Low-complexity Low-rate Residential Non-Intrusive Appliance Load Monitoring

Hana Altrabalsi

Electronic & Electrical Engineering

University of Strathclyde

A thesis submitted for the degree of

Doctor of Philosophy

Glasgow 2017

This thesis is dedicated to
my late parents
Fathi and Zahra

and my beloved daughters
Salsabil and Sulaf

Acknowledgements

First and foremost, I would like to express my sincere thanks to my supervisor, Dr.Vladimir Stankovic, for his excellent advice and constant support. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my PhD study.

Second, I would like to thank J. Kolter and M. Johnson for making the REDD database available. A. Monacchi for making the GREEND database available and D. Murray for making the REFIT database available. I would also thank O. Parson for making his HMM-based-method code available and J. Liao for helping to run Chapter 4 HMM-based-method results.

Third, I would also like to give my thanks to my colleagues both past and present for their companionship and coffee stories, especially Georgia Elafoudi and Jing Liao, because of you, I have enjoyed every minute of my PhD. I would like also to thank all my amazing friends, especially; Suad Al Zeheri from Iraq and Dabiah Alboaneen from Saudi Arabia, you have been very close friends in all times.

Fourth, my sincere thanks goes to my family which has been encouraging, supportive and shown belief in me and my work. Sadly, my both parents passed away before finishing my PhD, however, I am sure they are proud of me. Many thanks to my brothers; Osama, Walid, Emad, Amro and Nizar, for their constant support throwout my life and my study. I would like to thank my lovely sisters-in-law; Anna-Bella and Ozlem, my niece Anna-Gabriella-Nour and my nephews; Fathi and Ulash, for mastering Skype calls and sharing photos with me.

Fifth, huge thanks, love and appreciation go to my dear husband, Moftah. Studying for a PhD can be such an exciting journey and I am glad we got to enjoy it together, although, juggling time between your study and my study with our kids in between have been very challenging.

Finally, I would like to acknowledge the financial support provided by the Libyan embassy in London, without their scholarship, this work would not have been possible.

Abstract

Large-scale smart metering deployments and energy saving targets across the world have ignited renewed interest in residential non-intrusive appliance load monitoring (NALM), that is, disaggregating total household's energy consumption down to individual appliances, using purely analytical tools. Despite increased research efforts, NALM techniques that can disaggregate power loads at low sampling rates are still not accurate and/or practical enough, requiring substantial customer input and long training periods. In this thesis, we address these challenges via a practical low-complexity low-rate NALM, by proposing two approaches based on a combination of the following machine learning techniques: k-means clustering and Support Vector Machine, exploiting their strengths and addressing their individual weaknesses. The first proposed supervised approach is a low-complexity method that requires very short training period and is robust to labelling errors. The second, unsupervised approach relies on a database of appliance signatures that we designed using publicly available datasets. The database compactly represents over 100 appliances using statistical modelling of measured active power. Experimental results on three datasets from US (REDD), Italy and Austria (GREEND) and UK (REFIT), demonstrate the reliability and practicality of the proposed approaches.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Energy demand has dramatically increased in recent years especially in urban areas. Currently, buildings in Europe alone are responsible for 40 % of energy consumption and 36 % of CO2 emission [1]. The European Union officially adopted a plan of 20% improved energy efficiency by 2020 [1]. To meet that goal, an annual reduction of 1.5 % in national energy sales must be made, which should be facilitated by a rollout of close to 200 million *smart energy meters* [1] [2].

The U.S. Energy Information Administration (EIA) published results [3] with an international energy projection for the period of 2012 to 2040. The outlook included residential energy consumption comprising energy used for heating, cooling, lighting, water heating and consumer products. According to this report, Mexico and Chile combined energy demand grow by an average of 1.9% a year from 2012 to 2024, while electricity becomes the major residential energy source, increased from roughly 37% of their total energy use in 2012 up to 60% in 2040. In Europe, many countries have relatively low economic growth, therefore, energy consumption estimations were slightly lower than earlier years. However, electricity remains the fastest-growing source of residential energy. In Asia, Japan's energy consumption steady declines, but China and India together will hold a share of 27% of the world's total residential energy consumption in 2040. Indeed, the rapid increase in China's growth leads to an increase in energy consumption of 2.4% a year in average from 2012 to 2040. Middle East holds a share of 6.6 % of the total residential energy consumption in 2012, but it is estimated to grow by an average of 1.7 % a year, which is far higher than the world's average of 0.9% a year from 2012 to 2040 due to the increasing demand for cooling [3].

Renewable energy and nuclear power are the world's fastest-growing energy sources, each increasing with a pace of about 2.5 % per year. However, fossil fuels will continue to supply nearly 80 % of the world's energy use through 2040 [3]; they are also expected to end in the near future causing an energy crises.

Load monitoring and management are urgently needed in order to lower the unnecessary energy consumption [4] [5]. This could also encourage appliance manufacturers to upgrade traditional household appliances to energy efficient ones [6]. A possible solution to lower the increasing demand of power usage is to implement smart energy grid using feedback schemes based on smart meters [7]-[9].

A large scale deployment of smart meters in households has started or is about to start in many countries worldwide. For example, the UK Government has committed utilities to a roll out of automatic meter reading (AMR) systems by 2020. It is anticipated that by 2020 all UK households will be equipped with an AMR system that measures and displays in real time aggregate energy usage with an in-home display unit [11]. In Europe, Italy has the largest smart meter deployment but other countries are following its lead. For example, France plans to deploy about 35 million smart meters by 2020 and Spain plans to deploy 13 million smart meters by 2018 [12].

Australia launched a smart metering project in Victoria state at the end of 2013 aiming to install 2.75 million residental smart meters [13]. The U.S Government agreed a roll out of about 58.5 million smart meters in 2014, about 88 % were residential installations [3].

The Middle East smart meter market is forecast to reach 16.1 million units by 2022 with roughly 86 % of homes and businesses in the region expected to have smart meters installed [14]. For example, the Qatar's first phase of smart meter instillation is currently under deployment in Doha area and one million smart meters are to be deployed across the Emirates by 2020 [15]. Other countries in the region are following the lead of Qatar and Emirates, like Lebanon, which announced in 2014 to install 1.2 million units and Iran, which launched a limited smart meter project for a trial that was announced in April 2016, with 33 million meters to be installed over a period of 10 years [14].

Smart meters or *load monitors* installed in households could possibly provide energy information feedback, which could help consumers to learn about their energy consumption habits. Timing of energy consumption and dynamic pricing plan brought to consumers could result in persuading them to change their energy consumption routines toward off-peak period for lower billing, which could hopefully reduce energy peak demand. By managing the peak demand, we could prevent power outage from occurring on the grid during high power demand usage periods. However, electricity flows will be managed in a better way by utilities during peak periods and navigate effectively through system emergencies by using coordinated and networked smart meters [7], [10].

