation [55], 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 [55], 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 devices¹, the number of steady power consumption states of devices and their corresponding power levels, and non AC power features. There are thus numerous formulations of the energy disaggregation problem and these distinctions need to be considered while performances of algorithms are 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, i.e., the on/off state of each device is known. For instance, [92] assume that all devices are known and formulate the objective function as one of minimizing error between the disaggregated devices and the corresponding ground truth devices. In [74] 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 , and 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' . 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.

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 characteristics of each device, such as power levels are either unknown [51, 142] or assumed by the researchers to be fixed [64, 67, 74, 111, 123]. In spite of these difficulties, the disaggregation performance of unsupervised approaches may approach that of supervised approaches. Note that evaluation of any method, whether supervised or unsupervised, requires that ground truth information be available (number of devices and their on/off power states).

¹Disaggregation 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.

2.3.1 Definition

For our purposes, we will assume the following definition of the problem: Given an aggregate power consumption time series $Y = y_1, \dots, y_T$, and a set of power related and contextual features, $f = f_1, \dots, f_T$ over a period of time T , the problem is to estimate the disaggregated power consumption of M devices $\hat{X}_m = \hat{x}_1^{(m)}, \dots, \hat{x}_t^{(m)}, \dots, \hat{x}_T^{(m)}$, $m \in [1, M]$, such that a loss function on the sum of the power consumption of the M devices and the aggregate power consumption is minimal.

$$\min_{\hat{x}_t^{(m)}} \left\{ \sum_{t=1}^T \mathcal{L}_t \left(\sum_{m=1}^M \hat{x}_t^{(m)}, y_t \right) \right\} \quad (2.8)$$

where \mathcal{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 . \mathcal{L} is usually the $\mathcal{L}1$ -norm $\sum_{m=1}^M |\hat{x}_t^{(m)} - y_t|$ or the $\mathcal{L}2$ -norm $\sum_{m=1}^M (\hat{x}_t^{(m)} - y_t)^2$.

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

2.3.2 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 early stages of disaggregation research, 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 transforms of current, duration time of specific steady state of real power, the waveform of current or voltage, and current/voltage noise. In recent studies, algorithms have used eigenvalues of current, and device correlation information. As researchers explore new features to use for disaggregation, we must keep in mind that good results are not necessarily the only yardstick but ease of instrumenting sensors to capture such features is also a key objective.

Concomitantly, algorithms adopted in this area have experienced an accelerating progress; from supervised learning algorithms, including optimization algorithms and statistical models; to unsupervised learning and semi-supervised learning (see Fig. 2.9). Supervised learning employs each circuit/device's data as training data, which is laborious to collect. They include rule-based approaches; pair-wise matching and neural networks which were proposed in 1990s; k-nearest neighbor algorithms (kNN); support vector machines (SVM) and kernel based subspace classification (KSC); general likelihood ratio-based methods; auto-regression and moving average methods; radial basis function network methods (RBFN); decision trees; adaBoost; Bayesian classifiers; space coding; dynamic Bayesian networks; closure rules; Viterbi algorithms; dynamic programming, integer programming, and nonnegative tensor factorization. Since 2006, unsupervised and semi-supervised algorithms have become the preferred approach to identify devices. These include hierarchical clustering, factorial HMMs, approximate factorial additive MAP (AFAMAP),

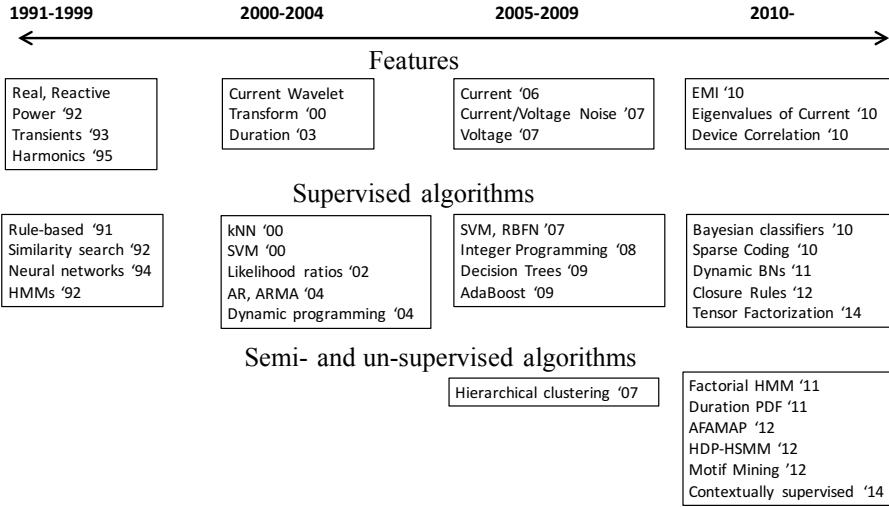


Figure 2.9: Timeline of features and algorithms proposed for energy disaggregation.

difference FHMM which adds prior knowledge of the device, hierarchical Dirichlet process hidden semi-Markov models, motif mining, and contextually supervised source separation. In the next two sections, we explain the features and the algorithms in detail.

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 introduced some of these features in Section 2.2 such as 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 (a) displays a 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. (In the below, a 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

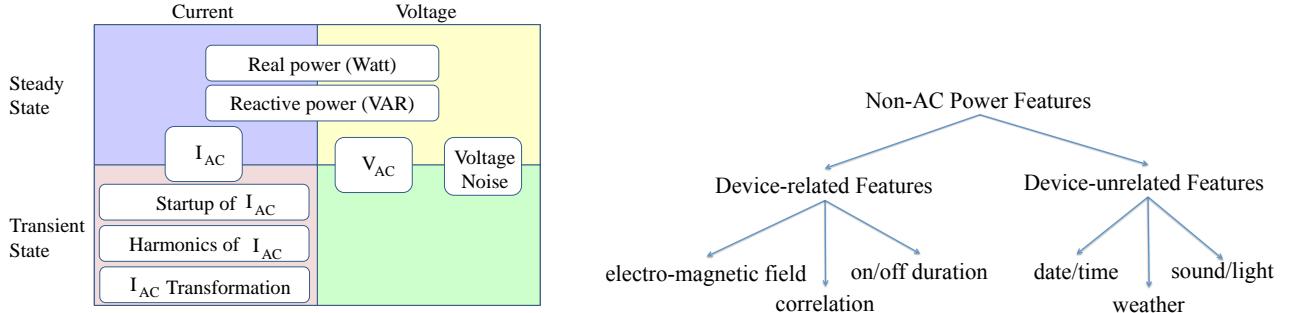


Figure 2.10: Category of (a) AC Power Features and (b) Non-AC Power Features.

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 can be classified into two categories: device-related and device-unrelated. Device-related category includes electromagnetic field (EMF); operational information such as on/off durations, and correlation between devices. 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-unrelated 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

2.4.2 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).

$$P_{av} = \frac{\sum_{t=1}^{f/60} v(t) \cdot i(t)}{f/60} \quad (2.9)$$

Real power is the most basic feature and used by almost all prior work in energy disaggregation [13, 14, 47, 55, 100, 115]. Reactive power is also widely used as a feature, e.g., in [42, 55, 80].

Figure 2.6 shows real power and reactive power features of different devices. For some devices real and reactive power are sufficient to distinguish between them. A refrigerator and water pump have similar reactive power but different real power; thus using the real power feature, we can separate them. A refrigerator and 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 [102], the slopes of both active and reactive power are extracted as vectors.

2.4.3 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 [11]. 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 $t \geq 0.45s$, the current comes again to a 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 (calculated using Equation (2.9)). Figure 2.11 (c) depicts the instantaneous real power. There are 200 points in each cycle. As can be seen, the instantaneous power changes very frequently.

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. 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 in integration with other features. [130] 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 [124], or transient power.

Current or voltage waveform: The current waveform I_{ac} , which can be simply read from high frequency recorded data, is a typical feature used to discriminate devices. The waveforms generated by non-linear devices are very different due to the waveform distortion introduced by each device.

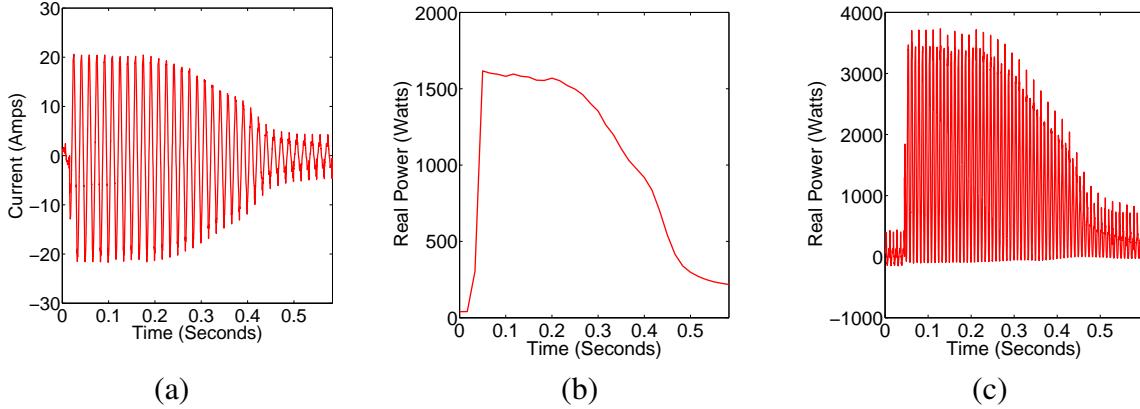


Figure 2.11: (a) Transient and Steady State of a Sinusoidal Current from a Refrigerator. Transient Shapes for a Refrigerator (b) Real Power and (c) Instantaneous Real Power.

Figure 2.12 (a-b) illustrates the current waveform of two devices—refrigerator and compressor—from the 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 the compressor is around 16 Amps.

Current or voltage waveform features have been applied in previous work. The unprocessed current waveform is regarded as a feature in [131]. 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 changes with noise. [43] analyze the current waveforms of eight devices and classify them as A/C, refrigerator, compressor, fan (VSD), elevator (converter), elevator (M/G set), fluorescent lights and computers. Similarly, standalone voltage waveform has been used as a feature in previous work. By analyzing voltage shapes, we can determine which device is on. While a combination of current and voltage waveforms has been proposed as features, [56] show that the disaggregation results are better when adopting either the current waveform or the voltage waveform-based features exclusively.

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 the voltage-current (VI) or current-voltage (IV) trajectory. This idea is useful because for dynamic devices such as an air conditioner, the current waveform may vary from cycle to cycle. Figure 2.12 (e-f) illustrates the current trajectory difference between two devices, viz a refrigerator and an air compressor in the BLUED dataset. From Figure 2.12 (c) and (d), we can see that there is only a slight difference between the current and voltage waveforms. But when comparing the current against the voltage as shown in Figure 2.12 (e) and (f), the V-I trajectories are quite different. [79] utilize geometrical properties of V-I trajectories to sift between devices.

Transformations of current or voltage waveforms, including the Fourier transform, the wavelet transforms, and the eigenvalue decomposition are also useful. For instance, a short-time Fourier

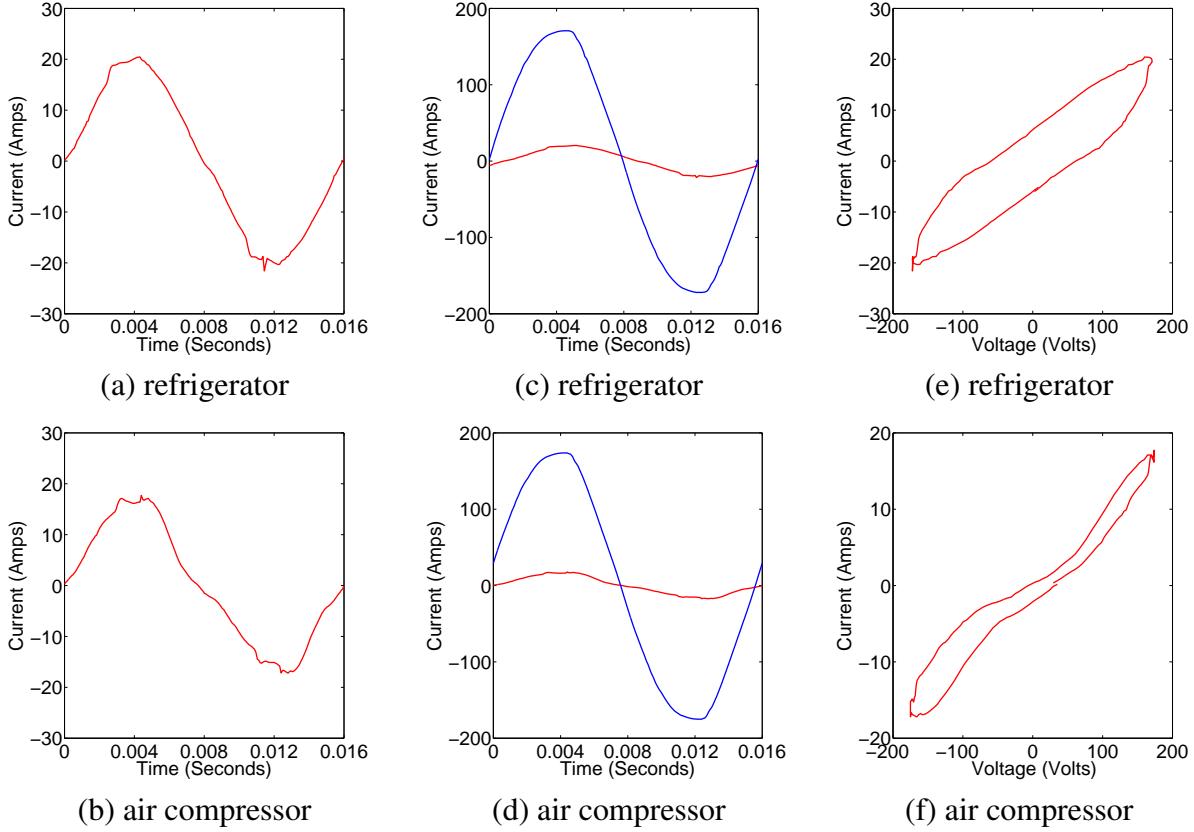


Figure 2.12: Current waveform of (a) a refrigerator and (b) an air compressor. The current and voltage of (c) a refrigerator and (c) an air compressor. The V-I trajectories of (e) a refrigerator and (f) an air compressor.

transform (STFT) approach has been used in [129] and [25] employs wavelet transforms of current or voltage waveforms to identify devices. The eigenvalue of current or voltage waveforms is analyzed as a feature in [92]. 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 constitute another class of features defined over current or voltage waveforms. Harmonics are integer multiples 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 reflecting the inherent characteristics of devices. They play an important role in helping 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 the fact that, while an air compressor and refrigerator have similar

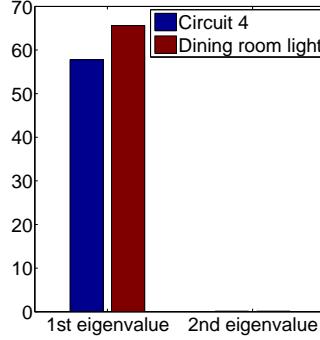


Figure 2.13: The eigenvalue of a circuit and dining room light.

real power, by analyzing the first three harmonics they can be distinguished. Here, the magnitude of the first harmonic of air compressor is larger than that of the refrigerator. Also, the magnitude of the second harmonic of air compressor is smaller than the magnitude of the second harmonic of the refrigerator.

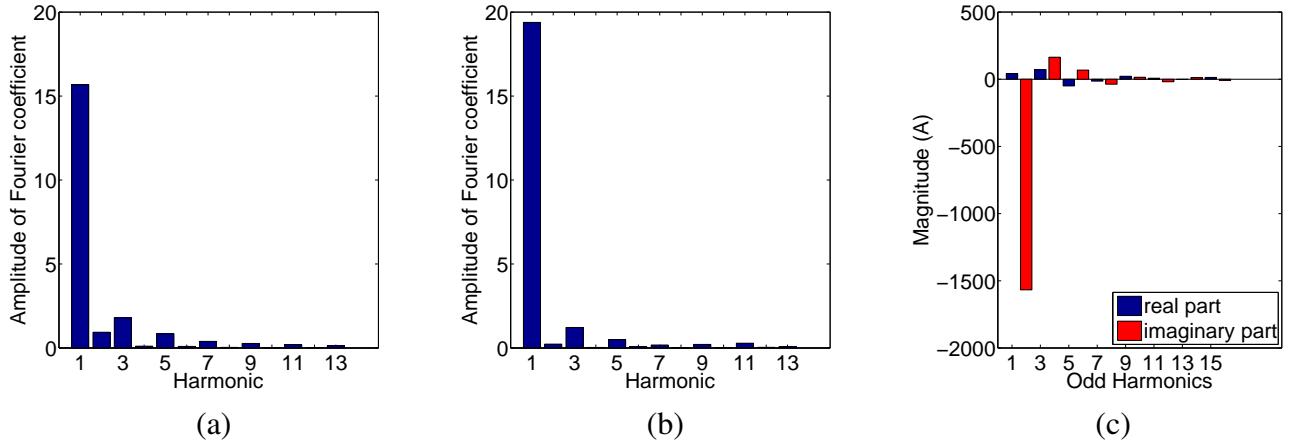


Figure 2.14: Harmonics Feature of (a) a refrigerator and (b) an air compressor (c) real and imaginary part of odd number of harmonics of a refrigerator.

Harmonics have been employed in prior work [5, 80, 86, 101, 127, 140]. Generally only the odd harmonics are utilized. To the best of our knowledge, the highest employed harmonics are the 15 odd harmonics [127]. VSDs are hard to distinguish but harmonics can be used to separate them. [85, 86] 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 k th apparent harmonic power is calculated as $A_k = \sqrt{P_k^2 + Q_k^2}$, where k is the order number of the harmonics. The correlation pattern between the real power and the k th apparent harmonic power is detected as a characteristic of each VSD.

The real and imaginary parts of harmonics are also separately usable as features. Again, 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 n th current harmonic and θ_n is the phase angle of the n th current harmonic. Figure 2.14 (c) shows the real part and imaginary part of the odd harmonics of a refrigerator. Also, [127] gives an example of how to separate devices by the real and imaginary part of harmonics.

A variant of harmonics is the spectral envelope. This is a short-time average of harmonics and was proposed as a device feature in [80, 89]. Further, harmonics can be used in conjunction with other features to disaggregate devices. [140] introduces a switching-function to identify variable speed devices. Figure 2.15 gives a comparison example of current before and after rectifier and inverter of

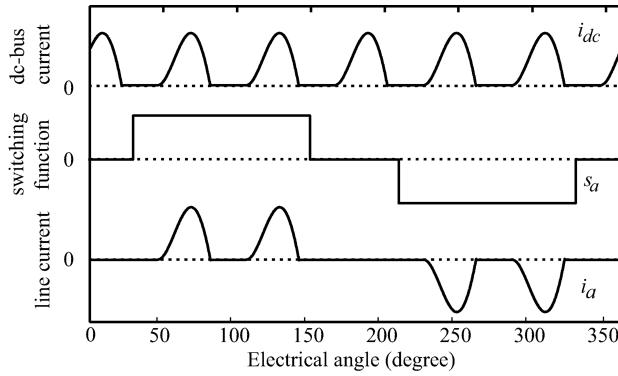


Figure 2.15: Switching-function for VSDs disaggregation (Courtesy:[140]).

VSDs. The operations of the rectifier of a VSD correspond to a switching function. The current is initially direct current I_{dc} and after the current goes through the switching process, 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 n th 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: [112] and [53] 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. [112] 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 has also been considered as a feature [53]. Figure 2.16 (a) depicts the frequency generated by an LCD TV's on and off events from the Kaggle dataset [22]. 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.