This global investment promises significant improvements in energy demand via automatic, more efficient and more informed billing. However, to provide a richer energy feedback, information about consumption of individual appliances is necessary. Indeed, up to 20% of reduction in energy consumption is expected via appliance-feedback and specific appliance replacement programs [16].

1.2 Intrusive vs. Non-intrusive Load Monitoring

Load monitoring is made by inserting a monitoring device between the socket and the appliance and then recording its operation. This method is generally called *"intrusive"* load monitoring, and up to date, is still considered inconvenient for large scale implementations as it requires a recording devise for each appliance of interest in order to monitor their power consumption or other load activities [17].

Non-Intrusive Appliance Load Monitoring (NALM), also referred to as NILM or NIALM [19], presents an attractive alternative, because it performs disaggregation at the metering point and does not require any measuring devises installed to individual appliances [17, 18].

With today's technology and increasing deployment of smart meters, monitoring individual appliances using individual sensors in a house is often impractical and expensive, especially having in mind that the number of electrical devices at home are rapidly increasing. On the other hand, energy disaggregation via NALM offers separating total energy readings obtained from a single energy meter into power loads of each appliance used, based purely on computational approach.

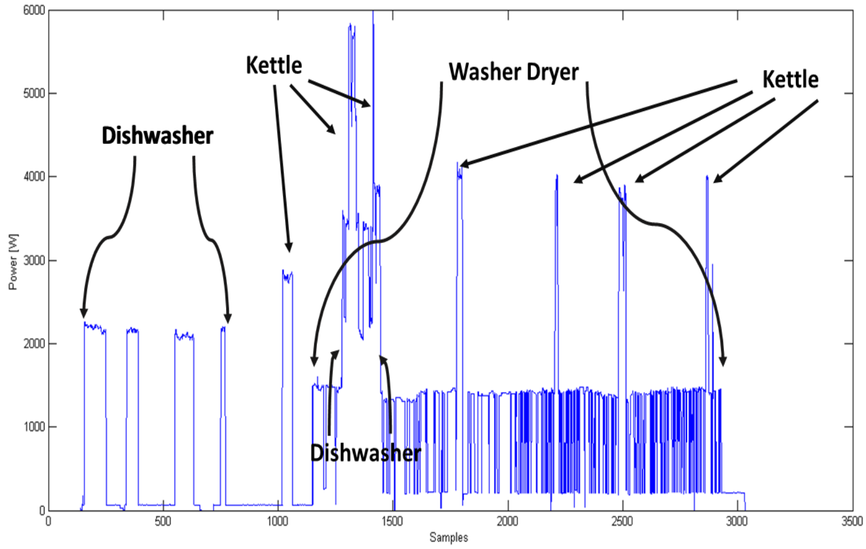


Figure 1.1: An aggregated load data obtained using single point measurement from REFIT dataset House 9 based on Hart's method.

1.3 Early Work on NALM

NALM was introduced in the research literature in 1982's by G. W. Hart [19] [20]. Later research in the area of NALM showed that NALM implementation could be very useful to consumers with installed smart meters [21]-[24].

Hart's method is based on disaggregating electrical loads by examining only the appliance specific power consumption signatures within the aggregated load data (total load) as shown in Figure 1.1 [25]. Hart assumed that each load in a house consumes a unique amount of real power and reactive power. Therefore, his work used real power (P) and reactive power (Q) to detect variations of loads (Δ) in the ΔP-ΔQ plane as shown in Figure 1.2 using the steady state variations. ON and OFF status of a certain load, based on Hart's method, are determined by observing the change in real and reactive power [22],[26]-[35]. However, results in Figure 1.2 show some overlapped clusters because there are some appliances with similar real and reactive consumptions, and in these cases, more signatures are required to identify these loads with more unique characteristics [25].

Since Hart's work, many NALM algorithms have been proposed that improve the initial design of [19] and adapt to advances in sensor technology, capturing energy

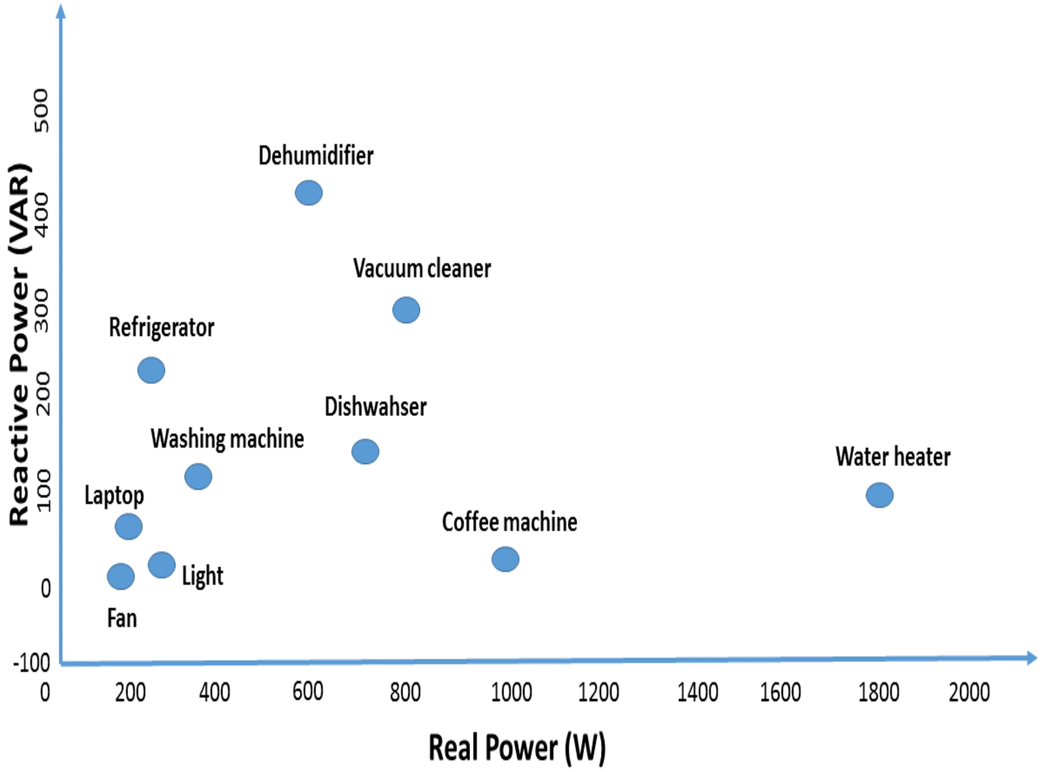


Figure 1.2: The Loads On Real and Reactive Power Plane based on Hart's method [25].

measured at a range of sampling rates, generally in the order of kHz. However, with large-scale smart metering deployments on the way, an increased interest in NALM algorithms that work at lower sampling rates, in the order of seconds and minutes made NALM a hot research topic [25]. It is not only the cost of the sensing technology [16], but also computational and storage cost as well as implementation efficiency that are key drivers towards the wide deployment of low-sampling smart meters. However, so far, there are no widely available efficient solutions for NALM, that offer high accuracy and low complexity smart metering [37, 25].

1.4 Problem Statement

The goal of this research work is to develop load disaggregation methods at the metering point and explore potential applications. By doing so, we could add new services to customers. The objective of this thesis is to propose and test reliable and low-complexity disaggregation solutions.

The approaches discussed here require that the aggregate power drawn by all appliances in a household is measured periodically. We use three available datasets of active power measurements, REDD from United States (US) [39], GREEND from Austria and Italy [40] and REFIT from United Kingdom (UK) [41] to simulate daily appliance activities in a household. For example, Figure 1.3 shows few-hour power usage for seven known appliances from the REFIT dataset House 17 [41]. The concern, as illustrated in Figure 1.4, is how to effectively separate these individual appliances in real-time using only total active power.

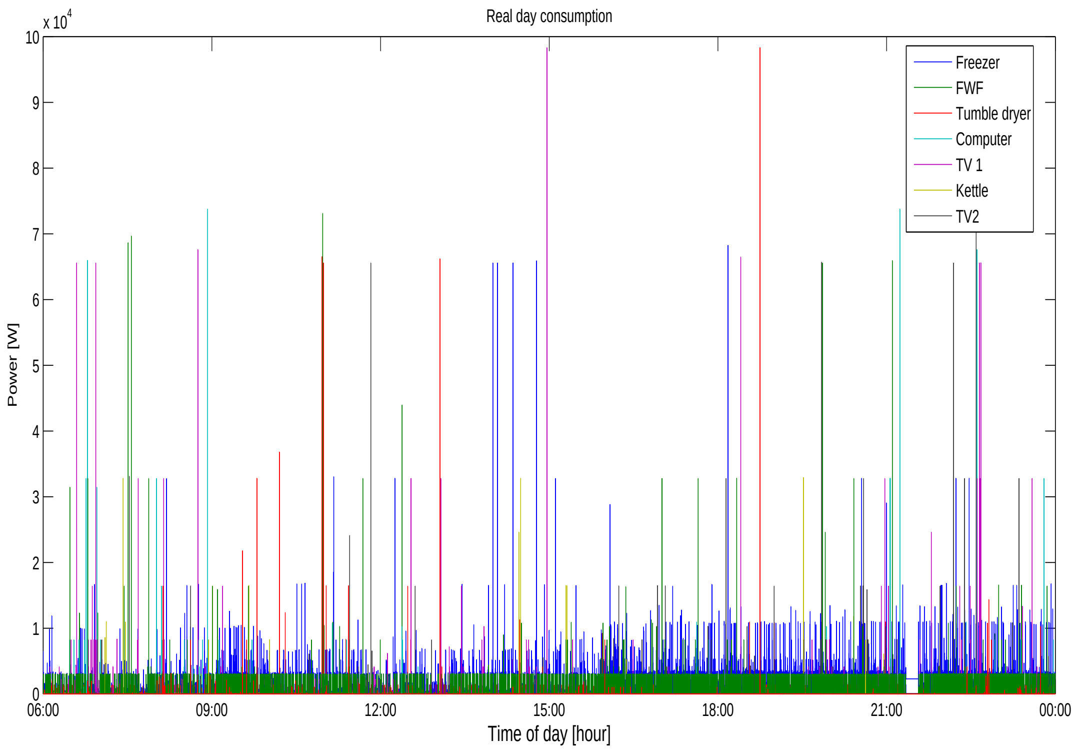


Figure 1.3: Daily energy consumption from REFIT dataset House 17. FWF stands for fridge with freezer.

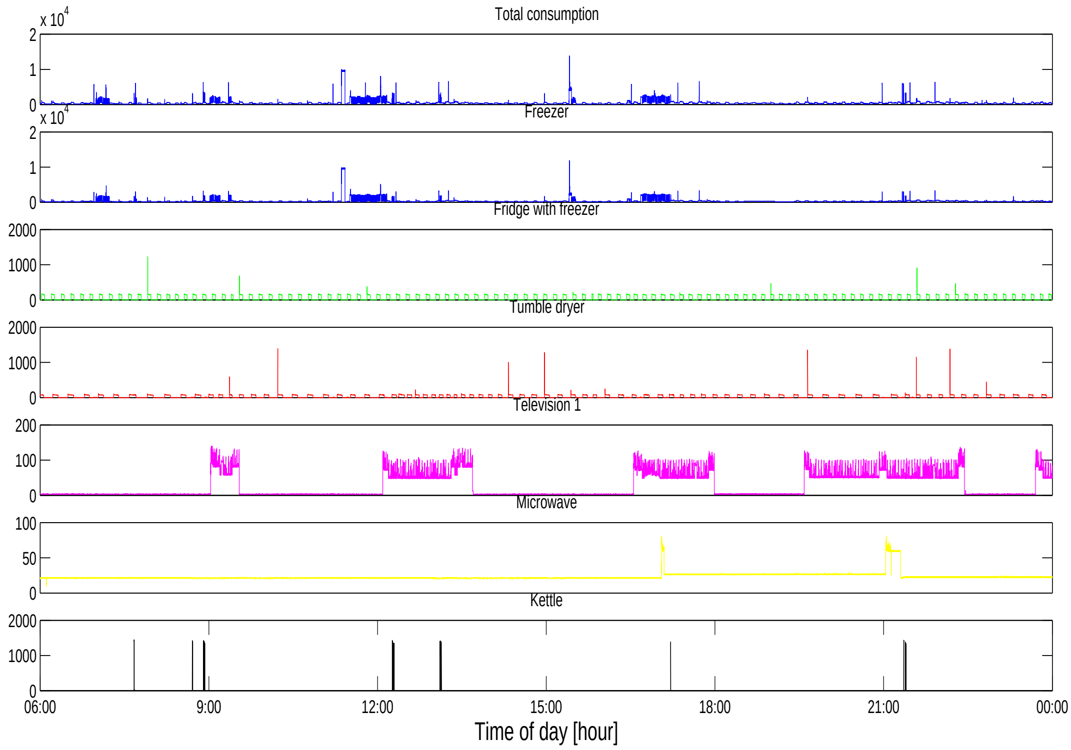


Figure 1.4: Total and individual appliance energy consumption from REFIT dataset House 17. y axes denotes energy consumption in Watt [W].