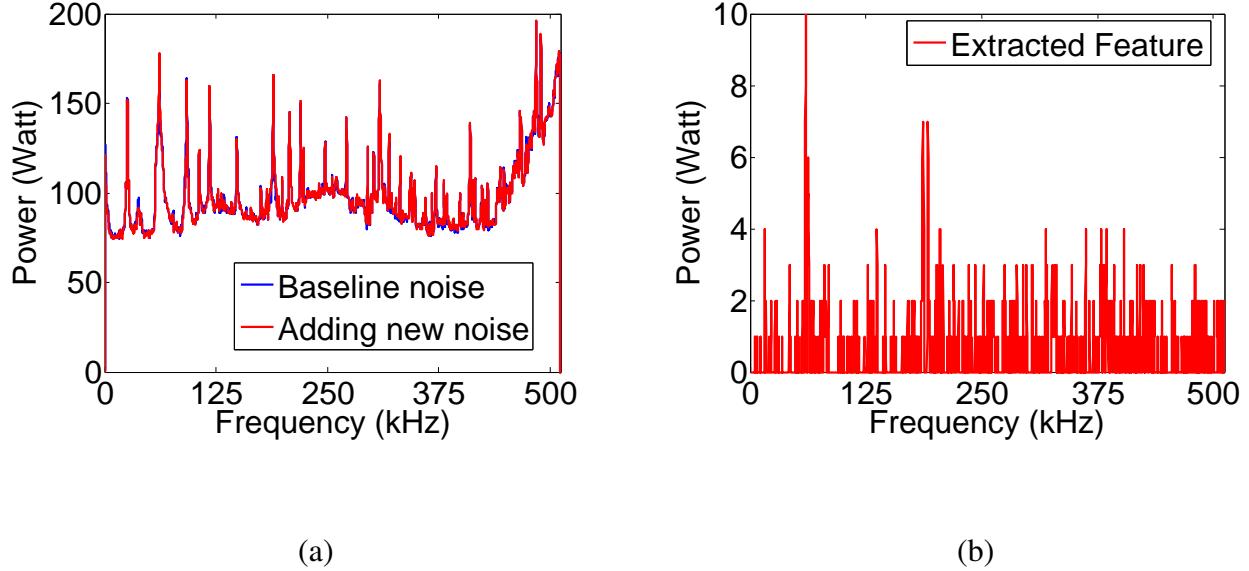


Figure 2.16: (a) Baseline noise with newly added noise. (b) Noise Feature of a device.

2.4.4 Features beyond Current and Voltage

Duration or time of use The operation of a device may conform to a routine schedule of when it is turned on or off, and how long it is on or off. The time of device usage involves the month of the year, 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 the total power usage of a commercial building [123]. Power usage is high during a weekday. In contrast, the power usage on weekends is quite low.

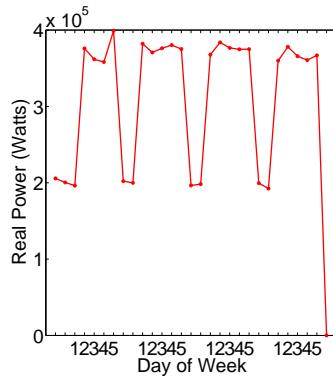


Figure 2.17: Day of Week feature.

Date, time, and week features have been used in previous work. For example, [67] points out that a laptop is often powered-on in the morning of workdays and that the TV is powered on during the evenings and weekends. Also, [67] analyze the on/off distribution of devices and find that they conform to a 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. Such correlation between devices has been analyzed and integrated with a HMM-based model [67]. This study 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. [31] combines the features of on/off time, duration, and the correlations among devices to mine for usage patterns. This approach is designed especially for correlations between different types of devices other than the devices of the same type. In continuing work, [30] develop an algorithm (CoMiner) to discover correlation between devices.

EMF, sound, and light. Electromagnetic fields (EMFs) are generated by electrical devices when they are powered on. Different devices can result in different EMFs which can be monitored by an EMF sensor around devices. This idea is used in [50] to detect on/off state changes of devices. In addition, sound and light resulting from devices can help judge the status of devices as described in [68]. This work monitors sound by an acoustic sensor at 4Hz sampling rate to identify the compressor status of a refrigerator. Also, the light intensity from a sensor 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 variation in the usage of the HVAC system in the summer and winter. The correlation between temperature and electricity usage is studied in [110]. This work exploits the fact that when the temperature is higher, the daily electricity usage increases.

Other possible features. Additional sensors, such as motion sensors [35], can determine activity inside a building, and provide useful input to a disaggregation model. [128] integrates data from electricity and other sensors at home to disaggregate low-power devices such as lightbulbs. [141] proposes using plug-in low-cost outlet sensors to help capture appliance state. Also, [136] uses limited plug-level sensors to improve 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 is necessary, which comprises its own set of algorithms. This phase extracts features or transforms data from one domain to another, for example from the time domain to the frequency domain. In this section, we first introduce pre-processing algorithms, then give an overview of all disaggregation algorithms (supervised, unsupervised, semi-supervised), and finally present the algorithms' advantages and disadvantages.

2.5.1 Pre-processing Stages

Event types

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; *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. 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. 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 is common practice for current and voltage to be transformed into harmonics and wavelets. These transformations are best captured as vector events.

Feature extraction algorithms

Point events are easily obtained from the raw dataset, which may provide real power or reactive power as a time series. A commonly used event definition is to rely on 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. After pre-processing a time series from the high-frequency data, we can obtain several features, like the startup value of the current, the current waveform, the harmonics of the current, the current transformation, the eigenvalue of the current, the voltage waveform, or voltage noises. Generally these pre-processing algorithms are classified into three types: 1) *Basic signal processing*, 2) *Fourier transforms*, and 3) *Wavelet transforms*. Basic signal processing is used to filter, shift or amplify time series data. A Fourier transform converts the time-series data from the time domain to the frequency domain, and harmonics are acquired from the result of the Fourier transform. Wavelet transforms organize a time series into different scale components, and each component is assigned to a frequency range. Features extracted in this manner 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. However, 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 gives examples of several vector events using these three types of pre-processing algorithms. Basic signal processing steps, a low-pass filter, amplification and shifting are used to extract the distorted voltage shapes in [37]. In [140], the Fourier transform of the waveform is

Table 2.1: Pre-processing algorithms with 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[37]						✓	
Fourier Transform [140]		✓	✓				
Wavelet Transforms[25]			✓		✓		

applied to find distinct coefficients for those VSDs. The Fourier transform is also used in [62] to extract harmonic features. The wavelet transform is employed to identify harmonics features of devices in [25].

2.5.2 Overview of Disaggregation Algorithms

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

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

I summarize the merits and shortcomings of supervised and unsupervised learning algorithms for disaggregation in terms of the installation cost of meters, dataset size requirements for building models, the computational cost of operation process, and the accuracy results of disaggregation. Compared to unsupervised learning approaches, the advantages of supervised learning algorithms are that the disaggregation accuracy is typically higher (for the same dataset), they require smaller data sizes, and have typically faster operation (in practice). Compared to unsupervised learning algorithms, the main disadvantages of supervised learning techniques lie in that the labelled data of each device is difficult to obtain due to the cost of meter installation.

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 matching, rule-based algorithms, SVMs, kernel-based subspace classification, Bayesian classifiers, neural networks, genetic algorithms, dynamic Bayesian networks, sparse coding, AdaBoost, decision trees, and combinations of techniques like SVM and AdaBoost.

Neural networks 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.

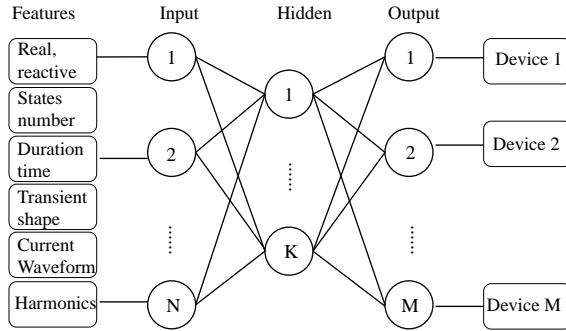


Figure 2.18: Neural Network Approach for Energy Disaggregation.

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 classification for classification based on real power and reactive power by transforming the aggregated data into images for processing [118]. Next, [12] employs a backpropagation (BP) neural network with the attributes of number of states, duration time, and average energy consumption. Furthermore, the training stage of [43] and [127] are based on 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 [127] compares several neural network approaches, namely multilayer perception (MLP), radial-basis

function (RBF) network, and support vector machine (SVM) with linear, polynomial and RBF kernels. The results suggest that MLP and RBF-based approaches have high classification accuracy.

Chang et al. extend the back propagation approach by employing an electromagnetic transient program (EMTP) with transient real power when devices start up [28]. The transient shape is a vector event rather than a point event. 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 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 the highest recognition accuracy, is retained.

Recently a basic adaptive neural network (ANN) applied to energy disaggregation was presented in [92]. Based on a combination of features, such as real power, reactive power, transient shapes, harmonics, the eigenvalue of the current waveform, the 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 approach have been proposed. Reference [143] classifies the transient events by way of back propagation, and [27] adopts learning vector quantization (LVQ) to recognize devices. Neural networks have also been combined with other approaches. For example, [26] combines a multi-layer feed-forward neural network with a genetic algorithm to analyze the device turn-on transient signatures.

Support vector machines SVMs attack the problem using multiclass learning techniques by learning event features only from the training data, in contrast with unsupervised methods that learn features from the entire dataset. Kernels such as the radial basis function (RBF) kernel are adopted to learn complex features like harmonics. At first, SVMs were employed to classify devices by 13 odd-order harmonic current and phase angles from on and off events [109]. Later, a kernel-based subspace classification (KSC) approach is used for events classification in SVM [105].

SVMs were widely utilized with noisy datasets although they do not scale well for large data sets. An SVM was adopted by [112] to classify transient pulses noise from various homes. In [49], SVMs are applied to transient and continuous voltage noise data. The noisy voltage generally produced by devices influences the power wiring. According to Gen Marubayashi [97], 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 is 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.

In addition to their use as a standalone classifier, SVMs are also utilized in energy disaggregation in combination with other approaches. In [106], both stand-alone SVMs and a combination of SVM and radial basis function network (RBFN) are implemented to compare the disaggregated data with the ground truth harmonics.

Bayesian networks A combination of SVM and dynamic Bayesian network was demonstrated in [49]. To start, a threshold value is predefined for the Euclidean distance between the new data and basic noise data. Next, a window appears 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. Finally a dynamic Bayesian network is utilized to classify the devices based on prior information, such as washing machines, dryers, and HVAC.

Rule-based algorithms 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 a maximal length of four were used for real power with transition states in [77] to classify devices. The principle of closure rules is that if only one device changes its state, the baseline signature is the same as before the occurrence of the state change. The rules become complex for each device if the vector events feature is introduced.

Rule mining is also proposed in [117]. 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, the time when devices are probably turned on for each hour of a day and each day of a week is ascertained. In the second step, those significant rules are chosen by a *JMeasure* (e.g., only those rules with values greater than 0.01 are selected).

Naive Bayes classifier Algorithms that use the Naive Bayes classifier (NBC) are proposed in [145] to distinguish devices. Based on that approach, [148] uses power and time as features to automatically disaggregate the major residential electronic devices.

Other strategies Reference [18] evaluates four approaches: k-nearest neighbor (kNN), Gaussian naive Bayes (GNB), decision trees (DT) and multi-class AdaBoost (MultiBoost) for high-frequency data. Reference [109] integrates SVM with AdaBoost to classify devices based on odd-number harmonics. If we suppose that in a support vector machine, the margin Q is defined as

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

where

$$\rho(z_i, f) = y_i f(x_i) \quad (2.11)$$

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

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

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 $\mathcal{G}(\alpha)$, where $t = 1, \dots, T$.

Computational Complexity The computational complexity is a function of the classification approach used. For training, decision trees tend to be faster than techniques which require quadratic optimization such as SVMs. Complexity is also a function of the number and type of features. 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 set becomes multi-dimensional. Neural networks have the advantage of naturally detecting interactions between the disaggregated features and output time series data but training can be fraught with local minima problems.

Nearest Neighbor-Based

Several energy disaggregation algorithms have been designed using nearest neighbor (NN) techniques. 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 disaggregation of single feature, viz. point event or vector event, Euclidean distance is a common choice [53]. For disaggregation of multiple features, 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 [124].

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 multiple features such as transient power shape [18, 19, 86, 124].

Reference [124] describes how transient shapes of power consumed by devices over time are discovered. Transient shapes that are exemplary of each device are summarized and recorded in the 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 is taken into account, consequently the data points in each window are 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, [125] decomposes the real power transient shape into two vectors; a shape vector and a time vector, rather than setting the whole transient shape as a device feature. Figure 2.19 depicts an exemplar with two shape vectors, s_1 and s_2 .

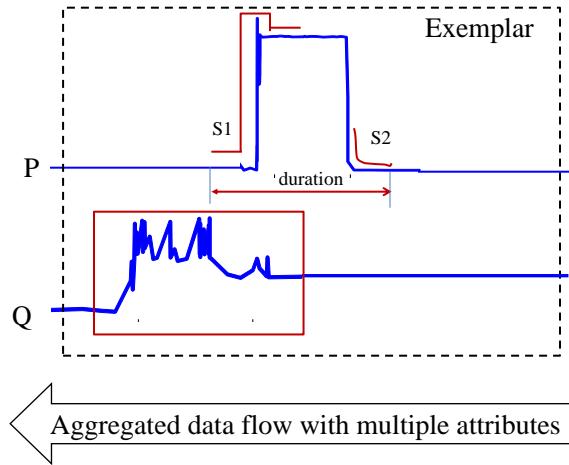


Figure 2.19: Transient Shape Decomposition and KNN Search

To identify which device the disaggregate signature belongs to, it is necessary to compare the signature with an exemplar by the least square criteria. After that, a similar exemplar comparison approach is applied to identify the devices. The advantage of real power shape decomposition is that when comparing the transient shapes, only some characteristic parts are needed rather than the entire transient shape in the data. This helps cut computational cost for the exemplar comparison phase. Although this paper doesn't mention the KNN algorithm explicitly, the description in this paper exactly matches the methods used in KNN algorithms to search for closest shapes.

A variant of the KNN approach measures the Euclidean distance with inverse weighting [53]. 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 the 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 [86], 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, [86] is classified into the KNN category.

Using relative density Techniques that estimate the 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 [75]. A real power probability density function is used as a feature to classify two-state devices in [147]. The number of the device is indexed by the power, as shown in Figure 2.20.

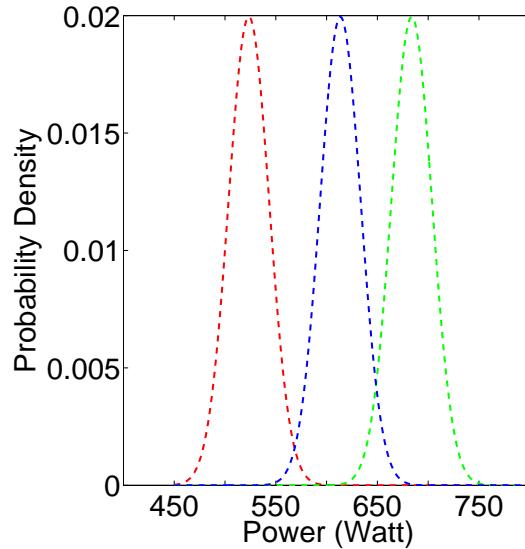


Figure 2.20: Probability density functions of three appliances neighboring by power draw.

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 m^{th} cluster represents device m . Next, 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 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 of the aggregated data of all device features at each window to obtain the nearest instance.
2. When multiple features are applied, defining the measure of distance 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 discussed above, 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.

Reference [62, 63] 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 $y_t, t = 1, 2, \dots, T$ are drawn from a distribution $G(y)$ and the supposed distribution $F(y)$. The binary hypothesis testing problem is defined as

$$H_1 : G(y) \neq F(y) \quad (2.13)$$

$$H_0 : G(y) = F(y) \quad (2.14)$$

After this 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.

A generalized likelihood ratio is applied in [10], [20] and [94]. These papers 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.

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

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 $y_t (t = j \dots k)$ about the mean value u . The greater the probability, the greater the chance the data points belong to a specified device.

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

A hierarchical probabilistic model is proposed in [139]. 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) probability 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 [67] 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:

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, and 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

Reference [13] utilizes 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 $\Gamma_l = S_{l1}, \dots, S_{lk}$ and a device's finite state machine path $\Gamma_m = S_{m1}, \dots, S_{mk}$, the Shannon entropy is calculated as Equation (2.16).

$$Q_l = - \sum_i \Delta e_i \log |\Delta e_i| \quad (2.16)$$

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 [26], genetic programming is integrated with the neural network to identify devices. In [137] clustering is integrated with finite state machine and dynamic programming to disaggregate the devices in real time with low cost.

Dynamic model [41] 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).

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

where $m \in 1, \dots, M$, M is the number of devices, x_m is the input to the device m , and h_m is

a function which denotes the underlying dynamics. To estimate $x[\cdot]$, blind system identification techniques [3] are used.

Integer programming Integer programming [131] 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 kinds of devices, and there are C_n appliances for each kind of device.

Let us suppose there is an aggregated load *current* y ,

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

where $c_m \in \{0, \dots, C_m\}$ is integer variable for $m \in \{1, \dots, M\}$, $t \in \{1, \dots, T\}$, $x_t^{(m)}$ 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 each device has only one operating mode.

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

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

subject to

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

Viterbi algorithm Another paper [145] 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.

$$\{\hat{S}_t\} = \operatorname{argmax}_{s_t} [\{S_t\} | \{\omega_t\}] \quad (2.20)$$

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

The Viterbi algorithm adopts a similar approach to the one mentioned in [14]. The difference of this method is that they introduce the probability density function of real power of each device.

Sparse coding Reference [73] 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, which aims to model each source using non-negative sparse coding by solving Equation (2.21).

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

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 s th 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 $B_m, m = 1, \dots, M$.

$$\hat{A}_{1:M} = \operatorname{argmin}_{A_{1:M}} \| Y - [B_1 \dots B_M][A_1 \dots A_M]^T \|_F^2 + \lambda \sum_{p=1, q=1, m=1}^{r, s, M} E(A_m)_{pq} \quad (2.22)$$

where M is the number of devices, $\hat{A}_1, \dots, \hat{A}_M$ are the activations related to aggregated power. Each column of $Y \in R^{T \times s}$ represents the s th 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 $A_{1:M}^*$ and $B_{1:M}$ alternatively, the sparse code and the real power consumption of each device is calculated.

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

where α is the step size.

Note that sparse coding is also extended to water disaggregation [40].

Nonnegative tensor factorization [48] 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.

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

where $S^{(l)}$ represents 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 $t = 1, \dots, T$.

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

Furthermore, to solve this problem, [48] implements two solutions: one is based on nonnegative sparse coding, and the other is a multidimensional representation and factorization method. Nonnegative sparse coding is introduced in [73]. For tensor decomposition, this paper adopts the PARAFAC approach [72] with nonnegative constraints.

Computational Complexity The computational cost of dynamic programming for classifying devices is polynomial. Let us assume m is the number of FSMs, n is the number of "diff" data, and the computational time cost of the dynamic programming step to classify devices is $O(mn)$ [33]. However, the computational cost of the whole procedure in [13] is higher because it contains the steps of fuzzy clustering and genetic programming.

Reference [131] 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 [131] is relatively high because it utilizes high-frequency data with large data size.

The computational cost of the Viterbi algorithm is linear i.e. $O(T)$, where T is the number of aggregated power points [21].

The computational cost of sparse coding is high. Therefore the energy disaggregation is formulated as a ℓ^1 minimization optimization problem. The computational cost decreases and becomes linear to the data points and number of devices $O(TM)$ [91], where T represents the aggregated data points and M represents the number of devices.

Advantages and Disadvantages of Optimization-based Techniques The *advantages* of the optimization solution are 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 techniques can be summarized as:

1. If more features are introduced, such as harmonics, it's difficult 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 [55] because the exact information of individual circuits or devices is unknown. More recently, unsupervised disaggregation has emerged as a hotbed for research. Clustering [51] is used to group similar events. Different approaches such as HMM [67, 74, 111] and temporal mining [122] 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 of the number of devices, some unsupervised learning methods also assume that the number of devices is also known.

Hierarchical clustering-based algorithm

Gonccalves et al. proposed a method that disaggregates devices without a priori knowledge of the total number of devices [51]. As the first step, in order to extract the real and reactive power features, blind source separation [88] 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 Euclidean distance ($[P_t, Q_t] - [P_{closest}, Q_{closest}]$).

Computational complexity [51] only studies disaggregating devices with on and off events. In this study, 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 $O(T^2)\log T$ [61].

Advantages and Disadvantages of Clustering-based Unsupervised Learning Techniques

The *advantage* of clustering-based unsupervised techniques can be stated as:

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 the same power levels.
2. These techniques are applied to devices with two states, on and off, but are not applicable to devices with multiple states.

FHMM-based technique

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 that the power usage of the entire house is available. Kim et al. propose an FHMM technique and FHMM [67] to disaggregate devices. 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.

Constraint FHMM extends 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 probabilities is given in Equation (2.25).

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

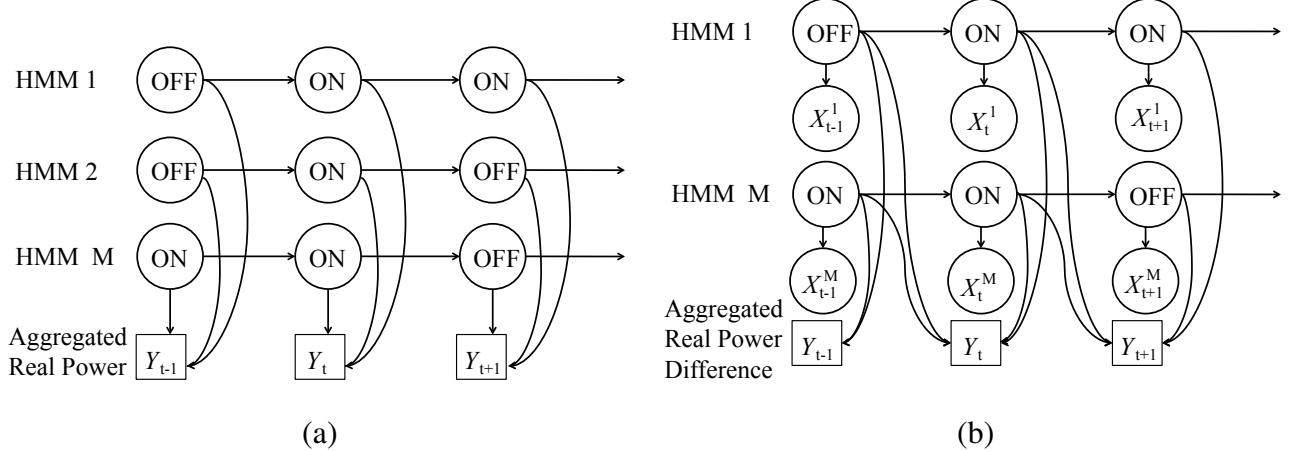


Figure 2.21: Graphical model with M devices. (a) FHMM and (b) Difference FHMM.

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

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

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) [74]. It is a mixture of the additive factorial model and difference FHMM model. The box diagram of AFAMAP is shown in Figure 2.22.

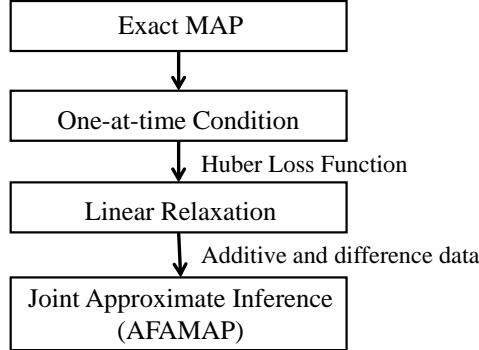


Figure 2.22: AFAMAP Flowchart.

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

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

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

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.

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

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

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

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

Until 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.

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

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 [111]. The difference FHMM is shown as Figure 2.21 (b). This method assumes that we know the labels of each device, 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 as unsupervised disaggregation. During the procedure, the features are repeatedly deleted. More device features are gradually identified. In the next step, the different appliance behaviors obtained from the previous step, like peaks arising from the device being turned on or the power demand of the device, are used as a

prior for the difference FHMM. The EM algorithm then is then used to evaluate the likelihood of whether the profile is of a certain device type.

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

where y_t, \dots, 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 $\hat{\theta}$, and \mathcal{D} is the predefined likelihood threshold. In the final step, all these devices are disaggregated by an extended Viterbi algorithm.

Further, [58] uses HMM for electric heat usage disaggregation. HMM and AFAMAP are also run by additional applications [93].

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 [21]. It's difficult to obtain the direct solution theoretically. Therefore Gibbs sampling is applied to the first FHMM solution [67]. Later in the AFAMAP, QP problem techniques are used in the solution. In another variant of difference FHMM [111], 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 [122] 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. A 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, and overlap exists between any two episodes. For example, the second instance of Episode 1 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

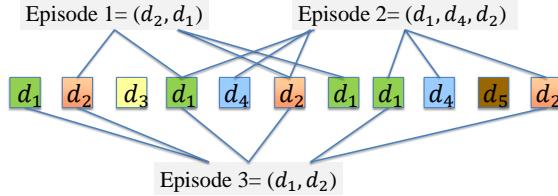


Figure 2.23: Motif Mining Example ([122]).

AFAMAP [74]. 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 can be 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 model

In addition to HMM, another probabilistic graph model was proposed by [65]. The model is composed of three layers. The component layer forms the the bottom-most layer, the second layer

comprises a probabilistic graph model that captures appliances, and finally the top-most layer is an inter-appliance layer. So far, this approach has not been fully implemented.

2.5.5 Semi-supervised Learning Algorithms

Semi-supervised algorithms assume that the features for each device, such as the power levels of the device are 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 algorithm

Lam et al. initially proposed to utilize the voltage-current (V-I) trajectory of an appliance as a feature to perform clustering [79]. Hierarchical clustering is used to cluster the appliances by analyzing these V-I trajectories. When hierarchical clustering is employed, pairwise differences between the features of V-I shapes are calculated. Then a dendrogram is created to show the relationship between devices.

HMM-based algorithm

The method for FHMM proposed in [67] 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, a hierarchical Dirichlet process hidden semi-Markov model (HDP-HSMM) [64] is adopted by extending a Bayesian nonparametric approach to capture the duration distribution of each device.

Optimization-based algorithm

Reference [142] proposes a contextual supervision approach to solving the single-channel source separation problem as an optimization problem. It uses the power levels and the times of turn-on and -off for each device as features.

$$\begin{aligned} \min_{x_1, \dots, x_M, \theta_1, \dots, \theta_M} & \sum_{m=1}^M \{\ell_m(x_m, Z_m \theta_m) + g_m(x_m)\} \\ \text{s.t. } & \sum_{m=1}^M x_m = y \end{aligned} \tag{2.31}$$

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

Advantages and Disadvantages of Semi-unsupervised Techniques

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

1. It either learns the 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:

1. The existing features of each device are difficult to obtain.
2. While the non-parametric approach works, the computational cost is still high.

2.6 Evaluation Methodology

To the best of our knowledge, there is no universally accepted evaluation measure(s) to evaluate the results of energy disaggregation algorithms. In the situation where we know the true power consumption value of each device, there are two broad classes of measurements to evaluate device disaggregation results: event-based and time series-based. In an event-based measure, we check whether the inferred on/off events are correctly determined for the target device. The time series measure gauges whether the disaggregated power values of each device are in the range of ground truth over a period of time. For both event-based and time series-based measures, the disaggregation results can be measured through a confusion matrix, F-measure, or a simple error rate.

2.6.1 Evaluation based on Events

Event-based evaluation measures, primarily on and off events, have been widely used in previous research work. They are identified by real power and reactive power as described in [18] and [26], by real power and transient shapes [49], by real power, reactive power and voltage-current trajectory [143], by just transient shapes [27], by comparing the waveforms as shown in [131] and [19], or by analyzing the voltage noise [112].

Generally, the event classification rate is calculated using the following procedure. For each device, we suppose there are a total of N on or off events $\{E_1, \dots, E_i, \dots, E_N\}$ during a period of time; assuming the corresponding predicted events are $\{\hat{E}_1, \dots, \hat{E}_i, \dots, \hat{E}_N\}$, then, according to [106], the coverage range of these on or off events is given in Equation (2.32).

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

Higher coverage for a device thus implies better prediction results.

After this calculation, the disaggregation accuracy rate can be calculated by judging whether the disaggregated devices are classified as the correct device. Reference [51] evaluates performance based on this criteria, as given in Equation (2.33).

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

where Ω is the set of all ground truth device labels, and C is the disaggregated device labels set. Let us suppose there are M number of clusters corresponding to M devices, N_m is the number of elements in cluster m , and ω_k is the subset with the highest frequency in each $c_j = m$ cluster. In addition to the classification accuracy rate discussed above, the F-measure is also employed in [145] to define the disaggregation performance.

2.6.2 Evaluation based on Time Series

The time series approach compares the disaggregated power values with the ground truth power values at each point over a period of time. For instance, [74] compare the disaggregated time series with the ground truth of each device as:

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

where y_t is the true real power value at time t , and \hat{y} is the disaggregated real power value. Note that the disaggregate error rate can be calculated over a specific time range [74]. In addition, [111] uses the square root of the error rate of all devices over a period of time to calculate the disaggregation accuracy, as shown in Equation (2.35).

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

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 the F-measure [67].

The fourth method is to evaluate by accumulating (and comparing) the total energy over a period of time, as in [122]. Once again, the 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. Reference [92] proposes three evaluation metrics; detection accuracy, disaggregation accuracy, and overall accuracy. The first one is the events classification accuracy. The last two metrics are similar to the standard F-measure metric that is commonly used in machine learning algorithms.

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 information and non-AC power information. To obtain AC power data, a real power meter, reactive power meter, ammeter, and voltage meter (usually consolidated into one meter) can be installed to record different power values, current or voltage values, and noise generated by power line. Sensors are installed to collect non-AC power data, like electromagnetic fields (EMF) around devices [50], light, and sound [68].

Figure 2.24 illustrates how four types of meters/sensors are installed in a building. After two-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. A power meter is installed on each circuit, such as for the refrigerator, to monitor the true status of the devices (to validate results). A sensor is plugged into the outlet 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 the refrigerator, to capture its electrical magnetic field, sound-related, and light-related operations.

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

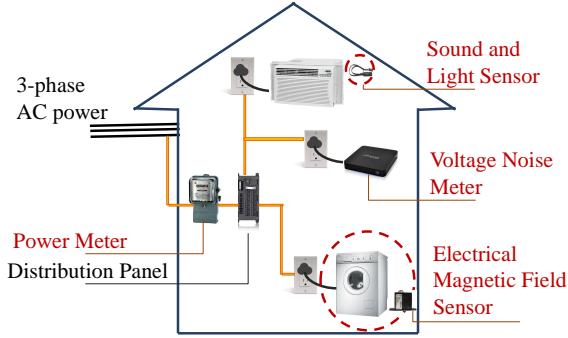


Figure 2.24: Four Types of Meters in a Building.

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, normally current and voltage are monitored separately to facilitate the capture of different device characteristics. To capture high-order harmonics, the sampling frequency used to record the data should be at least twice as much as the highest frequency. When targeting to capture harmonics with the highest order $N_{highest}$, the desired harmonics frequency is $N_{highest} \times 60\text{Hz}$, then the sampling frequency of recorded data f should meet the criteria $f \geq N_{highest} \times 1/60\text{Hz}$.

There are many meters that can record the low-frequency power. However, high-frequency data must be monitored by special devices such as a TED [134]. Some examples of aggregated energy data collection include [19], where a voltage/current meter is installed in a residential home in Pittsburgh, PA. This experiment chooses 17 on-off devices and installs plug-level meters for these 17 devices. The data is recorded at a high frequency (100 kHz). Then features such as real and reactive power and harmonics are extracted at the frequency of 20 Hz. Voltage noise data [112] is obtained at a high-frequency sampling rate by plugging the meters into an outlet. Reference [12] installs an optical sensor to collect the real power. Then the on-off events are extracted from the real power. A detailed comparison of a whole-house meter, circuit meters and plug-meters is given in [19].

Public datasets

Although there are many data sets available for energy disaggregation, especially in the power industry, the majority of them are not open to the public. So far there are a few public data sets, including REDD [75], BLUED [11], Smart [15], AMPds [96], CASAS [36], iAWE [16], and GREEND [104]. 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 this data is stored as plain text. Some work proposes to store these datasets in

a database [78] or builds a metadata as a standard [66].

2.7 Ongoing Research

Current analytics research in energy disaggregation is primarily focused on two areas: feature discovery and new learning algorithms. Feature discovery is primarily driven by specialists in electrical engineering who bring a nuanced understanding of device and electricity features to bear upon this problem. With greater emphasis in the data mining community on deep learning and other methods that emphasize novel representational layers, there is considerable scope for data scientists to play a role here, as research is still in its nascent stages. In the area of learning algorithms, unsupervised methods have a distinct advantage, since labelled data is not required and today's state-of-the-art unsupervised methods are quite competitive when compared with supervised methods. Research into novel machine learning models will continue to improve disaggregation performance.

As more companies participate in this area, new software tools are concomitantly being developed to help users analyze power consumption data. Smart![15] provides an interface for users to monitor their power consumption. A database has been built from the REDD dataset [78]. These tools benefit both developers of new methods and the end consumers.

Non-intrusive load monitoring paves the way for many other research problems, which are not surveyed here. One such area is occupancy research, which infers whether there are people occupying a building within particular segments of space or time [29]. The second area is demand response [6], which refers to ways by which utilities balance supply and demand by offering incentives to consumers to reduce or shift their peak energy demand. Inferring activities of daily life [126], promoting personal energy savings [87], and efficient energy management [34] are all current topics of research.

2.8 Conclusion

Significant increases in energy usage worldwide and the consequent impact on the environment have pushed energy disaggregation research to the forefront in recent years. We have surveyed features, algorithms, evaluation measures, and instrumentation required for energy disaggregation from a data miner's perspective. While initial disaggregation algorithms were focused on low-frequency data, today's installations support high-frequency data recording. Therefore, rich features such as harmonics, transient shapes, noise characteristics, and electromagnetic field measurements are available to improve disaggregation performance. While supervised algorithms were first used in energy disaggregation, it is becoming more common to use unsupervised algorithms. We have also identified the need for the research community to coalesce around accepted standards of evaluation.

In this survey, discussions of energy disaggregation primarily refer to electricity disaggregation, but similar algorithms are being explored for natural gas and water disaggregation. An interesting area of potential research is to jointly disaggregate multiple utilities, e.g., electricity and water, which will undoubtedly provide additional contextual features for analysis. As smart homes and Internet of Things (IoT) installations become more pervasive, it is clear that disaggregation research will continue to be relevant into the foreseeable future.

Table 2.2: Categories of energy disaggregation algorithms.

Category	Sub-category	Algorithm Name	Example	Features Adopted
supervised	Classification	Pair-wise match	[55]	real reactive power
		Neural Network	[118]	real power, reactive power
		SVM	[112]	startup of I_{AC} and voltage noise
		SVM, AdaBoost, RBF, NN	[109]	startup of I_{AC}
		SVM, KSC	[105]	startup of I_{AC}
		SVM, RBFN	[106]	real power, harmonics of I_{AC}
		Bayesian Classifier	[19]	real power
		Genetic algorithm	[14]	startup of I_{AC} , on duration
		Rule-based	[115]	real power
		Dynamic Bayesian network	[49]	real power
	Nearest Neighbor	Decision Tree	[18]	startup of I_{AC}
		AdaBoost	[18]	startup of I_{AC}
	Statistical Model	KNN	[124]	startup of I_{AC}
		Duration PDF	[147]	real power, on duration
	Optimization	General likelihood ratio	[10]	real reactive power, duration
		Dynamic Programming	[13]	I_{AC}
		Dynamic Model	[41]	real power
		Integer Programming	[131]	I_{AC}
		Sparse Coding	[73]	real power
unsupervised	Clustering	Hierarchical Clustering	[51]	real power, reactive power
	HMM-based	FHMM	[67]	real power, time, duration
		AFAMAP	[74]	real power, startup of I_{AC}
		Difference FHMM	[111]	real power
	Temporal mining	Motif Mining	[123]	real power
Semi-supervised	Clustering	Hierarchical Clustering	[79]	startup of I_{AC}, I_{AC}, V_{AC}
	HMM	HDP-HSMM	[64]	real power
	Optimization	Contextually supervised	[142]	real power

Table 2.3: Meters Used in Experiments.

Meter Types	Meter Name	Meter Example	Recorded Features
AC power	ammeter	TED, LEM LA55-P [134]	AC waveform, harmonics
	voltage meter	Pico TA041 [133]	voltage waveform, voltage
	real power meter	National Instruments USB-9215A [60]	real power
	reactive power meter	TrendPoints EnerSure [135]	reactive power
	voltage noise meter	Build by author	voltage noise
Non-AC power	electromagnetic field meter	Trifield [7]	electromagnetic field
	sound sensor	mindstorms [45]	sound strongness
	light sensor	extech [59]	light strongness
	temperature meter	amprobe [9]	temperature

Chapter 3

Energy Disaggregation

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.

3.1 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 [67] and sparse coding [74] 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 [32, 144] for background) to energy disag-

gregation. 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 [75] as well as a commercial dataset, demonstrating the ability of our approach to disaggregate different classes of electrical loads.

3.2 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 [55], 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.

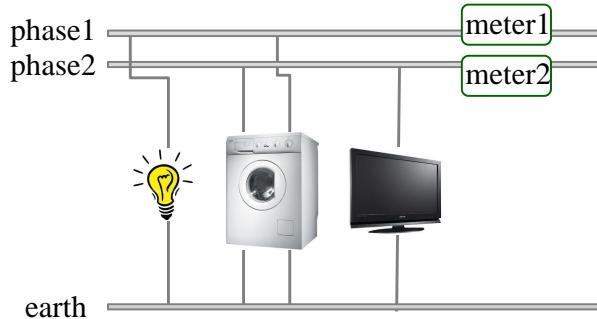


Figure 3.1: A residential setup for data collection.

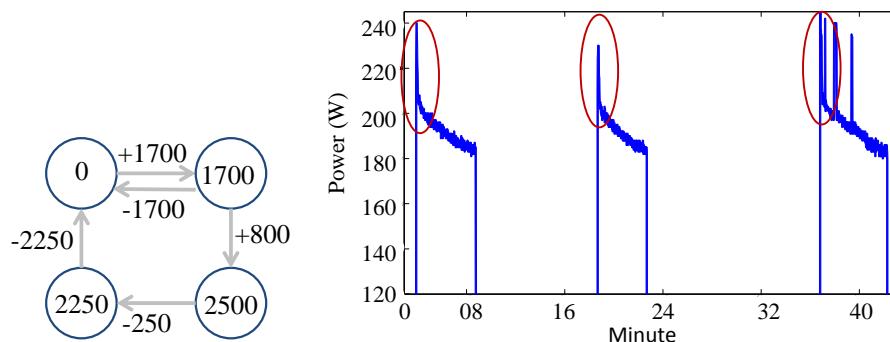


Figure 3.2: Steady state transitions and transient features at startup.

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 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 [108] 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.

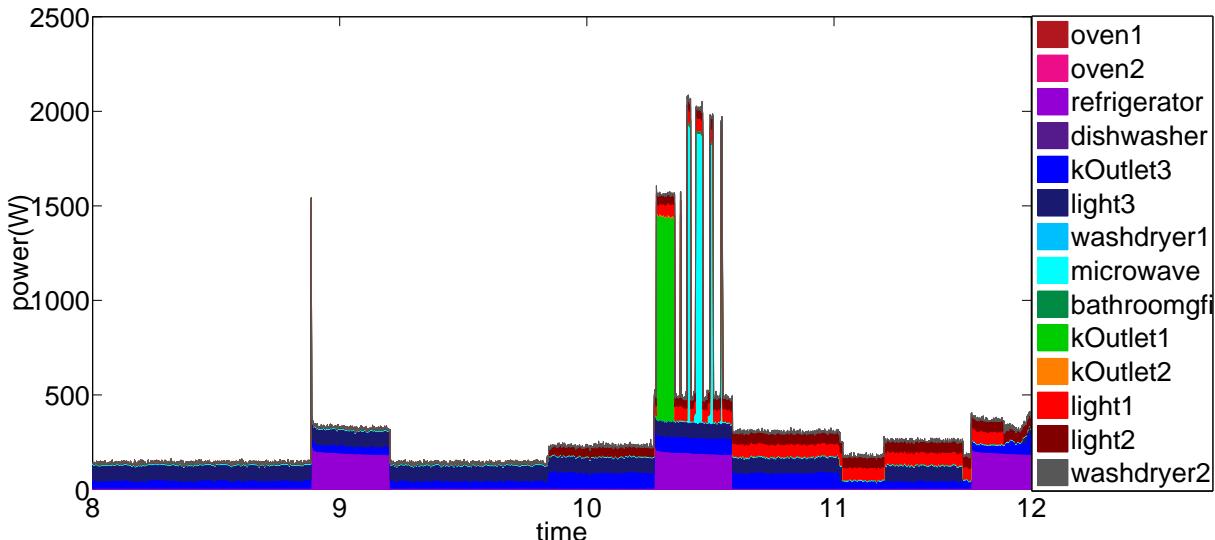


Figure 3.3: Example of energy disaggregation.