1.5 Novel Contributions

The key issues with current low-sampling rate NALM methods are high complexity, low accuracy and a requirement for impractical training which means intensive amount of training data which could slow the system response time. To overcome these problems, we propose an efficient low-complexity *supervised* NALM approach that combines k-means and Support Vector Machine (SVM) [42]. In particular, to benefit from high classification performance of linear and non-linear SVM and low computational cost of Trained k-means clustering, we effectively combine conventional k-means and SVM obtaining a hybrid method that outperforms k-means and SVM classification alone. We use k-means to cleverly select a subset of input data used to train an SVM. By training the SVM only on a small set of representative samples, we are able to significantly reduce computational cost.

To make this supervised approach practical and reduce or remove the need for a labeled training dataset, we build a database of appliance signatures by acquiring appliance power load measurements from houses as well as processing publicly available datasets. Such a database is then used to develop an unsupervised approach that requires no training (and hence no client's input). The database is a compact collection of appliance power load statistical features that is used for energy disaggregation in unknown houses. It is populated using open source datasets from US [39], Austria and Italy [40], and UK [41]. Some similar attempts are recently reported in [120] but for US houses only and high data-rates.

The main contributions of this thesis are:

- Novel real-time low-complexity Linear and Gaussian k-means and SVM-based NALM method for low sampling rate data;
- Innovative appliance-based selection of extracted features that maximize performance;
- Detailed experimental evaluation using different training sizes and errors in labeling the training data;
- Comparison with Trained k-means and Linear and Gaussian SVM in terms of time complicity and accuracy ;
- A generic database of appliance load profiles populated from 34 houses in UK, Europe, and US, containing over 200 appliance signatures.
- A low-complexity NALM approach that uses the developed database for training, irrespective of the house, and hence does not require customer input; using House-agnostic training data and compared with House-specific training data approach.
- Comparison with Linear and Gaussian SVM approaches using three households from the REDD data set [39], GREEND data set[40] and REFIT data set [41].
- Comparison with state-of-the-art Hidden Markov Model [43].

1.6 Thesis Layout

The rest of this thesis is organized as follows: Chapter 2 brings a brief background on NALM and available methodologies. Chapter 3 describes the proposed methodology of data processing and classification algorithms with an illustration of the used NALM algorithms. Chapter 4 studies the proposed algorithms and tests their performance and robustness as a supervised approach. Chapter 5 introduces a database of load-profile signatures as well as a deep investigation of our signature database with two methods to cluster appliance signatures into groups and sub-groups based on their Gaussian signature similarities. Chapter 6 introduces two novel unsupervised methods and their performance in comparison with the supervised approach. The last chapter discusses our main findings, limitations of our frameworks and future work.

Chapter 2

Background and Related Work

2.1 Introduction

NALM is a challenging task, but has the benefit of helping home owners and occupants conserve energy, which can be established by knowing how appliances within the home are used and how much energy they consume.

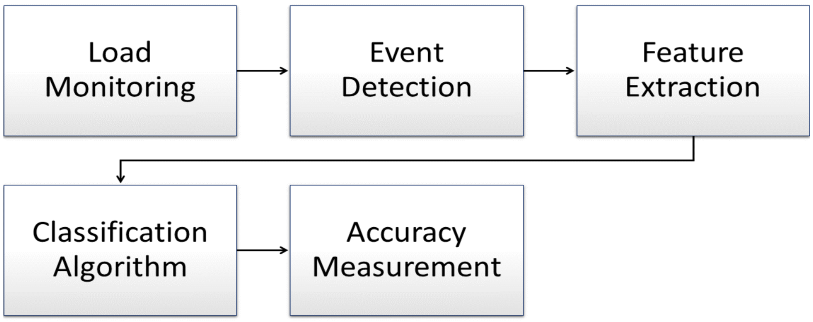


Figure 2.1: Flow chart for a typical classification method.

Figure 2.1 shows a block diagram of a typical NALM algorithm [44]. This chapter is organized in much the same manner, with each section discussing details of each of the blocks in the diagram as in [46], [47], [43] and [25].

Existing NALM approaches can be roughly categorized into event-based and state-based depending on whether or not they rely on detecting and classifying each appliance-state transition [45]. Event-based NALM systems use a filter or a pre-set thresholds to detect when a power signal changes its state. Traditional event-based NALM methods [19] consist of signal pre-processing, edge detection and feature extraction followed by classification. After acquisition, signal pre-processing can be established in the form of power normalization, filtering (for signal smoothing and getting rid of sudden peaks), and set a threshold values to remove small power loads that would appear as noise as well as the base-power load, from appliances that are always running. Next, edge detection is performed to identify events of appliances switching on, off or changed state. Edge detection is followed by extracting features in an identified event window, where a window of events is created between every detected events. Classification is then used to group sets of extracted windows with have similar characteristics, such as power levels, time profile, reactive components etc.

2.2 Measurement Acquisition

Any NALM method starts with collecting power meter reading from sensors placed on a power line/lines that needed to be monitored. In this section, we discuss three important issues that are considered when installing a power meter in a household:

- Measurement Type: which indicates the type of power load we are monitoring such as real power, reactive power, current waves, etc.
- Sample Rate: which indicates how many samples per second, minute or hour we are storing.
- Sensing Type: which indicates how many meters are installed in a single household.

2.2.1 Measurement Types

Many researchers used current and voltage and related loads to disaggregate appliances from a single point reading. Figueiredo et al. [48] [49] used voltage, current, and power factor measurements stating that a simple power load type can provide a high accuracy results with steady-state events matching. Gupta et al. [50] obtained interesting results using more advanced load measurements, Electromagnetic Interference (EMI) spectrum analysis, where similar appliances showed similar signatures but the EMI sensor placed in a home affected the readings from appliances and contributed to noise. However, the authors were still able to detect near simultaneous appliance events at 102 milliseconds apart.

Likewise, Berenguer et al. [51] found that tested appliances show some unique start up signatures using current short impulses when they were turned on. Tsai and Lin [52] tested five different appliances: fan, florescent light, radio, and microwave oven. They used current characteristic measurements such intensity, peak and average values in current pulses. They captured current waveforms as the appliances were being turned on at different voltage phase angles creating different appliances load profiles. However, they had to down-sample their measurements from 1 microsecond to 500 microseconds for memory requirements.