Prior approaches to disaggregation.

Initial research focused on using simple device features such as real power and reactive power [55]. With the development of automated meters, transient states generated when devices turn on have been employed to identify devices [124]. Raw current waveforms [127], and voltage waveforms [79], and transforms of the current waveform [25] have also been adopted as characteristics. In particular, harmonics of non-linear devices have been utilized in prior work [25]. Further, non-AC power features such as power line noises [112], time of day and device correlations [67], can be combined with AC power features to aid disaggregation. The underlying algorithms have been drawn from a variety of domains: supervised learning [106], data mining, optimization, and signal processing, e.g., kNN [124], SVM [112], sparse coding [73]. Recent research has placed a great emphasis on building in unsupervised learning features, including hierarchical clustering [79], semi-supervised approaches [111], factorial HMMs [67], and AFAMAP [74].

3.3 Temporal Motif Mining

Early approaches to disaggregation (e.g., Hart[55]) 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).

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).

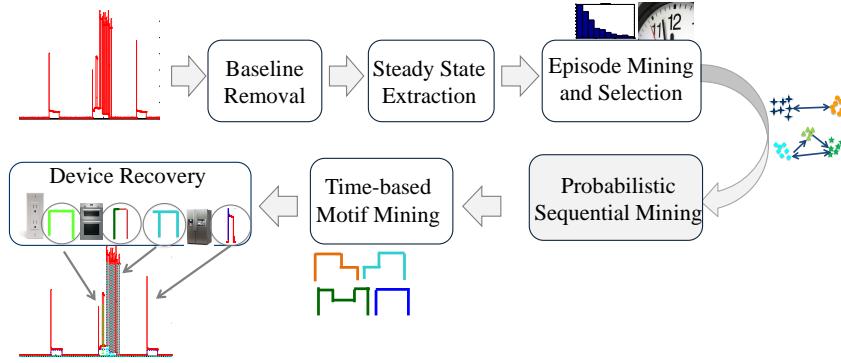


Figure 3.4: Temporal motif mining framework for disaggregation.

Steady state extraction.

Two basic approaches here involve a heuristic method (window-sized filtering) and the more systematic Dirichlet process Gaussian mixture models (DPGMMs) [52]. 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, $\mathbf{y} = \sum_{i=1}^k \alpha_i N(\mu_i, \Sigma_i)$ where $\sum_i \alpha_i = 1$, 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.

Episode mining and selection.

The goal of episode mining [113] 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.

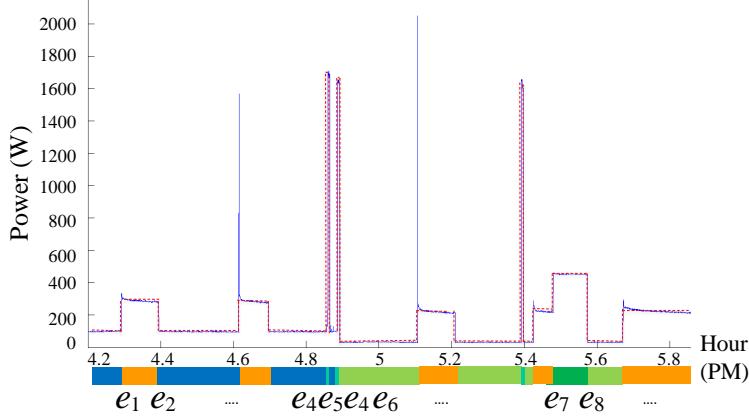


Figure 3.5: Mining episodes from a symbolic time series.

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: $(e_1, e_2, e_1, e_2, e_4, e_5, e_4, e_6, e_1, e_2, e_4, e_5, e_4, e_7, e_8, e_1)$. An episode of length N , $E = (e_1 \rightarrow e_2 \rightarrow \dots \rightarrow e_N)$, 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.

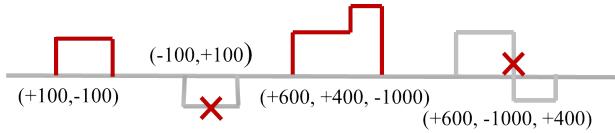


Figure 3.6: Episode constraints.

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 [4], 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 $S = \{e_1, e_2, \dots, e_n\}$ 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.

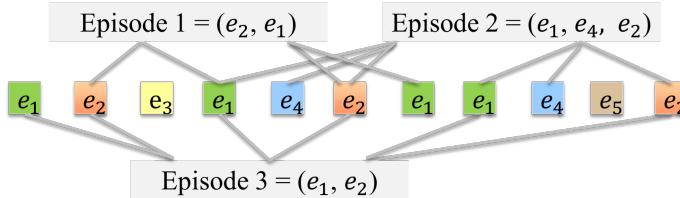


Figure 3.7: Illustration of motif mining. Note that there are 3 non-overlapped occurrences of Episode 3.

Motif mining.

Motif mining aims to find repetitive episodes in a time series using the technique of non-overlapped occurrences [83]. Assume there are five power event change symbols $\{e_1, \dots, e_5\}$ and a time series $(e_1, e_2, e_3, e_1, e_4, e_2, e_1, e_4, e_5, e_2)$ 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 instances 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 motif 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.

3.4 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 $\rho = 0.2$.

3.5 Experiments on REDD dataset

We conduct experiments on the low frequency data from the REDD [75] 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.

3.5.1 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 3.8 (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.

3.5.2 Comparison of Motif Mining and AFAMAP

Next, we conduct experiments comparing our approach with the AFAMAP algorithm [74], 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

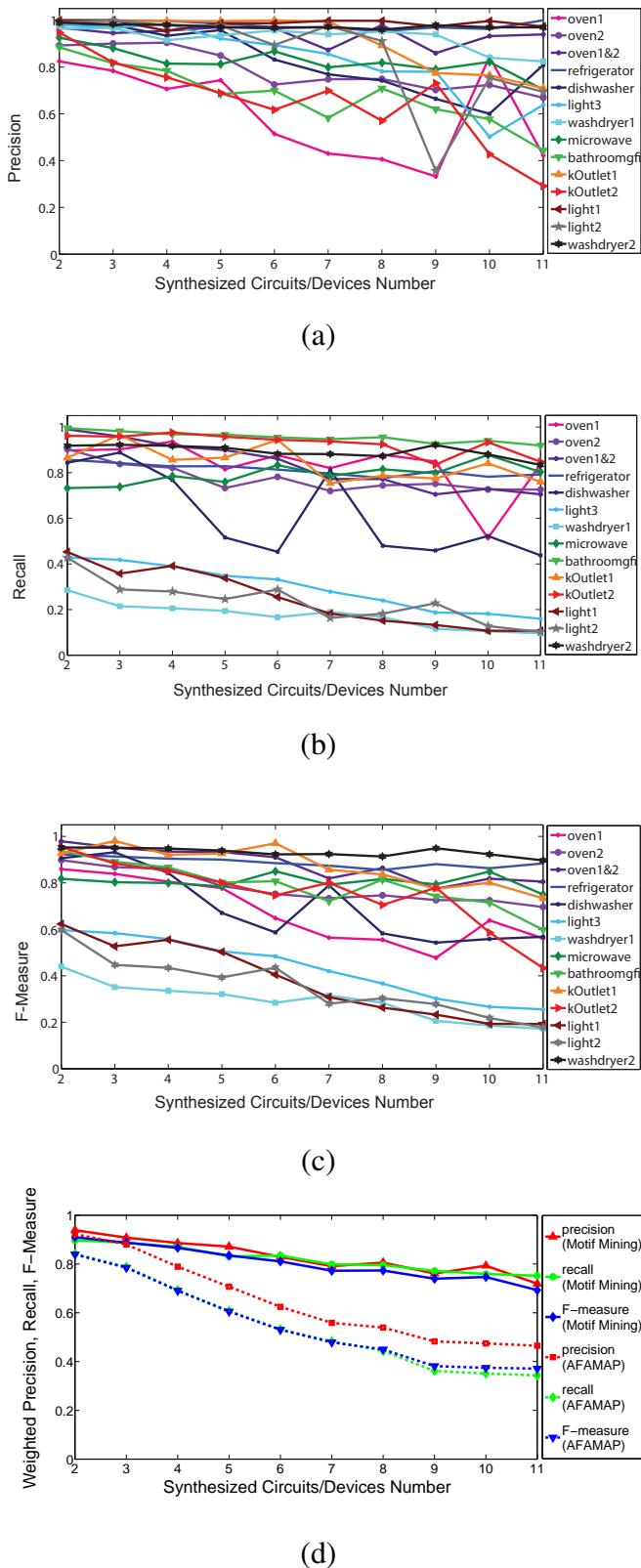


Figure 3.8: We increase the number of synthesized circuits from 2 to 11 and calculate performance measures for disaggregation of each device. (a) Precision (b) Recall (c) F-measure (d) The precision, recall and F-measure of all the devices are combined weighed by their average power levels.

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

device	True Power (W)	Motif mining			AFAMAP (True PL)			Motif mining & AFAMAP		
		Precision	Recall	F-Measure	Precision	Recall	F-Measure	Precision	Recall	F-measure
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

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

3.6 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

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

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.

3.7 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

Recursive Multivariate Piecewise Motif Mining to Disaggregation

4.1 Introduction

With the advent of modern sensor technologies, significant opportunities have emerged to help conserve energy in residential and commercial buildings. Moreover, the rapid *urbanization* we are witnessing requires optimized energy distribution. Energy disaggregation attempts to separate the energy usage of each circuit or each electric device in a building using only aggregate electricity usage information from the whole house meter. Usually two-phase or three-phase electric power is connected to residential and commercial buildings. Similarly, water disaggregation aims to discover each water use end by only knowing the hot and cold water usage from the whole house water meter. We generalize these two problems, energy disaggregation and water disaggregation, as a multiple-phase data disaggregation problem. The aim of this chapter is to identify electrical devices or water use ends from two phases of aggregated data. Unlike previous work which disaggregate devices from the sum of multiple phases, the time series information from each phase and the correlation of a device between/among phases are fully used. All of this information enables us to characterize more devices. This work makes the following contributions in the field of disaggregation:

1. It can disaggregate aggregate data from multiple phases.
2. It can separate the continuously variable loads which are mixed in electricity.
3. This approach can be used for both electricity disaggregation and water disaggregation.

4.2 Prior Work

Electricity disaggregation uses the electricity consumption level at the main entry into a building or house to infer whether a device inside the building is on or off. The features used include initial real power and reactive power [55] from a dataset which is recorded in a low-frequency range. With advances in electrical meter technology and the availability of less expensive meters, more and more features are being extracted from the high-frequency data set and used for disaggregation, such as the transient state generated when a device turns on or off [124], the raw current waveform [127], the voltage waveform [79], the transform of the current waveform [25], and harmonics of non-linear devices [25]. Even on-AC power features such as power line noises [112] are exploited jointly with AC power features like time of day, and device correlations [67] in modern systems.

Increasingly, research is being focused on unsupervised learning and semi-supervised learning algorithms because these algorithms do not require the power consumption of each device, and the power or water usages of individual devices are very difficult to obtain. It is only in the last few years that unsupervised learning algorithms have been used, including hierarchical clustering [51], factorial hidden Markov models (FHMMs) [67], additive factorial approximate MAPs (AFAMAP) [74], difference FHMMs [111], and motif mining [123]. Semi-supervised learning algorithms [64, 79] have also been proposed. In this chapter, we assume the number of devices and the number of power level states of each device are known. Hence, we formalize the disaggregation as a semi-supervised problem and provide solutions to the following three challenging problems.

1. Several devices may have the same real power, and it is difficult to distinguish these devices using only the recorded aggregated power time stamp.
2. Many devices may turn on or off at the same time.
3. Instead of having a discrete range of power levels, there are devices whose power consumption levels vary gradually, e.g., variable speed devices (VSD) and lights with dimmers. Once their power usage is aggregated with that from other devices, disaggregation becomes increasingly difficult.

Since obtaining a low-frequency dataset is more practical in real buildings, we focus mainly on real power, which can be easily extracted from a low-frequency dataset.

Water use disaggregation has emerged in recent years, and so far the applied algorithms are limited to supervised learning algorithms [24]. This chapter proposes water disaggregation as a semi-supervised learning algorithm by presuming that we know the number of water use ends and the water usage level of each water user end.

4.3 Disaggregation Formalism

I propose a semi-supervised approach for disaggregation; i.e. I assume that we know the on and off events for a short period of time for all devices or water end uses, and use that information to deduce the power levels or water usage, or to obtain the startup vectors of every device.

For our purposes, we define the disaggregation problem as follows: Given K -phase aggregated power or K aggregated water consumption time series $Y_k = y_1^{(k)}, \dots, y_T^{(k)}$, and a set of power or water related and contextual features, $f = f_1, \dots, f_T$ over a period of time T , the problem is to estimate the disaggregated power or water consumption of M devices $\hat{X}_m = \hat{x}_1^{(m)}, \dots, \hat{x}_T^{(m)}, m \in [1, M]$, such that a loss function of the sum of the power or water consumption of the M devices and the sum of the K phases of aggregated power or water consumption is minimized.

$$\min_{\hat{x}_t^{(m)}} \left\{ \sum_{t=1}^T \mathcal{L}_t \left(\sum_{m=1}^M \hat{x}_t^{(m)}, \sum_{k=1}^K y_t^{(k)} \right) \right\}, \quad (4.1)$$

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

4.4 Recursive Multivariate Piecewise Motif Mining

To solve the problem of separating a multi-dimensional time series into several time series, I propose the approach of recursive multivariate piecewise motif mining. Motif mining has been well studied in previous work [32] and [144]. Multivariate or multidimensional motif mining is further extended in [103] and [132] and [113].

Motif mining is applied to energy disaggregation in [123], in which discrete on/off events are exploited. This research enhances previous work by piecewise motif mining, where the on/off event is comprised of several consecutive data points, i.e. piecewise, other than individual discrete one. Also, I use multivariate motif mining to make full use of two- or three-phase aggregated data.

The framework of recursive multivariate piecewise motif mining to energy disaggregation is illustrated in Figure 4.1. The input includes multiple-phase aggregated data, such as two-phase data Mains1 and Mains2, and the power levels of each device. During the whole procedure, I recursively apply piecewise motif mining to two-phases and single phase diffs data. The first step is to identify electrical devices which draw power from both phases. Generally these devices consume a large amount of power, such as the water heater indicated by the blue line. These devices draw equal power or disparate power from both phases synchronously. Secondly, I remove the power consumption of the devices which draw power from both phases. This action decreases the noise interference caused by large power consumption and increases the possibility to disaggregate more devices with low-power consumption. Then we apply piecewise motif mining to single-phase data

to separate the devices that draw power only from that phase, such as the humidifier indicated by the green line. Generally, multivariate piecewise motif mining is divided into four steps, as shown

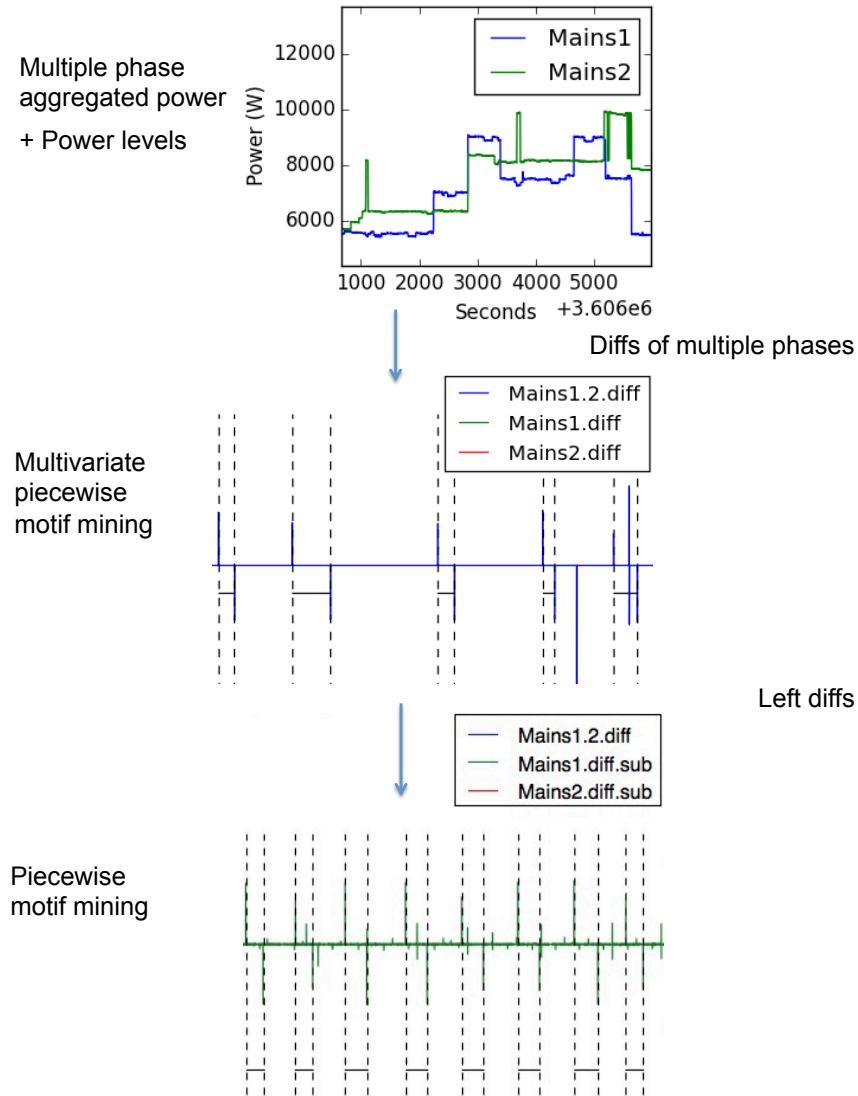


Figure 4.1: Recursive Multivariate Motif Mining Approach.

in Figure 4.2. Step 1 is to search for piecewise events from the two-phase or three-phase data. Step 2 is to encode events from multiple phases. Step 3 aims to mine frequent motifs from the encoded events list. The last step targets to recover devices from mined motifs.

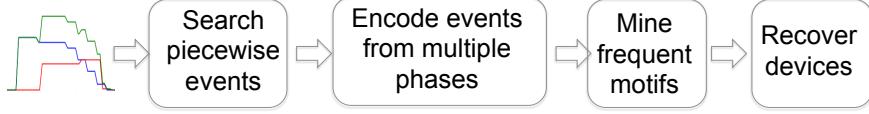


Figure 4.2: Multivariate Piecewise Motif Mining.

4.4.1 Piecewise Motif Mining

Motif mining aims to uncover the repetitive patterns in time series data, and works best for discrete events. Piecewise motif mining is proposed for energy disaggregation to detect on/off events.

Definition 1. Piecewise Event Given a time series diffs data $y_1, \dots, y_{n'}$, where $\forall |y_i| < \eta$. A piecewise event is the sum of these n' number of diffs data, $e = \sum_{i=1}^{n'} y_i$.

Each piecewise event corresponds to an on/off event of an electric device. The value of η is the noise range of each device, which is usually less than the 10% of $|e|$.

Piecewise Events Search from Multiple Phases

The majority of electrical devices which draw power from multiple phases consume larger amounts of power than electrical devices which connect to single phase. To disaggregate such a device, we need to discover specific on/off events features to separate them. Generally such an electrical device draws power from multiple phases synchronously and constructs a pattern. Some devices may consume equal power from both phases all the time, and so their power consumption patterns from both phases keep the same. Other devices may show different power usage patterns when drawing power from two phases. Algorithm 1 describes how synchronized events from two

Algorithm 1 Search Synchronized Events from Two-phase Aggregated Diffs Data

Require: 2-phases aggregated diffs data $y_k = y_1^{(k)}, \dots, y_n^{(k)}$ and $k = 1, 2$, big power consumption threshold θ

```

1: for  $i = 0 : n - 1$  do
2:   if  $|y_i^{(1)}| > \theta$  then
3:     for  $j = i - 5, i + 5$  do
4:       if  $|y_j^{(1)}| \in [|y_j^{(2)}| * 0.8, |y_j^{(2)}| * 1.2]$  then
5:          $e_i^{(1)} = e_i^{(1)} + y_j^{(1)}$ 
6:          $e_i^{(2)} = e_i^{(2)} + y_j^{(2)}$ 
7:       end if
8:     end for
9:      $e_i = e_i^{(1)} + e_i^{(2)}$ 
10:   end if
11: end for  $e_1, \dots, e_i, \dots, e_{n'}, \forall e_i > 2 * \theta$ 

```

phases are revealed. This input include the two-phase aggregated diffs data and big power consumption threshold θ . This threshold guarantees we only discover big power consumption devices. We review the phase-1 diffs data. If any absolute value $|y_i^{(1)}|$ is greater than θ , both previous and posterior five diffs data points from time i are checked. For these 10 points values, at each time j , if the difference between phase-1 $y_i^{(1)}$ and phase-2 $y_i^{(2)}$ is in the range of $0.2 * |y_i^{(1)}|$, we assert that the diffs data points from these two phases are relatively the same and synchronized. The synchronization implies that these two identical amounts power consumption comes from a single device. Therefore we sum the synchronized power level diffs data and compute the power consumption at time i as e_i . When $e_i > 0$ that denotes an on event, and $e_i < 0$ means an off event for a certain device.

Next we transfer these two-phase diffs data into an ordered on/off event list $e_1, \dots, e_{n'}$, then we apply motif mining to this events list. By matching the devices which consume power greater than $2 * \theta$, we can separate all devices which draw equal amount of power from two phases.

4.4.2 Encoding Events From Multiple Phases

After deleting all the synchronized events from phase-1 and phase-2, we apply multivariate piecewise motif mining to the remaining phase-1 and phase-2 diffs data, to detect devices which consume large amounts of power and draw power from two phases synchronously yet unequally. There are different power drawing patterns from these two phases. We encode these two-phase diffs data, which occur at the same time, as a new event e . Figure 4.3 gives an example of how the events from two-phase circuits are encoded. We extract an event which consumes power greater than θ , then we check five more data points before and after it. The values of the 11 data points relevant to this event in Main1.diff are $[0, 0, -18, 18, 1093, 1830, -196, -68, -37, -36, 0]$. The concurrent events listed in Main2.diff are $[0, 0, 0, 18, 9, 1946, 440, -51, -36, -36, 0]$. Since the events at the peak occur in the two phases as $(1830, 1946)$, and the difference of these two powers $1946 - 1830 = 116$ is in the $0.2 * 1830$ range, we consider that these two changes may come from a single device. When looking for insight into these two vectors, we observe that the sum of the changes of phase 1 is 2604W, and the sum of the changes of phase 2 is 2290W. They are in the same range, i.e. $2604 * 0.8 < 2290$. Therefore, we declare that the power changes from these two phases definitely come from a single device. We select two of these values and encode them as $e_{1'} = (1093, 9)$, $e_{2'} = (1830, 1946)$.

The piecewise events for this single device are $e = [e_{1'}, e_{2'}]$. Applying frequent motif mining, we separate this large power consumption device which draw power from two phases unequally.

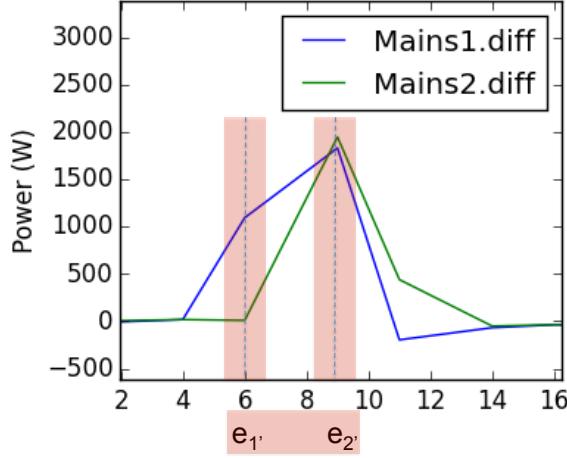


Figure 4.3: Encoding Events from Multiple Phases.

4.5 Evaluation

I use precision, recall and F-measure in our evaluation. The standard definitions 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.

Let us suppose there is a ground truth time series x with length T , and denote the corresponding disaggregated time series by \hat{x} . For any time $t \in (0, T)$, there are two values: the ground truth value of device m is $x_t^{(m)}$ and the disaggregated value $\hat{x}_t^{(m)}$. We define a parameter ρ for the range of true values $x_t^{(m)}$, and another parameter θ as the noise. For any given measurement, there are four total power values or water usage values of device m at each point: true positive $TP^{(m)}$, false negative $FN^{(m)}$, true negative $TN^{(m)}$, and false positive $FP^{(m)}$.

1. When $x_t^{(m)} > \theta$ and $\hat{x}_t^{(m)} > \theta$, at this point the disaggregation is a true positive. There are three situations in turn:

1.1. When $x_t^{(m)} \times (1 - \rho) < \hat{x}_t^{(m)} < x_t^{(m)} \times (1 + \rho)$, then

$$\begin{aligned} TP^{(m)} &= \hat{x}_t^{(m)} \\ FN^{(m)} &= FP^{(m)} = TN^{(m)} = 0 \end{aligned}$$

1.2. When $\hat{x}_t^{(m)} < x_t^{(m)} \times (1 - \rho)$, then only the disaggregated power or water usage is considered as true positive and the power or water usage that is not disaggregated is regarded as a false negative:

$$\begin{aligned} TP^{(m)} &= \hat{x}_t^{(m)} \\ FN^{(m)} &= x_t^{(m)} - \hat{x}_t^{(m)} \\ FP^{(m)} &= TN^{(m)} = 0 \end{aligned}$$

1.3 When $\hat{x}_t^{(m)} > x_t^{(m)} \times (1 + \rho)$, then the disaggregated power or water usage is a true positive, and those values which are greater than the truth values are treated as false positive.

$$\begin{aligned} TP^{(m)} &= \hat{x}_t^{(m)} \\ FP^{(m)} &= \hat{x}_t^{(m)} - x_t^{(m)} \\ FN^{(m)} &= TN^{(m)} = 0 \end{aligned}$$

2. When $x_t^{(m)} > \theta$ and $\hat{x}_t^{(m)} < \theta$, at this point the disaggregation is a false positive. Then,

$$\begin{aligned} FP^{(m)} &= x_t^{(m)} \\ TP^{(m)} &= FN^{(m)} = TN^{(m)} = 0 \end{aligned}$$

3. When $x_t^{(m)} < \theta$ and $\hat{x}_t^{(m)} > \theta$, at this point the disaggregation is a false negative. Then,

$$\begin{aligned} FN^{(m)} &= x_t^{(m)} \\ TP^{(m)} &= FP^{(m)} = TN^{(m)} = 0 \end{aligned}$$

4. When $x_t^{(m)} < \theta$ and $\hat{x}_t^{(m)} < \theta$, at this point the disaggregation is a true negative. Then,

$$TP^{(m)} = FN^{(m)} = FP^{(m)} = TN^{(m)} = 0$$

In our experimental dataset, we set $\theta = 100$ and $\rho = 0.2$. Although the maximal power consumption of all these devices is 11000W, we can still set $\theta < 11000 * 0.1$ because we apply multivariate piecewise motif mining recursively, so the devices which consume a large amount of power are deleted in the first few rounds. Therefore the power noise which is caused by the high-power electronic devices is greatly decreased.

4.6 Experiments

We run experiments on the dataset Study10 from the University of Virginia on electricity disaggregation. This dataset collects data from 02/10/2014 to 02/21/2014 in a residential building. Two individuals were asked to live in an instrumented home for around two weeks. To ensure the data consisted of the personal usage patterns of the participants, they were encouraged to live in and use the home as they normally would use their own. An eMonitor [44] sensor was used to collect both mains data for testing and circuit-level information for ground truth. Additional data, such as the opening of appliance doors and the flicking of light switches, was collected to provide sub-circuit level ground truth information for events such as lights. Both the two-phase aggregated data and each device's data are collected at intervals of 2-3 seconds. In total, 25 devices were connected to two phases at the entry of the house. Five of these devices are seldomly operated; less than five times. Fourteen devices consume power less than 100W, and the majority of them are lights. The largest power consumption of these devices is 11000W by indoor heating. The noise caused by the heating device is large; greater than 100W. Therefore we focus on disaggregating the six major electronic devices with power levels greater than 100W.

Table 4.1: Power Levels, Standard Deviation of Power Levels, On/off Duration, Connected Phases and Disaggregation Results of Electricity Devices from Study10.

Device	Power Levels	Standard Deviation	On/off Duration	Phase	Recursive Multivariate Motif Mining			AFAMAP		
					Precision	Recall	F-measure	Precision	Recall	F-measure
HeatingIndoor+ HeatingOutdoor	10590W	1270W	60s	1+2	0.979	0.928	0.953	0.870	0.45	0.598
Waterheater	4450W	350W	2-5s	1+2	0.999	0.997	0.998	0.627	0.882	0.733
Humidifier	1470W	90W	10s	1	0.997	0.992	0.995	0.725	0.858	0.787
Microwave	1850W	200W	10s	2	0.95	0.758	0.843	0.032	0.819	0.06
Dryer	5200W	400W	2-5s	1+2	0.911	0.996	0.952	0.011	0.561	0.021
	875W	225W	2-5s	1						

4.6.1 Electricity Disaggregation

We assume that we know the power levels of each device. If the power levels of each device are unknown, we can use the sum of two-phase aggregated data and the on/off events of the ground truth to extract them. We set a window size $w = 30s$ ahead and behind of the ground truth events to match the aggregated data. If there is only one power change in the aggregated data during these 60 seconds, this power level change must come from an on/off event of an electrical device. Usually, it takes around 2-5 seconds for an electrical device to reach a steady power level. The on and off events reflect different durations for a device to reach a steady state. Therefore, we measure the minimal duration of the on event and off event of each device. After we go over all the aggregated data and ground truth on/off events, we run a Gaussian mixture model to model the positive power changes and negative power changes independently. The means and standard deviations correspond to the on/off event of each device. The power levels, standard deviation, and on/off duration of each device of dataset study10 are listed in Table 4.1.

We apply recursive multivariate piecewise motif mining to dataset Study10 and compute the precision, recall and F-measure. Devices which draw power from both phases are separated first. They are heatingIndoor, waterheater and dryer. Figure 4.4 (a) gives an example of an on event in the two-phase Mains1 and Mains2. Mains1.diff denotes the diffs data from Mains1 and Mains2.diff represents the diffs data from Mains2. Mains1.2.diff shows as blue when Mains1 and Mains2 share the similar power changes. We can see that the power consumption of a specific device jumps twice in two phases simultaneously. The first time, both phases jump 2572W. After nine seconds, the power of both phases increases 2520W. The sum of these four changes is 10184W. Compared with the power levels of all devices, we speculate that these power changes are caused by the device heatingIndoor. Figure 4.4 (b) shares the same snippet of time series as Figure 4.4 (a). The red line indicates that the on event of heatingIndoor is recognized. Similarly, the off event plunges twice in two seconds -2877W and -1759W in both phases, as shown in Figure 4.4 (c). The sum of this off event is -9272W. After matching the power levels, we categorize it as the off event of heatingIndoor as indicated in Figure 4.4 (d).

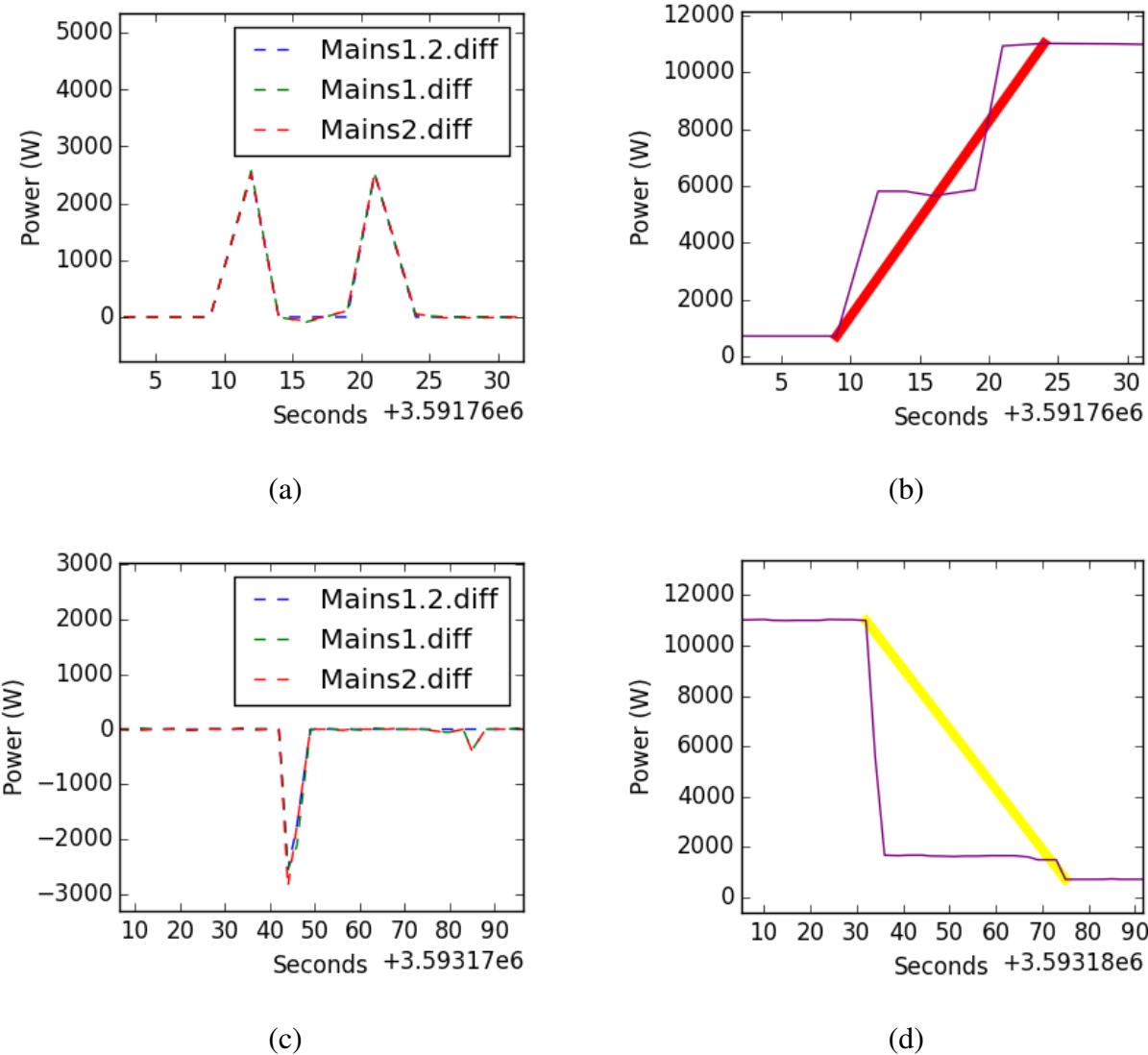


Figure 4.4: (a) On piecewise event and (c) Off piecewise event of heatingIndoor. heatingIndoor is disaggregated by motif mining the on event (b) and off event (d).

The dryer has the same power level as the waterheater at around 4800W. If we disaggregate these two devices from the sum of the two phases, it's difficult to distinguish them, but with multivariate piecewise motif mining, these two devices can be distinguished.

Figure 4.5 (a) and (b) are the diff data of the dryer and waterheater from the two-phase circuit. We can see that the waterheater draws power from Phase 1 and Phase 2 at the same time, but the dryer shows a different pattern. It draws power from Phase 1 at a lower power of 1093W, then jumps to 1830W; at the same time, it draws power from Phase 2 at the high level of 1946W immediately. We encode the power usage as shown in Figure 4.3, then apply motif mining to disaggregate them.

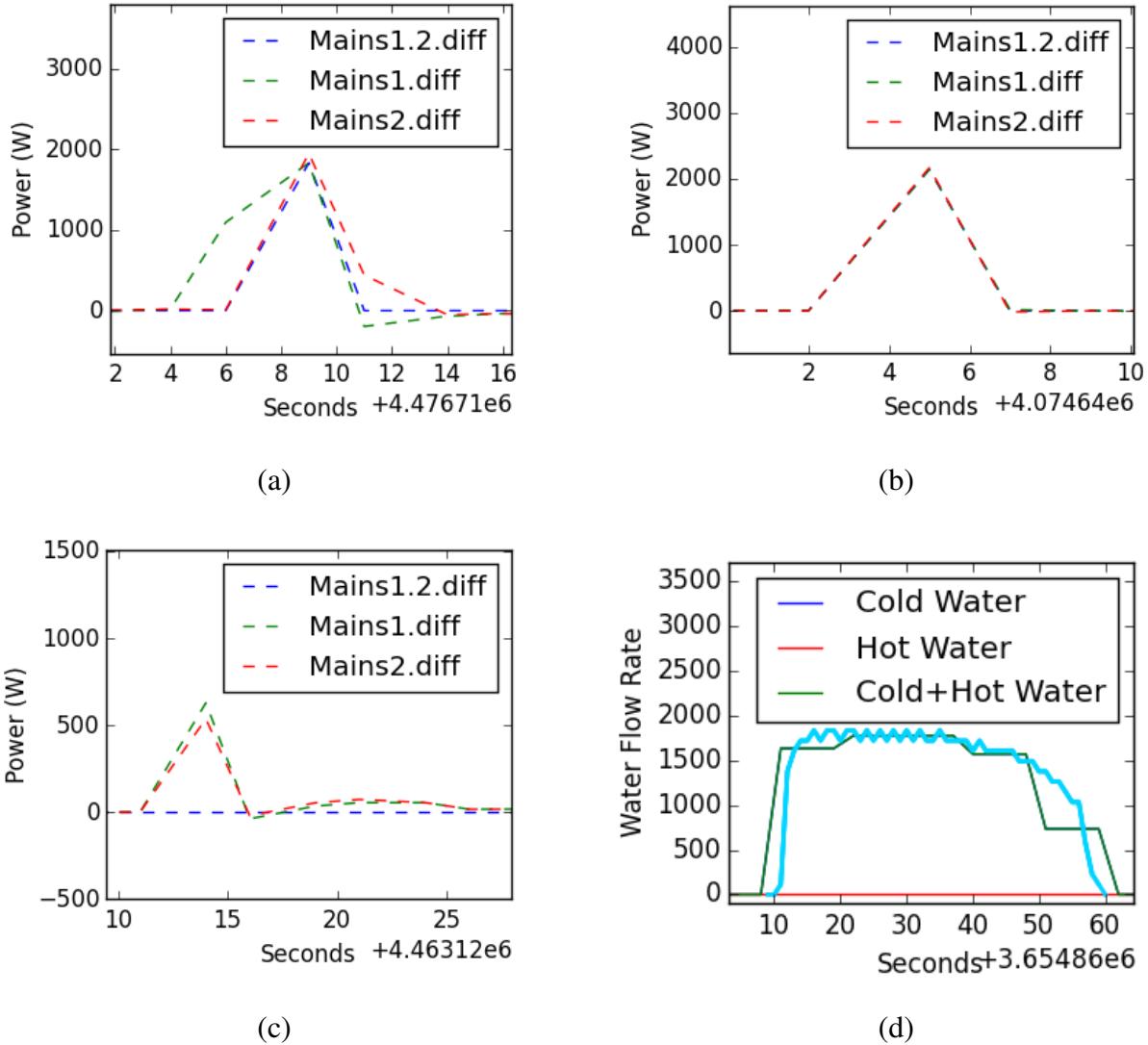


Figure 4.5: Disaggregating dryer and continuous variable load heatingOutdoor with multivariate motif mining. The on event of a device and the corresponding diffs in the two phases for (a) dryer, (b) water heater, (c) heatingOutdoor. (d) Disaggregating the toilet water use end with dynamic time-warping subsequence search. This Y-axis is water flow rate in $10000 \times \text{liter}/\text{minute}$.

After deleting the power consumption from both aggregated phases, we apply piecewise motif mining again to a single phase. We then discover the humidifier from Phase 1 and the microwave from Phase 2. When we only disaggregate the sum of Phase 1 and Phase 2, the precision recall result of the microwave and humidifier is not very accurate because sometimes their power consumptions are similar. However, using multivariate motif mining, we can separate them very clearly with good precision and recall. The precision and recall results for the data set Study10 are listed in

Table 4.1.