Inspired by Hart [19], Norford and Leeb [53] collected commercial buildings active and reactive power readings focusing on transient waves. They concluded that active power was more useful than reactive power in their transient detector. They also noted that transient signatures helped detecting appliances start up but did not detect appliances that were running. Laughman et al.[54] modified Hart's approach by adding the third dimension to the $\Delta P - \Delta Q$ plan which was harmonics. They noted that adding transient harmonics could lead to a better distinguish in appliances that were overlapped using ΔP-ΔQ plane only.

Fisera and Macek [55] also used active and reactive power in conjunction with the existing building management system. They noted that using control signals caused a higher computational cost, especially when more appliances were introduced, however, they managed to detect appliances in near real-time disaggregation.

Researchers like [Kim et al. [56], Berges et al. [44], Zeifman [57], Kolter et al. [58], Kolter and Jaakkola [59], Parson et al. [43]] used real power measurements. Kim et al. [56] found it difficult to disaggregate steady-states and had to use appliance changing states. They also concluded that more appliances considered means less accuracy, from one appliance (100 %) to eight appliances with accuracy drop (between 73–65 %). Berges et al. [44] studied only one appliance (refrigerator), and they found it significantly hard to detect when it changed its states to defrost cycle, due to the need for better signature and more advanced machine learning technique than kNN.

Zeifman [57], stated that six days of active power data took two minutes of processing off-line using MATLAB, and concluding that *"real-time implementation is feasible"*.

Kolter et al. [58], Kolter and Jaakkola [59] were again influenced by smart meters. Using complex unsupervised machine learning algorithms, they managed to successfully disaggregate appliances that are similar in terms of power load. Parson et al. [43], collected real power measurements across six different homes, used then to build and tune general appliance models using Hidden Markov Models.

Power meters installed in homes can store only active power readings for billing, therefore, in this thesis, we focus on measurements using active power only.

2.2.2 Sample Rates

In general, *the higher sampling rate the better the accuracy* to correctly detect household appliances. Analyzing appliance events needs as much information as possible, but that comes at the cost of memory and computational complexity. Based on the sampling rate, NALM can be: high sampling rate (>60Hz) and low sampling rate (<60Hz). High frequency sampling has to be greater than 1 reading every 16.667 milliseconds (Norford and Leeb, [53]; Liang et al. [62]). Researchers have used different ranges of sampling rates such as 10 kHz (Berges et al. [44]), 15kHz (Chang et al. [60]), to rates of 100kHz and over (Patel et al. [61]).

In this thesis, we focus on low-rate NALM solutions, where sampling rates are in the range of seconds and minutes due to memory needs and complexity cost and also availability, since we rely on installed smart meters.

2.2.3 Sensing Types

To store power data in order to test algorithms, two sensing techniques are available: *single-point* sensing and *multi-point* sensing. As it seems clear from the names, single-point and multi-point indicate the number of sensors used to collect data in a household.

Single-point sensing is usually placed in the point where main electricity breaker is, which is mainly at the entrance of a house. In this technique, there is no need to install any extra sensors, which provides a relatively low cost method but requires occupants to manually record their activity to provide ground-truth to researchers, to run their experiments. Researchers used single-point sensing such as [(Froehlich et al. [63]), (Intille et al. [64]), (Tapia et al. [65])] taking advantage of low-memory requirements but had a questionable ground-truth as it can be prone to human errors unlike multi-point sensing.

Multi-point sensing refers to monitor power consumption using more than one point. Researchers like (Berges et al. [44]; H. Kim et al. [56]; Tsai & Lin [52]; Froehlich et al. [63]; Zeifman [57]) have used multiple sensors to monitor individual appliances running in parallel with total meter reading having specific appliances plugged into their own plug-level meter; this provides an accurate ground-truth. However, it has higher computational cost in case of a greater number of appliances being monitored and tested due to memory capacity.

2.3 Event Detection

Detecting appliance-events (when an appliance turns ON or OFF) as they occur can be challenging and still under development. Without a reliable way to detect events, NALM algorithms cannot proceed with the tasks of feature extraction, which calculates certain features per event-window and classification which is deciding which event belongs to what appliance. Events not detected contribute to inaccuracies. Two methods are usually used by researchers which are: Edge detection and signature matching, using probabilistic approaches. The two methods have their advantages and disadvantages as it is discussed in this section.

2.3.1 Edge Detection

Monitored power constantly changes its state depending on which appliance is running and in what mode (eg, operating or standby), when power reading goes high or low, creating a raising edge or falling edge (edges are threshold values to detect possible change in power values). These edges (if high enough) can signal that an event has occurred. This is known as *edge detection* which is usually used in NALM methods to create event windows. However, researchers (Norford and Leeb [53], Baranski and Voss [66], Tsai and Lin [52], Liao et al. [105]) have used different thresholds to detect edges; if a threshold was set too high then it will not detect small appliance activities, and if it was too low then it will be too sensitive and creates many events for the same appliance.

Norford and Leeb [53] used two thresholds of 3 kilo Watt and 5 kilo watt [kW] to detect appliances activity in commercial buildings. However, they argued that their edges were not suitable to detect residential homes, because household appliances consume much less power. For example, a heat pump, would only consume about 1.5 kilo watt in average. Likewise, Baranski and Voss [66], used a much lower threshold of 80 watt, as they found that most household appliances run in the range of 200 watt. On the other hand, Tsai and Lin [52] used an adaptive threshold. They represented their threshold step based on rising or falling of the current value ΔI in Amps with a different value α in the measured current signal. Leeb et al. [67] used complex multi-scalar edge detectors that can correctly identify events on multiple measurements but needed some tuning to correctly identify events.

Luo et al. [68] published work on extending *Generalized Likelihood Ratio (GLR).* They monitored HVAC loads in commercial buildings at different frequencies (8Hz, 1Hz, 0.5Hz, etc). GLR used a ratio of probability distributions before and after a step change in power load was detected. The natural log of this ratio was used to calculate a *"decision statistic"*. In other words, instead of using edge detection they used a statistical algorithm that is triggered on deviations from the mean power reading. After five hours of monitoring, they were able to detect 16 out of 17 on/off events.

Since we are concerned with low-complexity methods, we use multiple thresholds to detect different ranges of appliances in this thesis.

2.3.2 Signature Matching

Berges et al. [44] argues for a probabilistic approach towards event detection, and used the Generalized Likelihood Ratio (GLR) work from Luo et al. [68]. Berges et al. [44] modified the work by continuously computing the standard deviation instead of setting fixed event parameters during initial training. They added a *"voting scheme"* that allowed each sample taken within the detection window to vote optimal change value to be selected. These modifications to the GLR improved the accuracy of the output.