Recursive multivariate motif mining is capable of disaggregating continuous variable loads. Figure 4.5 (c) shows the diff data of heatingOutdoor from the two phases. During this event, its power levels change nine times, then continue at a relatively stable state. By applying piecewise motif mining, we can successfully identify this as the heatingOutdoor device after matching its power level. If another device D which draws from Phase 1 or Phase 2 is turned on or off during this period, multivariate piece-wise motif mining can still identify this heatingOutdoor device. This is because D only uses one phase's power; hence its power change is not counted in our piecewise event.

4.6.2 Water Disaggregation and Constraints

Water usage displays different characteristics. The total water consumption is zero most of the time. Whenever a water use end is operated, water is consumed intensively for a period of time. Then it will stay off for a much longer time. We observe that the operations of water use ends reflect a series of user behaviors. For instance, a person may use the toilet in the bathroom first, then wash hands in the sink and finally take a shower afterwards.

Similar to electricity disaggregation, we use a period of aggregated water usage data to extract features and obtain the water flow rate level of each water use end. Table 4.2 lists the water consumption rate for each device. For instance, taking a shower uses hot water at a flow rate between 0.1822 liter/min and 0.1986 liter/min. Let $\frac{\alpha}{10000}$ denotes this range of water flow rate. The total hot and cold water consumption by shower is 0.1904 liter/min. Therefore, the cold water flow rate caused by shower is $0.1904 - \frac{\alpha}{10000}$ liter/min. Turning on the water for the shower takes around two seconds.

Table 4.2: Water Flow Rate Levels of Water End Uses.

Device	Hot water (liter/min*10000)	Cold water (l/min*10000)	Duration (second)
Shower	$\alpha \in (1822, 1986)$	$1904 - \alpha$	on: 2
Washing Machine	$\alpha \in (1988, 2276)$	$2132 - \alpha$	on: 5
DownToilet	0	(1270, 1400)	whole: 50
UpToilet	0	(1480, 1700)	whole: 50

After these calculations, we apply a multivariate piecewise motif mining approach to water disaggregation. For the shower and washing machine, the total flow rate of hot and cold water is high, nearly 0.2 liter/minute. Therefore by only searching the total hot and cold water flow rate, we can identify these two devices. The event of shower usage usually lasts for more than one minute, but the washing machine uses water for less than one minute, repeating six to nine times. Both the

shower and washing machine use hot water and cold water. However, the washing machine uses hot water for only the first one or two times. For the rest of its cycle, only cold water is used. Whenever the washing machines starts, the power consumption starts as well.

Applying piecewise motif mining to the water usage lets us disaggregate the shower and washing machine. The precision, recall and F-measure for the shower disaggregation are 0.999, 0.972, and 0.986, and the precision, recall and F-measure for the washing machine disaggregation are 0.997, 0.969, and 0.983. However, with a variable water flow rate, piecewise motif mining has limitations in handling water use ends such as the toilet. Therefore we use the dynamic time warping subsequence [116] search as a complementary to discover these water use ends. For the two toilets, we apply dynamic time warping to match the time series. The water usage results of one toilet, UpToilet, is shown in Figure 4.5 (d).

We can see that multivariate piecewise motif mining is capable of disaggregating water use ends which have sharp on/off water flow rates. However, it has limitations in dealing with water use ends with irregular water use patterns, such as toilets and sinks. Since the water usage of toilets is relatively fixed if used alone, some toilet water usages can be disaggregated by using the dynamic time warping subsequence search which was researched in [107].

4.7 Conclusion

This chapter proposes a semi-supervised recursive multivariate piecewise motif mining approach to electricity disaggregation. We use data from a period of time to find the features of individual devices, including the power level and standard deviation. Based on these features, we recursively utilize multivariate motif mining to uncover devices that draw power from both phases equally or unequally, then separate the devices from each phase using motif mining. Devices that use a large amount of power are removed from the two phases in the first few rounds, which brings the benefit of decreasing the noise caused by these devices during the process of disaggregating from single phase. Therefore more devices with smaller power consumption are separated using the piecewise motif mining from a single phase. In addition, this piecewise motif mining approach can identify continuously variable loads such as outdoor heating. Furthermore, when we apply motif mining approach to water disaggregation, it can separate water use ends that have steady water usage, such as a shower, but cannot disaggregate those water use ends which consume water variably for the whole cycle, like a toilet.

In the future, for energy disaggregation we will use more features from the aggregated data from the multiple phases, such as the startup shape of individual devices. For water disaggregation, we will explore how to integrate the dynamic time warping with multivariate piecewise motif mining.

Chapter 5

Occupancy Prediction

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, it would be useful to automatically adjust the HVAC of residential buildings based on occupancy. In this work, we mine people's energy activity profile to predict the occupancy of residential buildings. I propose a novel hybrid method, which uses episode mining for target event detection and a mixture of episode-generating HMM (EGH), combined with the standard kNN approaches and demonstrate how this hybrid approach always yields the best results.

5.1 Introduction

Modeling activity of daily life (ADL) has become a burgeoning research topic, since people demand a comfortable life at home at a lower cost. Since heating and cooling spaces consumes $\sim 53\%$ of the total electrical usage by heating and cooling spaces of an average household, automating the operation of 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 [70], smart phones' GPS data [71], electricity consumption [46] and sensor data by tracking indoor activities [8, 120] are effective ways to approach this prediction problem. Prediction of occupancy using sensor data has been broadly researched. By capturing daily activities, like room occupancy of the house, usage of electrical devices, and usage of water systems using sensors, researchers have modeled occupancy [17, 46, 95] and used these results to automate the control of the HVAC system.

Although the supervised learning kNN [120], neural network [95] and Markov model [46] are effective, the detailed household activities represented as a time series are not fully utilized. Daily activities such as waking up, cooking, washing, and commuting to work/school and back 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 episode mining analysis, which can be used to predict household occupancy. Using 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). Unlike previous models, the time and order of each kind of activity are fully utilized. Third, the algorithm predicts according to the scenario-based probabilistic model episode generative HMM (EGH). The prediction accuracy is better than the existing models.

This work's 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 the 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 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 unoccupancy 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 HVAC controlling system, the HVAC can be turned on or off ahead of the occupancy change. Since the HVAC does not work during occupancy, this saves electric usage.

5.2 Related Work

Accurately predicting whether a home is occupied is a difficult task. People in the same home have different daily schedules; some go to work and others stay at home for a period of time. A great deal of research has been done to track the activities of people to infer the home occupancy. Researchers have made efforts to collect data by sensors, smart phones, the calendar, and weather information. Most of the approaches that model and predict occupancy primarily use sensor data to detect conditions such as room occupancy, use of electrical appliances, water usage, etc. Several supervised learning approaches, such as kNN, neural networks, rule-based models, and Markov chain models have been used to model and predict building occupancy [8, 17, 46, 95, 120]. Using the kNN supervised learning algorithm and monitoring sensor data for a portion of the day, Scott et al. predict an entire day's occupancy in [120]. A neural network approach using a binary time series based on occupancy/unoccupancy along with exogenous input network (NARX) is proposed in [95]. Mahmoud et al. tackle the problem by presenting a non-linear autoregressive model with

an exogenous input (NARX) network. Several Markov chain models, like the blended Markov chain, closest distance Markov chain, and moving-window Markov chains are presented in [46]. A mixture of multi-lag Markov chains was used to predict the occupancy of single-person offices [98]. In that work, the authors also compare their model with the Input Output Hidden Markov Model, First Order Markov Chain and the NARX neural network.

A recent survey [70] compares major occupancy predictions algorithms against the LDCC dataset [69], which was collected by GPS and other sensors. It shows that time-based presence probability [76] performs slightly better than the preheat kNN approach [120]. Since the preheat kNN approach [120] is more widely applicable, in that it can be used against both GPS and sensor datasets, we set it as a baseline method for comparison.

5.3 Problem Formulation

Given M time series, each time series $X^{(m)} = X_1^{(m)}, \dots, X_t^{(m)}, \dots, X_T^{(m)}$ represents a sequence of room occupancy of person m inside a home over K days, where $X_t \in s$ denotes that X belongs to a finite room set s at the sequence number of t , and $m \in \{1, \dots, M\}$. Let Z denote that the home is unoccupied and $Z \in s$. We predict whether person m stays at home the rest of a day from time T , i.e. during $T + 1, T + 2, \dots, \Delta T$,

$$\hat{Y}^{(m)} = \hat{Y}_{T+1}^{(m)}, \dots, \hat{Y}_{T+\Delta T}^{(m)} \quad (5.1)$$

where $Y_{T+\Delta t}^{(m)} = Z$ if person m does not stay at home at time $T + \Delta t$; otherwise, $Y_{T+\Delta t}^{(m)} \neq Z$. If any person m stays at home $Y_{T+\Delta t}^{(m)} \neq Z$, then this house is occupied $Y_{T+\Delta t} \neq Z$.

5.4 Temporal Mining Mixture Model

We use a three-pronged approach to tackle the problem of mining and predicting unoccupancy, as shown in Figure 5.1. Given indoor activities time series of a person over a period of time, first, we

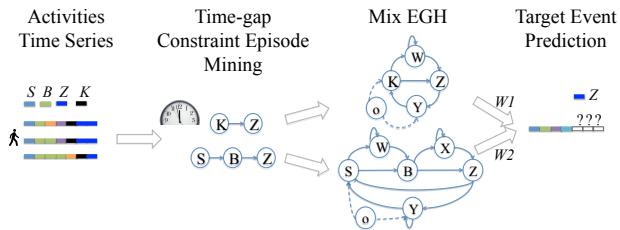


Figure 5.1: Occupancy Prediction Framework.

use an episode mining algorithm to discover frequent episodes from the past days' data. Then we

connect each episode with an EGH and build a mixture EGH model. Based on the mixture model, we predict when each person will leave and come back the house. If all people leave, then the house is unoccupied.

Episode An episode is a collection of ordered events. Here an episode refers to an ordered events which are highly relevant to the occupancy status inside a building. 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 $\alpha = \langle X_1, \dots, X_t, \dots, X_T \rangle$, where X_t denotes that X occurs at a sequence of t . The event X_t may be the point event or dwelling event. The dwelling event has a start time $X.start$ and end time $X.end$. In this paper X denotes a dwelling event and represents which room a person stays inside a building, i.e., this building is *occupied*. Since Z denotes a room is unoccupied, $Z.start$ is the point at which a person or all people inside the building leave, and $Z.end$ is the point at which a person or all people come back.

5.4.1 Time-gap Constraint Episode Mining

Episode Mining Episode mining has been studied in previous research [99]. It uses a non-overlap mining approach to find the frequent episodes. Episode mining has been applied to energy disaggregation to help conserve energy in buildings [123] in sustainability research. In contrast with previous research, the events in this application of occupancy prediction dwell at an event for a period of time. As a result, we extend the above two episode mining algorithms [84, 114] and enforce more constraints. One change is to adopt right alignment for the first element in the episode mining. The second modification is to add time constraints and apply gap duration constraints between two consecutive events inside an episode. Figure 5.2 shows a time-gap constraint episode mining example. Let us assume we have a frequent episode $S \rightarrow B \rightarrow K \rightarrow Z$. We then 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. In addition, we set gap duration between any two consecutive events. The gap duration of SB is calculated as $\Delta SB = B.start - S.end$. We set the maximal gap time between SB , BK , and KZ as 10 minutes, 40 minutes and 100 minutes; the minimal gap time is 0. Then we have a stream composed of the sequence of dwelling events "Event seq," as shown in Figure 5.2. The time-gap constraint episode mining process to discover a frequent episode uses the following method. Let the *node* structure denotes each element in any episode, as depicted as a square box in Figure 5.2. Let *waits* refer to a structure which pairs with an episode and has the same length of this episode. Initially, a *waits* structure related to episode $S \rightarrow B \rightarrow K \rightarrow Z$ is created. 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 S_1 is checked. Since S_1 is in the range of 3 – 6 hours, S_1 passes and is put into the node structure *node* related to S . Next a new *node* structure is created to wait for $B\langle 2, 20 \rangle$. When $T = 2$ and $T = 3$, both B_2 and B_3 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 and 10 minutes. These two nodes B_2 and B_3 are then input into the *waits* structure. At the same time, a new *node* structure is created for $K\langle 5, 60 \rangle$. When $T = 4$,

Event Seq: S1<12:30,7:00>, B2<7:05, 7:09>, B3<7:10,7:20>, K4<7:22,7:24>,
 D5 <7:30,7:50>, K6<7:50,7:55>, L7<7:55,5:58>, Z8<8:00,8:02>,
 L9<8:05,8:09>, Z10<8:10,18:05>,

Episode: S<180,360> -[0,10]-> B<2,20> -[0,40]-> K<5,60> -[0,100]-> Z<180,540>

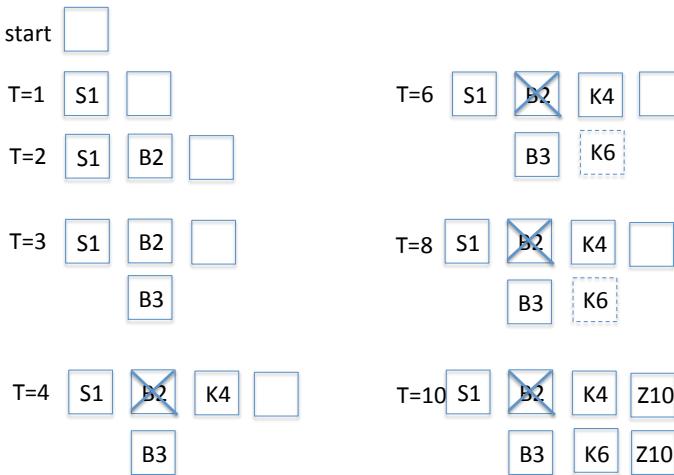


Figure 5.2: Time-gap constraint episode mining example.

the gap between $\langle B_3, K_4 \rangle$ is satisfied with the distance condition between B and K 0-40 minutes. However, the gap between $\langle B_2, K_4 \rangle$ is longer than the constraint gap. Therefore, B_2 is canceled out. Now a new Z waits for the symbol $Z\langle 180, 540 \rangle$. When $T = 6$, the gap from B_2 to K_6 is too large. Therefore, K_6 is not added into the *node K* structure in *waits*. When $T = 8$, the duration of Z_8 does not qualify for the condition of between 3-9 hours, so Z_8 is not added. When $T = 10$, the duration of Z_{10} meets the requirement of between 3-9 hours and its distance to K_4 meet the requirement of $\Delta KZ \in [0, 100]$ minutes. Thus Z_{10} is added into the *node Z* structure in *waits*. Therefore, complete mining of an episode is complete, and we have mined two instances here; $S_1B_3K_4Z_{10}$ and $S_1B_3K_6Z_{10}$.

5.4.2 Mixture EGH

Episode Generating HMM Episode generative HMM (EGH) model is a type of HMM model which connects with frequent episode, and the more frequent an episode inside a sequence, the likelihood of the state sequence including this episode is larger [84]. The uniqueness of the EGH is that the transition matrix and emission matrix is only decided by a noise parameter η . The noise parameter η of frequent episode α is calculated as $\eta = \frac{T-Nf_\alpha}{T}$, 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 .

In the mixture EGH model shown in Figure 5.3, the transition matrix of an EGH is given as an example. Let us assume we have a N -node frequent episode $S \rightarrow B \rightarrow Z$, where $N = 3$. We define $2N$ number of hidden states; N for episode states, and N for noise states. The noise states are $\{W, X, Y\}$. 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 $1 - \eta$. To calculate the emission matrix, first we let M denote the total number of symbols in the event stream. For any hidden states in the episode, M 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 $\frac{1}{M}$.

Theorem 1 from [84] is crucial. The theorems in [84] prove that the more frequent an episode is inside a sequence, the greater the likelihood of the state sequence including this episode. The proof for this theorem is explained in detail in [84].

THEOREM 1. [84] *Let $D_Z = X_1, \dots, X_K$ 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 $f_\alpha > f_\beta$, then $P(D_Z, q_\alpha^* | \Lambda) > P(D_Z, q_\beta^* | \Lambda)$ (2) if $P(D_Z, q_\alpha^* | \Lambda) > P(D_Z, q_\beta^* | \Lambda)$, $f_\alpha > f_\beta$*

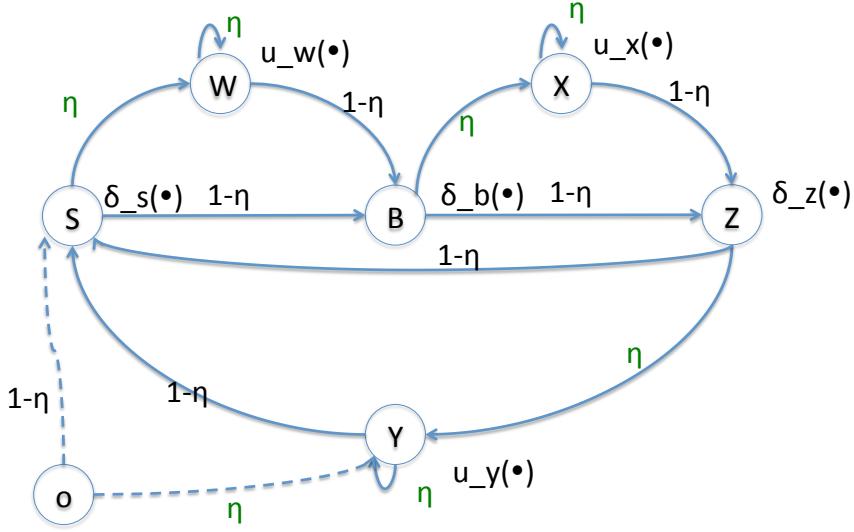


Figure 5.3: States Transition of Episode Generating HMM (EGH).

Mixture Model The mixture EGH model is fully discussed in previous work [84]. This model gives different weights to each EGH to predict a target event. We can assume we obtain whether an episode occurs on a certain day. Let $D_Z = \{X_1, \dots, X_K\}$ denote the K days data set. $F = \{\alpha_1, \dots, \alpha_J\}$ denote the frequent episodes in the dataset D_Z . An EGH Λ_{α_j} is associated with frequent episode α_j . Λ_Z denotes a mixture EGH model. The likelihood of D_Z under the mixture model is written as Equation (5.2).

$$Pr(\Lambda|Z) = \prod_{i=1}^K P[X_i|\Lambda_Z] \quad (5.2)$$

$$= \prod_{i=1}^K \left(\sum_{j=1}^J \theta_j P[X_i|\Lambda_{\alpha_j}] \right) \quad (5.3)$$

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 (5.2) are additive; the coefficients θ are computed by EM algorithm. During the initialization part of the EM algorithm, the episode frequency over the times series T over K days 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 (5.4); Then Bayes rules is applied to compute the new coefficient θ_{new} .

$$Pr(X_i|\Lambda_{\alpha_j}) = \left(\frac{\eta_{\alpha_j}}{M} \right)^{|X_i|} \left(\frac{1 - \eta_{\alpha_j}}{\eta_{\alpha_j}/M} \right)^{|\alpha_j| f_{\alpha_j}(X_i)} \quad (5.4)$$

In the maximization step, we update the objective value based on Equation (5.2) until it converges, i.e., until the difference of two consecutive objective values is smaller than a threshold.

5.4.3 Predict When the Target Event Occurs

Target event prediction is studied in [84], but only insofar as it predicts whether a target event will occur, rather than when the target event will happen. Our occupancy prediction algorithm enriches the previous event prediction algorithm by breaking into three sub-problems; whether the target event un-occupancy Z will appear, when the target event Z starts, and when the target event Z ends.

Since the solution to the first sub-problem is similar to previous work, this sub-section emphasizes on the last two sub-problems. After obtaining the result of the first sub-problems, we assume we already know that the target event Z will surely happen, and so we need to predict when the person leaves or comes back. The leaving time corresponds to the start time of dwelling event $Z.start$ and the returning time refers to the end time of dwelling event $Z.end$.

After running episode mining and mixing the EGH model, we have obtained all the frequent episodes $F = \langle \alpha_1, \dots, \alpha_J \rangle$, the corresponding EGH $\Lambda_{\alpha_j}, j = 1 \dots J$ with noise parameter η_j , and the mixture models Λ_Z with coefficients θ_j . We use the coefficient of these mixture models to predict the leave time and return time of target event Z . Each day is cut into three phases: 1) Before a person gets up; 2) After the person gets up but before the person goes out; and 3) After the person goes out but before they come back.