Kolter et al. [58] used a probabilistic approach because edge detection is not suitable for very low sampling rates (hourly in this case). The authors used a *sparse coding* technique that focused on the task of disaggregation not classification and used hourly energy consumption amounts rather than real power readings. They then trained basis functions to detect appliance usage. They note that edge detection works best with high frequency sample rates.

2.4 Feature Extraction

Once an event is detected, it is compared to certain values that is calculated as a feature based on a wide range of measured characteristics such as current, real power etc, would be of significance. These different features usually lead to various signatures which are categorized into steady-state or transient-state and a derived *signal transform*. The NALM algorithm should be trained by labeled data of the feature selected. A proper set of features will be extracted and compared against the labeled data to ensure a precise classification task, hence accurately identifying appliances when an event happens. The following subsections present the proposed signatures in the current literature.

2.4.1 Steady-State Signature

A signature is a certain value corresponds to certain behavior. Steady-state signature is one of the signatures that is usually devoted to a monitored signal which does not vary or varies very little with respect to time. It can be formally defined as a fixed sum of waveforms or a finite number of waveforms.

The steady-state signatures were defined by Fiqueiredo et al. [69] [48] as *" a difference between any two samples of a sequence that does not exceed a given tolerance value"*. They carried out their investigations into real power steady-state signatures only, however, very similar signatures were shown based on different appliances, resulting in significant classification errors. Therefore, ratios between rectangular areas that are usually characterized by the so-called successive states values were identified to differentiate the appliances.

It is worth mentioning here, that steady-state signatures might become unclassified particularly for low powered equipment, such as cell phone charger, smoke alarm etc. This can be attributed to no change in power values, thus the event cannot be easily detected and will be considered as noise.

2.4.2 Transient Signature

Transient signature is a time dependent type of signatures which usually occyres at the beginning of appliance operating cycle. In fact, an appliance normally consists of various electrical components (for example transistors, resistors, capacitors etc.) and noticeable alterations in the power signal can occur, which signify an incomparable transient signature [Laughman et al. [54]; Berenguer et al. [51]].

Signatures of different appliances and different monitored power lines can be captured by using the transient waveform analysis procedure [for background information see Chan et al. [70]; Tse et al. [71]; Leeb et al. [67]].

Norford and Leeb [53] conducted a study to store transient signatures in the form of a *"precise time of v-sections"*, where *v-sections* are considered to be the monitored variation through the the transient signature. A study was carried out by Tsai and Lin [52], who did the calculation of the maximum, average, and root mean square values of the transient signature and represented them as features, which were used for subsequent classification applying neural network and a k-nearest neighbor algorithm.

In this work, we focus on steady-state features as high frequency sampling is required to obtain a high degree of signal uniqueness to capture transient signatures, as well as, transient stage is usually fast, then calling for more expensive acquisition equipment.

2.5 NALM Classification Strategies

Classification or *Machine Learning* technique is the most important step in NALM; machine learning is used to help computers or robots to make a decision when exposed to new data without being explicitly programmed. In this section, we provide details on famous machine learning techniques.

2.5.1 Supervised Learning Methods

Supervised learning is the task of learning information from a labeled data which can be then used to decide the outcome for a classification problem.

2.5.1.1 Optimization-based Methods

Optimization based methods deal with the task of power load disaggregation as an optimization problem, as they compares between an unknown load with possible known loads that are present in an appliance dataset trying to find the best match between them with the lowest error [62]. It can be formed mathematically as:

$$\text{class} = \text{argmin}_i \, \|\hat{y}_i - y_i\| \tag{2.1}$$

where $\hat{y}_i$ is the appliance feature available in the signature library, and y_i is the new feature extracted due to occurrence of an unknown event of an appliance i [62].

However, the main challenge using this method is with overlapped loads, as disaggregation algorithms need to consider possible overlapped appliances. Researchers [73],[62],[66], [75] proposed different techniques to solve the optimization problem.

In [73], a number of appliances was monitored and their switching active power events were stored and matched to new testing events using an optical sensor sampled by one second, clustered their training data using *Self-Organizing Data Analysis Technique (ISODATA)* and adjust clusters based on appliance types.

In [62], researchers used a collection of algorithms in parallel combining optimization and pattern recognition methods to compare an unknown power load with a set of known appliances and the most common decision by all of them is then voted to be the correct decision.

In [66], researchers used Genetic algorithm that works similar to *pseudo code genetic* to detect unknown appliances, pseudo code genetic algorithm presented in [76]. By doing so, the genetic algorithm combines different states to create appropriate finite state machines due to constraints of electrical end uses. However, their method managed to detect typical on and off loads.

In [75], a technique based on integer programming was introduced. The authors measured current waveforms for selected appliances to form a pool of signatures in a dataset with a high sampling rate of 25 micro seconds. However, this method suffered from similar waveform appliances not being correctly detected such as fluorescent lamps.

The major drawback of these methods is that it becomes a challenge to reduce the complexity of these methods especially if any unknown loads are present in the aggregated power load data. Additionally, appliances with similar or overlapping load signature are difficult to discern using these approaches.

2.5.1.2 Pattern Recognition Methods

Pattern Recognition Methods are very common method used by researchers to detect certain behavior in appliances. Researchers in [77], used a Bayesian approach to detect household appliances using real power and state-change (change in operating or standby) information. Naive Bayes classifier was trained using individual appliances and used to classify total power load measurements. However, the method showed good performance but only a small number of appliances were tested. Likewise, in Sanquer [78], *Hierarchical Bayesian* method to reduce the computational cost as feature extraction and training steps was performed jointly using a set of features. Their method tested two appliances vacuum cleaner and refrigerator over a database of transients.

Researchers have shown that the *temporal information* in combination with real power values can facilitate the power load disaggregation algorithms [37], [124]. However, *Artificial Neural Networks (ANN)* [Ruzzelli [79], Kelly [80], Chang [60]] was proven to perform well for the task of power load disaggregation due to their ability to incorporate in their learning, temporal as well as appliance state transition information,

In [60], the authors proposed a *Back-propagation Artificial Neural Network (BP-ANN)* method that uses *Particle Swarm Optimization (PSO)* algorithm to optimize the parameters in order to enhance performance of ANN, using steady-state signatures such as real and reactive power. By finding the optimum solution of power load pattern recognition, they managed to significantly improve their performance.

In [80], a combination of three different Neutral networks (NN) were used together to perform a deep machine learning task; roughly 1 to 150 million trainable parameters were tested using five known appliances which are fridge, washing machine, dish washer, kettle and microwave samples in 6 seconds. The first Neutral Network model is Recurrent NN which is a feed forward NN and has only one dimensional input and output because it suffers from collapsing with limited memory. The second layer of this approach is called De-noising Auto-encoder NN, this technique is mainly used in image processing which is basically eliminates background noise from an image, it is done by encoding the input and them decode the output by a compact vector representation. The final layer is responsible for the start time and end time of a targeted appliance event window, basically the start time in a time window is 0 and end time is 1 and it is 0.5 half way through. However, this method showed well performance for two-state appliances but does not perform well on multi-state appliances such as the dish washer and washing machine.

Decision Tree (DT) has been used for power load disaggregation also and showed good performance. In [81], multi-class DT is used to classify four loads; battery charger, fluorescent lamp, personal computer and incandescent light bulb sampled at 10 seconds and voltage, current and phase angle were recorded. After applying Discrete Fourier Transform to extract the fundamental frequency component, the change in these power components is calculated to help developing the classification tree model. It was concluded that an increase in accuracy with up to 26.27 % was made by using the change in power compared to using the actual measured power.

In [82] an application of discrete wavelet transform was used in the process of NALM using an ensemble of DT, by assigning weights to the predictions made by each base classifier testing the same four loads; battery charger, fluorescent lamp, personal computer and incandescent light bulb at the transient state. It was found that third order (out of five) Daubechies filter (Daub3) can achieve highest classification accuracy reaching 95.83 % and 93.06 % for training and testing respectively. Also it was concluded that both accuracy and computational cost increased when increasing the number of decision trees.

Graphical Signal Processing (GSP) is a signal processing concept that effectively captures correlation among data samples in time and space by embedding the structure of signals onto a graph but suffers from high computational cost. In [83], two novel methods were introduced using GSP. The first method minimizes the total number of iterations. The second method further refines the total number of iterations using a *simulated annealing* technique which allows to split the data into manageable *windows* in order to decrease the execution time. It was concluded that, both methods showed competitive performance and the second method successfully reduced complexity, however, it did not enhance performance.

On the other hand, *Support Vector Machines (SVM)* have shown good performance in classifying appliances especially using harmonic signatures and low frequency features. Researchers in [84], generated data from 10 houses in India with roughly 30 devices in a minute-wise logs. They compared performance between Support Vector Machines (SVM), Decision Tree (DT), Naive Bayesian and Artificial Neural Network (ANN); the best performance was accomplished by SVM with roughly 30% accuracy. However, they also argued that by adding time stamp as a variable to classifiers, accuracy significantly increased to 70%.

In [85], authors used Linear SVM in comparison with polynomial SVM and ANN to disaggregate similar appliances such as LCD TV, CPU and laptops using *Discrete Wavelet transformation coefficients* as an input to their classifiers. However, Linear SVM gave the best performance with 100% accuracy in correctly detect similar appliances.

In [49], authors used a combination of multi-state linear and Gaussian SVM techniques along with k nearest neighbours with k=5. They collected steady state data with roughly 15 seconds for six appliances including microwave and 2 LCD screens. They concluded that linear SVM and 5NN gave the best performance against all with up to 97% F-measure.

In [86], linear, polynomial and Gaussian SVMs and ANN were put into comparison using data from eight devices by generating fast Fourier transform coefficients as input to their classifiers. They concluded that all methods performed well but SVM algorithms showed relatively high computational resource requirements.

In [87], authors tested 24 appliances with excluding one appliance out for testing and performed one-class SVM. However, they used Receiver Operating Characteristic (ROC) curve to evaluate false positive and true negative values and achieved a total rate accuracy of 97.7%.

A hybrid Support Vector Machine/Gaussian Mixture Model (SVM/GMM) model has recently been proposed by Lai et al.[88] in which GMM is used to describe the distribution of current waveforms, so as to find power similarity; while an SVM performs classification on the extracted power features in order to recognize operations of targeted loads.

SVM have shown significant performance in classifying appliances especially using low frequency features and high number of existing appliances. However, it provides high computational cost, therefore, SVM will be the main approach in this thesis, with the focus on lowering its computational cost for real-time response.

2.5.2 Unsupervised Learning Methods

Unsupervised learning methods do not require labeled data which is a label for every event-window for which appliance is running, unlike supervised learning method, but often need preset threshold values to study similarities between power data.

2.5.2.1 Hidden Markov Model (HMM) based methods

A widely used structure for modeling appliance consumption behavior is the Hidden Markov Model (HMM). It requires *tunning* a set of hidden parameters, such as the operation condition of an appliance, and an estimation of consumption data. There are different versions of HMM used by NALM: Figueiredo et al. [69] used *Expectation-Maximization (EM)* to tune models before disaggregation; an optimization step is needed to be used to estimate the power demand of each appliance.

Kim et al. [89], used factorial Hidden semi-Markov model (FHSMM) to model appliances to provide a better estimation of the state occupancy durations of the appliances, introducing a *Conditional FHMM (CFHMM)* to integrate additional features related to the behavior of the appliances that are used in the house, using Expectation Maximization (EM) algorithm and Maximum Likelihood Estimation (MLE) to tune other features like ON-duration distribution, OFF-duration shape, correlation of appliances and behavior of occupants.

Kolter and Jaakkola [59], used *Additive Factorial hidden Markov models (FHMMs)*, where each HMM chain produces its own emissions, but observations are defined as collections of the independent emissions using an optimization problem called *Additive Factorial Approximate Maximum A-Posteriori (AFAMAP)*.

Parson et al. [43] used HMM to model each appliance type in a household, extracting only the signal that is useful for the appliance tunning by using a sliding window method in a given aggregated data. Pattem [90] used HMM with *segmented Viterbi* to successfully disaggregate as many known appliances as possible by using ON/OFF power transitions as a signature and also applied residual analysis to handle lower power signatures.

Johnson and Willsky [91], benefit from using *Hierarchical Dirichlet Process HMM (HDP-HMM)* structure to solve large scale complexity from provided data, and provided *semi-Markov modeling (HSMM)* to define the Markov transition with higher accuracy.

Finally, in [92], a Graphical Signal Processing (GSP) method was introduced using low-rate four real houses, it was concluded that *blind* GSP method provided comparative performance to that using supervised approach.

2.5.2.2 Source Separation

The source separation problem is the task of recovering an original event from a combination of events. Figueiredo et al. [63] used the single-channel source separation method to solve energy disaggregation problem; his method was based on a *Non Negative Matrix Factorization (NMF)*; where a matrix of events is factorized into *usually* two matrices with no negative elements in all three matrices. Such approximation makes the resulting events more solvable. However, in order to improve the performance, additional information about dependencies between different appliance are used, introducing the technique called *Source Separation Via Tensor and Matrix Factorizations (STMF)*; using these methods together helped making the dependencies between the sources clear.