1. Usually before a person gets up, there is only one frequent episode named 'SZ'. The start time and end time of Z depends on 'S'. Therefore $Z.start$ and $Z.end$ are calculated by the probability density function of going out and coming back time in the past days.
2. After the person gets up, if he/she has a lot of activities at home, there are several frequent episodes that are mined before the person leaves home. If there are several frequent episodes ending with Z , the leave time and return time of each episode is checked to determine whether they are in a range of probability density function (PDF) value in the past. If yes, the mean value of these episodes are recorded. Since each episode generates an EGH, the mixture EGH model computes a weight for each EGH as coefficients. The leave time $Z.start$ and back time $Z.back$ are the weighted mean leave time and back time of these frequent episodes.
3. After a person leaves home, we already know when the person leaves home $Z.start$. If the person has come back, nothing needs to be predicted. If the person has not come back, the return time $Z.end$ is the weighted historical return time of mined frequent episodes, viz. the probability density function of backing time based on the time-constraint going out time.

5.5 Experiment Results

We have conducted experiments on three datasets, where each dataset is obtained by monitoring 24-hour activities of two adults in a house via RFID. All these activities occur in twelve rooms;

the basement, bathroom, bedroom, dining room, hallway, kitchen, living room, mudroom, nursery, outside-front, outside-back, and upstairs. The dataset comprises events in the form of timestamped room occupancy data points. For instance, an event can correspond to person 1 being in the kitchen at 7:00 am. The summary of these three datasets is shown in Table 5.1. We define *unoccupancy*

Table 5.1: Datasets summary.

Dataset	Number of entries	Period(day)	Start date
study10	6596	12	02/10/2014
study11	1696	10	01/29/2014
study14	3453	13	12/09/2013

of a person as one of these conditions: the person leaves the *outside-front* or *outside-back* for more than 30 minutes; the person stays in the living room or dining room for more than 9 hours without any other activities; or the gap between any two events is more than 30 minutes. Since our research goal is to automate the turning on and off of the HVAC system at least 30 minutes before occupancy, the first and third constraints are in place. We are only interested in events where the *unoccupancy* period is for an extended duration (> 30 minutes). The second constraint comes from our observation that if a person stays in one room for more than 9 hours without moving to other rooms, this usually means that the person has gone out but left the RFID equipment at home. Furthermore, we delete events with a duration less than 2 minutes since these correspond to the individual walking back and forth across rooms and generally do not contribute to meaningful episodes. We conduct four types of experiments to compare four approaches; kNN, mixture EGH, PDF, and support vector regression (SVR). For each dataset, we use 2/3 of the data for training and the remaining 1/3 of the data as a test. Following the approach in [120], we organize one day's data into 96 15-minute chunks. For the test data, we assume that we only know some of the 15-minute chunks. Our target is to predict the occupancy in the rest of the day, or 30 minutes ahead.

5.5.1 Occupancy Prediction of Individuals

The individual occupancy prediction results on datasets Study10 and Study11 are summarized in Table 5.2. In Study10, the mixture EGH performs better than kNN for the occupancy prediction for person2 on 02/17/2014. Furthermore, mixture EGH outperforms kNN for both persons on 02/20/2014. However, for person1 on 02/19/2014, kNN works a little bit better. When checking the original data on this test date, we find that the activities on this date are very similar to the historical activities in the training data. This observation leads us to the conclusion that when the test data is very highly similar to the historical data, the kNN approach sometimes performs a little better. In Study11, mixture EGH gets higher precision, recall and f-measure scores on 02/05/2014, but the opposite is true on 02/04/2014. We analyzed the original data to find the reason for the kNN's better performance, and found the date is an anomaly from the normal pattern, since both individuals went to sleep late that day (after 12:00am). Before sleep, person1 even stayed in the

Table 5.2: Precision Recall F-measure Comparison of Individual and Whole House Occupancy Prediction in Study 14.

Dataset	Date	Person	EGH			kNN			SVM		
			precision	recall	fmeasure	precision	recall	fmeasure	precision	recall	fmeasure
study10	02/17/2014	person2	1.00	1.00	1.00	0.99	0.98	0.98	0.71	0.76	0.71
	02/19/2014	person1	0.98	0.99	0.98	0.99	0.99	0.99	0.71	0.76	0.70
	02/20/2014	person2	0.93	0.92	0.92	0.92	0.91	0.90	0.72	0.77	0.72
	02/20/2014	person1	0.95	0.94	0.94	0.94	0.93	0.93	0.71	0.77	0.72
	02/20/2014	wholehouse	0.92	0.92	0.91	0.91	0.89	0.91	0.79	0.74	0.74
study11	02/04/2014	person2	0.93	0.93	0.92	0.95	0.95	0.95	0.71	0.77	0.72
	02/04/2014	person1	0.93	0.93	0.92	0.95	0.95	0.95	0.70	0.77	0.71
	02/05/2014	person2	0.85	0.92	0.86	0.87	0.87	0.84	0.71	0.76	0.71
	02/05/2014	person1	0.84	0.90	0.84	0.79	0.90	0.80	0.70	0.77	0.71
	02/04/2014	wholehouse	0.918	0.924	0.913	0.916	0.921	0.911	0.77	0.69	0.71
	02/05/2014	wholehouse	0.90	0.84	0.84	0.88	0.81	0.81	0.74	0.70	0.70

kitchen for around two hours. The frequent episode KZ , which represents 'kitchen-unoccupied', usually occurs in the morning instead of around midnight. However, the mixture EGH model still assumes that the KZ pattern happens during the morning; therefore the prediction results are not accurate. Since kNN ignores this fine granular activity pattern at a house and only considers the occupancy status in the past most similar five days, its performance is better. Generally speaking, the mixture EGH helps predict when a person leaves home and the period of sleeping and its performance is competitive to the kNN approach. For all these experiments, the SVR approach performs the worst because of the limitations of this approach. SVR uses the latest several data points about the occupancy state as the training vector for the prediction of the next occupancy state. Here we set eight past data points as the predictor. Other features such as the time of day and day of week cannot be utilized fully.

We also conduct experiments for individuals' *rest-of-day* occupancy prediction at different times. Figure 5.4 illustrates a person's occupancy prediction result in Study10. There are three sub-figures. Each sub-figure describes the precision, recall, and f-measure of *person1* on 02/20/2014. The blue line represents the mixture EGH model, the green line represents the PDF model, and the red line denotes the kNN model. The x-axis is the number of known 15-minute chunks of the test day. For instance, at $x = 20$, we already know $20 * 15$ minutes' data and need to predict whether the home is occupied during the remaining 76 chunks. The y-axis denotes the precision, recall and f-measure values in the three sub-figures from the top down. The first sub-figure shows that the mixEGH has the highest precision, recall and f-measure on test day 02/20/2014 for occupancy prediction. The other two baseline approaches are comparable, except that kNN performs better than PDF when the person comes back home after slot 72. Looking into the original data, we find that *person1* actually comes home later than usual in the training dataset.

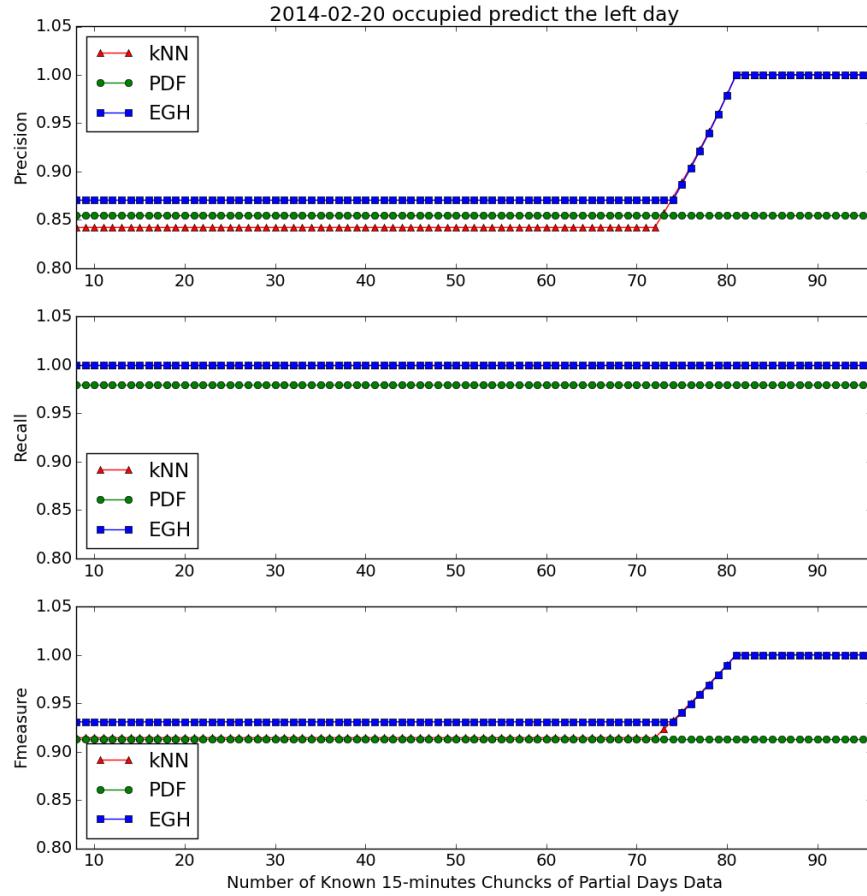


Figure 5.4: Occupancy prediction precision, recall and f-measure comparison of three approaches of person1 on 02/20/2014 on Study10.

5.5.2 Occupancy Prediction of Residential Buildings

Based on individual prediction results, we deduce when a house is occupied using logic OR operations on the prediction results of two persons. The whole house occupancy prediction results are listed in Table 5.2 and marked in bold. In Study10, the precision, recall, and fmeasure values of the whole house are 0.92, 0.92 and 0.91, respectively, which are higher than the values from the kNN approach of 0.91, 0.90 and 0.91, and of the SVR approach 0.79, 0.74 and 0.74. Similarly, the mixture EGH model outperforms kNN in Study11. Note that in Study 11 on 02/04/2015, EGH does not perform as well as kNN on individuals but performs a little bit better than kNN, and much better than SVR in occupancy prediction of the whole house. The reason behind this is because the activities of the two people inside the home are not synchronized. The mixture EGH model can predict the occupancy for each person and grasp each person's activities more accurately.

Table 5.3: Precision Recall F-measure of Individual and Whole House Occupancy Prediction in Study 14.

Dataset	Date	Person	EGH			kNN			SVM		
			precision	recall	fmeasure	precision	recall	fmeasure	precision	recall	fmeasure
study14	12/18/2013	person2	0.91	0.91	0.89	0.87	0.87	0.84	0.73	0.77	0.71
	12/18/2013	person1	0.92	0.92	0.91	0.90	0.90	0.89	0.73	0.76	0.71
	12/19/2014	person2	0.86	0.86	0.85	0.90	0.90	0.88	0.73	0.76	0.71
	12/19/2014	person1	0.85	0.84	0.84	0.86	0.86	0.85	0.73	0.76	0.71
	12/20/2014	person2	0.92	0.94	0.92	0.98	0.97	0.97	0.75	0.79	0.75
	12/20/2014	person1	0.90	0.91	0.90	0.95	0.95	0.95	0.75	0.79	0.75
	12/18/2013	wholehouse	0.91	0.91	0.90	0.88	0.88	0.86	0.75	0.72	0.70
	12/19/2013	wholehouse	0.841	0.845	0.838	0.848	0.853	0.842	0.79	0.74	0.74
	12/20/2013	wholehouse	0.92	0.90	0.90	0.94	0.93	0.93	0.74	0.72	0.70

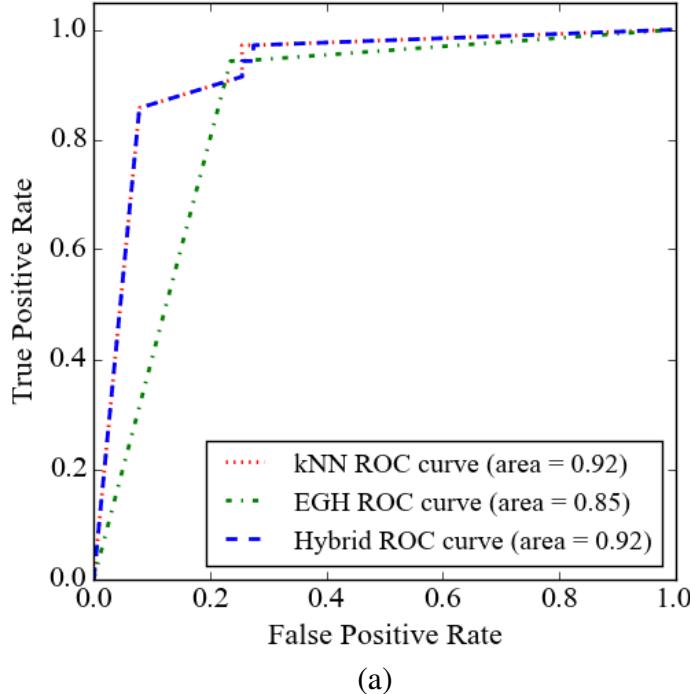
5.5.3 Limitations of Mixture EGH Model

Although the temporal mixture EGH model performs well on the datasets Study10 and Study11, the same is not true for the dataset Study14. Table 5.3 shows that, in Study14, the mixture EGH model works better for the individual and whole-house occupancy predictions on 12/18/2013 but not on 12/19/2013 or 12/20/2013. We check the activities of both individuals on these two days and find that both of them went out again after coming back and staying home for a while. Since the episodes of going out after coming back home from work do not occur frequently, mixture EGH cannot detect this pattern. Thus the occupancy prediction probability of these events is completely missing. However kNN performs well because it leverages all the historical data; therefore, even if the abnormal event occurs once, this prediction approach incorporates it and obtains the average value. To relax the limitations of abnormal events, we propose a hybrid model for prediction; when deploying this occupancy prediction in reality, for example for a prediction that is 30 minutes ahead, just 15 minutes before the prediction, if a person goes out again after coming back, the deployed system switches to the kNN approach rather than the mixture EGH model for prediction. In such cases, this hybrid model can always get the best prediction results.

5.5.4 House Occupancy Prediction 30 Minutes Ahead with Hybrid Approach

To preheat the house, we need to evaluate how much time in advance to automatically turn on/off HVAC, and the advance notice time estimation is given in [120]. Here we use prediction of 30 minutes ahead of time house occupancy. We compare the receiver operating characteristic (ROC curve) of three approaches: mixture EGH model, kNN, and a hybrid approach of the mixture EGH model and kNN. In this hybrid approach, we set mixture EGH results as the baseline, then replace the values of the mixture model by the values from kNN model in the following two situations: 1) After a person comes back home; and 2) When the prediction probability of kNN is greater than

0.8. Figure 5.5, 5.6, and 5.7 illustrate the ROC curve of the whole house occupancy prediction on 02/20/2014 of dataset Study10, 02/04/2014 of dataset Study 11 and 12/20/2013 of dataset Study14, respectively. The red and green lines represents the kNN and mixture EGH models; the blue line denotes the hybrid approach. The ROC curves show that the hybrid approach always has the largest area, namely 0.96, 0.92 and 0.92, which indicate that the hybrid approach always performs best.



(a)

Figure 5.5: ROC curve of house occupancy prediction in Study10 (02/20/2014).

5.6 Conclusion

Residential occupancy prediction is a hot research topic in controlling HVAC systems. The accuracy of occupancy prediction influences the comfort of the persons inside the home and energy savings. In order to achieve the highest prediction result, we propose to integrate the mixture EGH model and kNN together as a hybrid approach.

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.

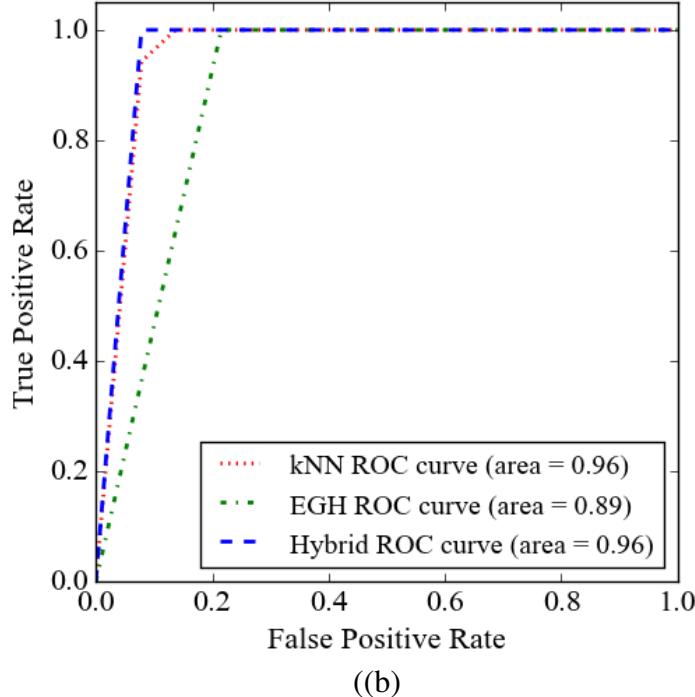


Figure 5.6: ROC curve of house occupancy prediction in Study11 (02/04/2014).

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 hydrate prediction solution performs best on the workday occupancy prediction: in case of normal activities, we apply mixture EGH model; in case of abnormal events, we utilize kNN, which is generally considered a benchmark in occupancy prediction problem.

In the future work, we will continue working on the holiday occupancy prediction. The occupancy patterns for these days are completely different. For example, on certain weekdays, a person may never go out. Therefore the occupancy prediction probably depends more on date than the indoor activities. Furthermore, we will apply this temporal mining approach to the GPS datasets [71] to check the effectiveness of occupancy prediction with different kinds of data.

Algorithm 2 Gap Constraint Episode Mining on Dwelling Events

```

1: for all event type A do
2:   Initialize  $\text{waits}(A) = \emptyset$ 
3: end for
4: for all  $\alpha \in C$  do
5:    $\text{prev} = \emptyset$ 
6:   for  $1 \leq i \leq N$  do
7:     Create  $\text{node}$  with  $\text{node.visited} = \text{False}$ ;  $\text{node.episode} = \alpha$ ;  $\text{node.index} = i$ ;
8:      $\text{node.prev} = \text{prev}; \text{node.next} = \emptyset$ 
9:     if  $i = 1$  then
10:      Add  $\text{node}$  to  $\text{wait}(\alpha[1])$ 
11:    end if
12:    if  $\text{prev} \neq \emptyset$  then
13:       $\text{prev.next} = \text{node}$ 
14:    end if
15:  end for
16: end for
17: for  $i = 1 : n$  do
18:   for all  $\text{node} \in \text{waits}(E_i)$  do
19:     set  $\text{accepted} = \text{false}$ 
20:     set  $\alpha = \text{node.episode}$ 
21:     set  $j = \text{node.index}$ 
22:     set  $tlist = \text{node.list}$ 
23:     if  $j < N$  then
24:       for all  $tval \in tlist$  do
25:         if  $(t_i - tval.init) > \alpha.t_{high}[j]$  then
26:           remove  $tval$  from  $tlist$ 
27:         end if

```

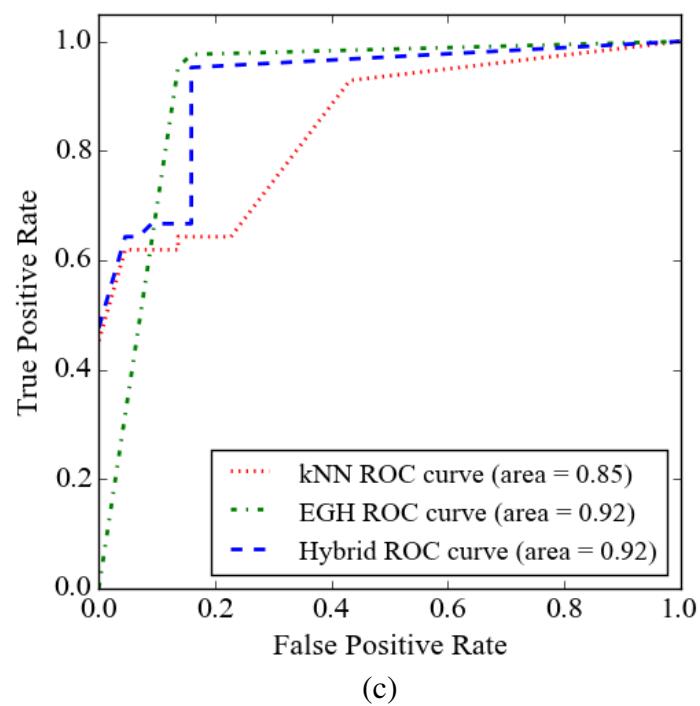


Figure 5.7: ROC curve of house occupancy prediction in (c) Study14 (12/20/2013).

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

```

27:      if  $j = 1$  then
28:          update  $accepted = true$ 
29:           $tval.init = t_i$ 
30:          add  $tval$  to  $tlist$ 
31:          if  $node.visited = false$  then
32:              update  $node.visited = true$ 
33:              add  $node.next$  to  $waits(\alpha[j + 1])$ 
34:          end if
35:      else
36:          for all  $prev_{tval} \in node.prev.tlist$  do
37:              if  $t_i - prev_{tval} \in (\alpha.t_{low}[j - 1], \alpha_{high}[j - 1])$  then
38:                  Update  $accepted = true$ 
39:                  update  $tval.init = t_i$ 
40:                  add  $tval$  to  $tlist$ 
41:                  if  $node.visited = false$  then
42:                      update  $node.visited = true$ 
43:                      if  $node.index \leq N - 1$  then
44:                          add  $node.next$  to  $waits(\alpha[j + 1])$ 
45:                      end if
46:                  end if
47:              else
48:                  if  $t_i - prev_{tval} > \alpha.t_{high}[j - 1]$  then
49:                      remove  $prev_{tval}$  from  $node.prev.tlist$ 
50:                  end if
51:              end if
52:          end for
53:      end if
54:  end for
55: end if
56: if  $accepted = true$  and  $node.index = N$  then
57:     update  $\alpha.freq = \alpha.freq + 1$ 
58:     set  $temp = node$ 
59:     while  $temp! = \phi$  do
60:         update  $temp.visited = false$ 
61:         if  $temp.index! = 1$  then
62:             remove  $temp$  from  $waits(\alpha[temp.index])$ 
63:         end if
64:         update  $temp = temp.next$ 
65:     end while
66: end if
67: end for
68: 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 $F = \{\alpha_1, \dots, \alpha_J\}$; symbol set ε ; threshold γ

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

- 1: calculate the number of episodes J , and number of days K
- 2: calculate all η s' 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 }
- 3: **for** $0 \leq j \leq J$ **do**
- 4: $\theta[j] = 1/J$
- 5: $episodeFreq[j] = \sum_i^K e_{ij}$
- 6: **end for**
- 7: select those frequent episodes starting with 'S' and ending with 'Z' and separate these episodes by workday or holiday
 - { calculate eta for each episode }
- 8: **for** $0 \leq j \leq J$ **do**
- 9: $\eta[j] = 1 - episodeLen[j] * episodeLen/T$
- 10: **end for**
 - { likelihood prediction of each episode j in the k th day }
- 11: **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**
- 15: **end for**
 - { calculate the obj value based on J , K , $likelihood_{ij}$ and θ }
- 16: **while** $newObj - obj > \gamma$ **do**
 - 17: $\theta_{new} = []$
 - 18: **for** $0 \leq l \leq J$ **do**
 - 19: $temp = 0$
 - 20: **for** $0 \leq j \leq K$ **do**
 - 21: $temp = temp + \frac{\theta_l * likelihood_{il}}{\sum_0^J \theta_j * likelihood_{ij}}$
 - 22: **end for**
 - 23: $\theta_{new}[l] = temp/K$
 - 24: **end for**
 - 25: calculate the $newObj$
 - 26: **if** $newObj - obj > \gamma$ **then**
 - 27: $obj = newObj$
 - 28: $\theta_{new} = \theta$
 - 29: **end if**
 - 30: **end while**
 - 31: Output $\Lambda_Z = \{(\Lambda_{\alpha_j}, \theta_j), j = 1, \dots, J\}$

Chapter 6

Conclusion

Due to urbanization, various aspects of people's lives, such as traffic flow, power consumption, air quality, and social network, change rapidly. In recent years, policy-makers and scientists have realized the urgent need for a transition to an environmentally friendly and sustainable city, and have proposed *urban computing* research. One of the critical components of intelligent cities is smart building research. In particular, energy distribution and consumption are important topics because of their two-fold impact on society. First, power is an essential resource which has a great impact on our economy, so effective consumption is important. Secondly, the large impact the energy industry has on the environment, such as climate change, makes it an extremely sensitive and relevant topic of research.

The central theme of this dissertation is using data analytics methods to tackle major issues in smart building. We provide a general framework and propose analytical solutions within that framework. The data mining techniques we use for discovering knowledge from the datasets are temporal mining and probabilistic models. We classify smart building events into two categories: device-related profiles and human-related events. This dissertation provides strategies to handle these two challenging energy-related problems.

- **Characterization of electrical devices by energy disaggregation:** We studied energy consumption inside a building with a non-intrusive approach, extracted the hidden patterns, and disclosed the correlation among these patterns. Furthermore, we connected these patterns with devices to infer the electricity usage of individual devices. We demonstrate that with the use of frequent-episode mining in conjunction with temporal mining techniques, we can effectively glean insights into usage patterns of electrical devices. Our approach describes a novel motif discovery approach that utilizes on/off events to unravel the operation frequency and duration of devices. We show that the our approach is very adroit at discerning multiple power levels and at effectively untwining the combinatorial operation of the devices. Moreover, we also show that this approach is not just an aid to disaggregation but, as a byproduct, also extracts temporal episodic relationships that shed insight into

consumption patterns.

- **Characterization of water use ends by water disaggregation:** We characterized the water usage patterns inside a building, and inferred underlying water use ends. Additionally, we associated these patterns according to time to derive the activities of the people inside. To improve upon our initial work, we propose a semi-supervised recursive multivariate piecewise motif mining approach for both energy and water disaggregation. Since the algorithm operates in two phases and effectively filters out appliances that have a large power consumption in the first stage, it can effectively discover usage patterns of smaller appliances. This insight provided by our approach allows for more precise energy disaggregation. Moreover, this approach can be effectively utilized to identify continuously variable loads, like outdoor heating. Similarly, by regarding hot and cold water as two phases, we can separate many water use ends.
- **Characterization of activities of daily life by occupancy prediction:** We investigated the activities patterns of people inside a residential building. Then we connected these patterns with a graphical model. Using the graphical model, we inferred whether the room/house was occupied. We demonstrate that by integrating the mixture EGH model and kNN together as a hybrid approach, we get the best prediction result. In our work, 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. 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 to be within a proper range.

This research demonstrates that data analysis methods are able to solve smart building challenges. The proposed heuristic approaches utilize the characteristics of two typical profiles in smart buildings, thereby fundamentally paving the way to advanced controlling and scheduling systems for devices inside the buildings.

Future Work

This dissertation has opened up many opportunities for future work. From a theoretical perspective, the probabilistic model of piecewise events has great research potential. We can try to connect frequent piecewise episodes with a generalized probabilistic model, as in [81]. Furthermore, we can associate the dynamic time-warping models used in water disaggregation with similar probabilistic models.

While we have made significant headway in energy disaggregation, there is significant room for improvement. One of the immediate extensions is to incorporate more features (in the multi-phase aggregate) when a device turns on. The sudden spikes in the aggregated data, when normalized, can indicate the *startup shape*, which can be correlated to a device, and hence improve the overall effectiveness of device prediction, thus leading to more accurate energy disaggregation. Moreover, we can significantly improve the temporal mining approach to disaggregate more devices. Furthermore, our disaggregation algorithms can be explored for water disaggregation as well. We will

exploit our temporal mining algorithms, integrated with dynamic time warping and motif mining, to propose an algorithm to effectively conduct home-level disaggregation. We can also extend our disaggregation algorithms for bandwidth distribution for internet service providers. Using our disaggregation algorithms, we can decipher the device-level internet usage and plan for effective distribution to a home/neighborhood.

Solving energy disaggregation is an important practical problem, which, when resolved, results in a highly monetizable insight: consumption patterns. As a result, policy makers and energy distributors can design packages for consumers based on their needs. This will enable both consumers and distributors to effectively and *smartly* buy and sell power that is highly customized to the needs of a home.

The occupancy prediction work lends itself to future extensions via hybrid approaches. We can integrate kNN and a mixture of EGH, which has the best performance on the sensor data set. One of the future directions is to incorporate GPS-based information to track movements of the house residents. This has the potential to be an excellent surrogate to automated power control of devices in a home. Another interesting problem to tackle is holiday occupancy prediction. The occupancy patterns for these days are completely different. For example, on certain weekdays, a person may never go out. Therefore the occupancy prediction probably depend more on date than on other indoor activities.

One of the critical problems to ensure that we can reduce high energy consumption by automatically controlling the temperature regulation systems. Moreover, with the advent of more mainstream technologies like *Nest* and other intelligent automatic control systems that remotely control the thermostat, occupancy prediction is a crucial parameter in determining the settings. Furthermore, comfort levels based on individuals can determine the temperature at which different parts of the residence should be set. Thus optimizing energy usage and maximizing user comfort are very important and immediate problems to solve.

Research in the domain of energy consumption has a two-fold impact on the society we live in. The rapid urbanization of our society requires the *smart* and optimized distribution of power to meet the demands of the city. Logistical problems in power distribution, particularly meeting the high-volume requirement with minimal failure, are important problems. Moreover, the larger problem here is to ensure that we leave a small carbon foot print. Even though renewable energy resources are being tapped, fossil fuels are still the fundamental source of energy in our society. The optimized distribution of power and a well-established balance between supply and demand can ensure that we use only just as much of the fossil fuels as we need. As responsible researchers, it is important that we turn our talents towards this overarching goal of saving our planet and doing every thing we can to ensure that it is in good shape when we hand it over to our next generation.

Bibliography

- [1] Buildings energy data book. *Energy Efficiency and Renewable Energy*, 2011.
- [2] Buildings energy data book. *Energy Efficiency and Renewable Energy*, 2014.
- [3] K. Abed-Meraim, W. Qiu, and Y. Hua. Blind system identification. *Proceedings of the IEEE*, 1997.
- [4] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, 1995.
- [5] M. Akbar and D. Khan. Modified nonintrusive appliance load monitoring for nonlinear devices. In *Proceedings of the IEEE International Conference on Multitopic*, 2007.
- [6] M. H. Albadi and E. El-Saadany. Demand response in electricity markets: An overview. In *2007 IEEE power engineering society general meeting*, 2007.
- [7] AlphaLab. Emf meter.
- [8] 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.
- [9] Amprobe. Temperature meter.
- [10] 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 Conference(IECON)*, 2012.
- [11] 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.
- [12] M. Baranski and J. Voss. Nonintrusive appliance load monitoring based on an optical sensor. In *Proceedings of the IEEE Power Tech Conference*, 2003.

- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] M. Berges, E. Goldman, H. Matthews, and L. Soibelman. Enhancing electricity audits in residential buildings with nonintrusive load monitoring. *Journal of Industrial Ecology*, 2010.
- [20] M. Berges, E. Goldman, L. Soibelman, and K. Anderson. User-centric non-intrusive electricity load monitoring for residential buildings. *Journal of Industrial Ecology*, 2010.
- [21] C. Bishop and N. Nasrabadi. *Pattern recognition and machine learning*. 2006.
- [22] Blekin. 2013.
- [23] T. Blumensath and M. Davies. Shift-invariant sparse coding for single channel blind source separation. *SPARS*, 2005.
- [24] D. Carboni, A. Gluhak, J. A. McCann, and T. H. Beach. Contextualising water use in residential settings: A survey of non-intrusive techniques and approaches. *Sensors*, 16(5):738, 2016.
- [25] 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.

- [26] 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.
- [27] 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.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] 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.
- [33] C. Chow and J. N. Tsitsiklis. The complexity of dynamic programming. *Journal of Complexity*, 1989.
- [34] 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.
- [35] 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.
- [36] D. J. Cook, A. S. Crandall, B. L. Thomas, and N. Krishnan. Casas: a smart home in a box. *Computer*, 2013.
- [37] 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.

- [38] 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.
- [39] M. E. Davies and C. J. James. Source separation using single channel ica. *Signal Processing*, 2007.
- [40] 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.
- [41] 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.
- [42] S. Drenker and A. Kader. Nonintrusive monitoring of electric loads. *IEEE Computer Applications in Power*, 1999.
- [43] 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.
- [44] EnergyCircle. emonitor sensor.
- [45] L. Engineering. Mindstorms.
- [46] 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.
- [47] 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.
- [48] M. Figueiredo, B. Ribeiro, and A. de Almeida. Electrical signal source separation via non-negative tensor factorization using on site measurements in a smart home. *IEEE Transactions on Instrumentation and Measurement*, 2014.
- [49] 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.
- [50] 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.
- [51] H. Gonçalves, A. Ocneanu, and M. Bergés. Unsupervised disaggregation of appliances using aggregated consumption data. 2011.

- [52] D. Görür and C. Rasmussen. Dirichlet process Gaussian mixture models: choice of the base distribution. *Journal of Computer Science and Technology*, 2010.
- [53] 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.
- [54] A. Hambley. *Electrical engineering principles and applications*. 2006.
- [55] G. Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 1992.
- [56] 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.
- [57] D. D. Hatley, R. J. Meador, S. Katipamula, M. R. Brambley, and C. Wouden. Energy management and control system: Desired capabilities and functionality. *Pacific Northwest National Laboratory, Richland, WA*, 2005.
- [58] 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.
- [59] E. Instruments. Light sensor.
- [60] N. Instruments. Ni-9239.
- [61] A. Jain, M. Murty, and P. Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 1999.
- [62] 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.
- [63] 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.
- [64] M. Johnson and A.S. Bayesian nonparametric hidden semi-markov models. *CoRR*, 2012.
- [65] J. Kelly and W. Knottenbelt. Disaggregating multi-state appliances from smart meter data. *SIGMETRICS*, 2012.
- [66] J. Kelly and W. Knottenbelt. Metadata for energy disaggregation. 2014.
- [67] 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.

- [68] 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.
- [69] N. Kiukkonen, J. Blom, O. Dousse, D. Gatica-Perez, and J. Laurila. Towards rich mobile phone datasets: Lausanne data collection campaign. *Proc. ICPS, Berlin*, 2010.
- [70] W. Kleiminger, F. Mattern, and S. Santini. Predicting household occupancy for smart heating control: A comparative performance analysis of state-of-the-art approaches. *Energy and Buildings*, 2014.
- [71] 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.
- [72] T. Kolda and B. Bader. Tensor decompositions and applications. *SIAM review*, 2009.
- [73] J. Kolter, S. Batra, and A. Ng. Energy disaggregation via discriminative sparse coding. In *Proceedings of Neural Information Processing Systems*, 2010.
- [74] 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.
- [75] 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.
- [76] J. Krumm and A. B. Brush. Learning time-based presence probabilities. In *Pervasive Computing*. 2011.
- [77] H. Kuhns, M. Roberts, and B. Bastami. Closure rules for energy load disaggregation. Pittsburgh, U.S.A., 2012.
- [78] P. Lai, M. Trayer, S. Ramakrishna, and Y. Li. Database establishment for machine learning in nilm. Pittsburgh, U.S.A., 2012.
- [79] 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.
- [80] C. Laughman, K. Lee, R. Cox, S. Shaw, S. Leeb, L. Norford, and P. Armstrong. Power signature analysis. *IEEE Power and Energy Magazine*, 2003.
- [81] 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.

- [82] S. Laxman and P. S. Sastry. A survey of temporal data mining. *Sadhana*, 31(2):173–198, 2006.
- [83] 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.
- [84] 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.
- [85] K. Lee. *Electric load information system based on non-intrusive power monitoring*. PhD thesis, Massachusetts Institute of Technology, 2003.
- [86] 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.
- [87] 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.
- [88] 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.
- [89] 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.
- [90] M. Lewicki and T. Sejnowski. Learning overcomplete representations. *Neural computation*, 2000.
- [91] Y. Li and S. Osher. Coordinate descent optimization for l1 minimization with application to compressed sensing; a greedy algorithm. *Inverse Problem Imaging*, 2009.
- [92] 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.
- [93] 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.
- [94] D. Luo, L. Norford, S. Leeb, and S. Shaw. Monitoring hvac equipment electrical loads from a centralized location- methods and field test results. 2002.
- [95] S. Mahmoud, A. Lotfi, and C. Langensiepen. Behavioural pattern identification and prediction in intelligent environments. *Applied Soft Computing*, 2013.

- [96] 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.
- [97] G. Mambayashi. Noise measurements of the residential power line. In *Proceedings of the IEEE International Symposium on Power Line Communications and Its Applications*, 1997.
- [98] 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.
- [99] H. Mannila, H. Toivonen, and A. Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1997.
- [100] 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.
- [101] 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.
- [102] 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.
- [103] D. Minnen, T. Starner, I. Essa, and C. Isbell Jr. Improving activity discovery with automatic neighborhood estimation. In *IJCAI*, volume 7, pages 2814–2819, 2007.
- [104] 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.
- [105] 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.
- [106] 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.
- [107] K. Nguyen, H. Zhang, and R. Stewart. Development of an intelligent model to categorise residential water end use events. *Journal of Hydro-Environment Research*, 7(3):182–201, 2013.
- [108] 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.

- [109] 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.
- [110] A. Pardo, V. Meneu, and E. Valor. Temperature and seasonality influences on spanish electricity load. *Energy Economics*, 2002.
- [111] 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.
- [112] 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.
- [113] D. Patnaik, M. Marwah, R. Sharma, and N. Ramakrishnan. Sustainable operation and management of data center chillers using temporal data mining. In *Proceedings of the 15th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining*, 2009.
- [114] D. Patnaik, P. Sastry, and K. Unnikrishnan. Inferring neuronal network connectivity from spike data: A temporal data mining approach. *Scientific Programming*, 2008.
- [115] 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.
- [116] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 262–270. ACM, 2012.
- [117] 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.
- [118] 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.
- [119] 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.
- [120] 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.
- [121] K. C. Seto, M. Fragkias, B. Güneralp, and M. K. Reilly. A meta-analysis of global urban land expansion. *PloS one*, 6(8):e23777, 2011.

- [122] 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.
- [123] 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.
- [124] S. Shaw. *System identification techniques and modeling for nonintrusive load diagnostics*. PhD thesis, 2000.
- [125] S. Shaw, S. Leeb, L. Norford, and R. Cox. Nonintrusive load monitoring and diagnostics in power systems. *IEEE Transactions on Instrumentation and Measurement*, 2008.
- [126] 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.
- [127] D. Srinivasan, W. Ng, and A. Liew. Neural-network-based signature recognition for harmonic source identification. *IEEE Transactions on Power Delivery*, 2006.
- [128] 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.
- [129] 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.
- [130] F. Sultanem. Using appliance signatures for monitoring residential loads at meter panel level. *IEEE Transactions on Power Delivery*, 1991.
- [131] 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.
- [132] Y. Tanaka, K. Iwamoto, and K. Uehara. Discovery of time-series motif from multi-dimensional data based on mdl principle. *Machine Learning*, 58(2-3):269–300, 2005.
- [133] P. Technology. Ta041.
- [134] TED. Ct products, 2010.
- [135] TrendPoint. Power meter.
- [136] 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.

- [137] 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.
- [138] R. Vullers, R. Schaijk, H. Visser, J. Penders, and C. Hoof. Energy harvesting for autonomous wireless sensor networks. *IEEE Solid-State Circuits Magazine*, 2(2):29–38, 2010.
- [139] B. Wang, H. Dong, A. Boedihardjo, and C. Lu. A hierarchical probabilistic model for low sample rate home-use energy disaggregation. 2013.
- [140] W. Wichakool, A. Avestruz, R. Cox, and S. Leeb. Modeling and estimating current harmonics of variable electronic loads. *IEEE Transactions on Power Electronics*, 2009.
- [141] 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.
- [142] M. Wytock and J. Kolter. Contextually supervised source separation with application to energy disaggregation. 2014.
- [143] 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.
- [144] 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.
- [145] M. Zeifman. Disaggregation of home energy display data using probabilistic approach. *IEEE Transactions on Consumer Electronics*, 2012.
- [146] M. Zeifman and K. Roth. Nonintrusive appliance load monitoring: review and outlook. *IEEE Transactions on Consumer Electronics*, 2011.
- [147] 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.
- [148] 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.
- [149] A. Zoha, A. Gluhak, M. Imran, and S. Rajasegarar. Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. *Sensors*, 2012.