Wytock and Kolter [93] used source separation for energy disaggregation by finding a correlation between different appliances in a *convex optimization problem*; assuming that every appliance can be represented as a linear function of some component bases and performed a contextually supervised method which is based on temperature time series, the non-linear dependence, represented as Gaussian functions, is defined as basis for the energy consumption. However, after theoretical analysis, it was found that linear independence between features for different signals is essential to perform accurate source separation.

Goncalves et al. [94] present a *Blind Source Separation (BSS)* using Genetic k-means algorithm to obtain a representation of the appliances consumption state from the observation of steady-state event in real and reactive power profile. After that, *Matching Pursuit (MP)* method is used for reconstruction, by using the cluster information to characterize each appliance.

Liao et al. [95] and Elafoudi et al. [96] presented a method based on *Dynamic Time Warping (DTW)* for template matching by performing a non-linear mapping of new detected events. After a signature detection within the aggregate power consumption, classification is then performed by calculating the *minimum distance* between the signature detected and the matching cluster.

2.6 Evaluation Measurements

There are different evaluation methods that are used by researchers such as basic accuracy, Precision, Recall and F-Measure.

2.6.1 Accuracy

No consistent method with respect to measure performance accuracy was found, according to (H. Kim et al. [56]; Zeifman & Roth,[37]). With the new efforts that are being devoted to this subject in particular, a progress or even a slight change will be surely seen over the next decade.

Despite of numerous datasets have been released in public in order for researchers to conduct their own testing, the majority of those researchers still deal with the basic form in assessing the performance accuracy.

2.6.1.1 Basic Accuracy

The basic accuracy [52] measure employed by the vast majority of NALM algorithm researchers is expressed as:

$$Accuracy = \frac{TP}{total \quad number \quad of \quad events} \tag{2.2}$$

where true positive (TP) presents the correctly detected event. Tsai and Lin [52] investigated this accuracy measure by using correct signals matched and showed a sufficient accuracy of approximately 95%. A recognition accuracy on training and testing results was implemented by Chang et al. [60], who reported as high as a 100% accuracy. One of the drawbacks of the aforementioned reports is that the classification's performance was not assessed, so researchers may not rely on the work outcomes (Metz, [97]; Sokolova et al. [98]). For example, a fridge would have an accuracy measure of 90% as long as it operates only 10% of the time whilst an NALM algorithm (100% of the time) shows that the fridge was not working. H.Kim et al. [56] concluded that accuracy results can be *"very skewed"* particularly when dealing with appliances that are off-power and also appliances that have a relatively rare event. Consequently, a high accuracy will be noticed as better accuracy performance measures are needed.

2.6.1.2 Total Energy Correctly Assigned (TECA)

Another form of accuracy measurements is Total Energy Correctly Assigned (TECA) [92] which described as:

$$Acc = 1 - \frac{\sum_{t=1}^{T} \sum_{i=1}^{K} |\hat{y_t}^{(i)} - y_t^{(i)}|}{2 \sum_{t=1}^{T} \bar{y}_t} \tag{2.3}$$

where T is the number of test samples. $\hat{y_t}^{(i)}$ denotes the estimated power of the i_{th} appliance and $y_t^{(i)}$ is the actual energy consumption by appliance i at a time instance $t^{(i)}$. $\bar{y}_t$ denotes the observed total power consumption at time instance $t^{(i)}$ [92].

In [39] and [80] authors used TECA to measure accuracy of their approaches of FHMM and Neural Nets respectively.

2.6.1.3 F-Measure

F-Measure is one of the accuracy measure forms that usually deals with the classification of text/document and information results. Figueiredo et al. [48] [49] examined their NALM algorithm accuracy by applying F-Measure based on 50 samples of data for each appliances. Berges et al. [44] carried out an investigation into both training and testing of their NALM algorithm by using F-Measure, for comparing consumption of 5.5 days. The rational was to compare the NALM algorithm predictions of an appliance against corresponding readings supplied by a plug-level meter. Kim et al. [56] stated that *"F-Measure measures binary classifier outcomes and power signals cannot be considered binary"*. Accordingly, it can be claimed that a better accuracy performance measure is importantly required.

However, in this thesis, we focus on events that were correctly classified and wrongly classified into other appliances taking into account if a feature is unique enough or not. Therefore, we do not use standard accuracy, and we turn to use Precision, Recall with a focus on F-Measure, as they include a sense of how many events being falsely correctly or wrongly classified.

The evaluation metrics used are precision (PR) [105], recall (RE) [105] and F-Measure (FM) [105] defined as:

$$PR = TP/(TP + FP) \tag{2.4}$$

$$RE = TP/(TP + FN) \tag{2.5}$$

$$F_M = 2 * (PR * RE)/(PR + RE), \tag{2.6}$$

where true positive (TP) presents the correctly detected event, false positive (FP) represents an incorrect detection, and false negative (FN) indicates that the appliance used was not identified [105]. The values of these measures are between 0 to 100, the higher the values the better the disaggregation is.

2.7 Available Datasets

It is significantly important to validate results with other approaches/researchers, therefore, recent researches collected and published energy consumption datasets as a reference. One can easily apply simulations on real life scenarios from different houses and environments and compare results and finding with other researchers.

A comparison between available datasets, highlighting their main characteristics, such as duration, number of houses and signal sampling frequency is shown in Table 2.1 [99] and [100].

We have chosen REDD [39] from USA, GREEND [40] from Italy and Austria and REFIT [41] dataset from UK as they come from different environments with different low sampling rates with a range of different types of appliances. REDD dataset has 6 different houses with roughly 6 appliances each recorded at 1 Hz sampling rate which was downsampled to 1 minute intervals, and worth 119 days in total. GREEND dataset has 8 average homes with roughly 9 active loads each with a total of 58 appliances with 8 seconds intervals, worth roughly 10 months in total. REFIT dataset has 20 UK homes with roughly 9 appliances each at 8 seconds intervals, the whole dataset is roughly worth 2 years in total.

2.7.1 Labeled Data

In order for NALM algorithms to disaggregate and classify events accurately that are incurred by appliances, more information associated with the appliances operated at the home as well as background knowledge relevant to the home itself, are needed to be accomplished. Therefore, labeling the unlabeled data/events that exist in the power signal based on selected features is the main task of the classifier; thus labeling plays a major role in characterizing how electricity at homes is being used.

As far as labeled data is concerned, current NALM algorithms apply three main methods: signature corpora, finite state machines (FSM) and historical data. For example, in high frequency NALM algorithm particularly when unlabeled transient and steady-state signatures associated with "current readings" are compared against a set of acquirable labeled signatures, the signature corpora is always recommended (Tsai & Lin [52]; Figueiredo et al.[48]).

Table 2.1: List of available Energy Datasets as summarized in [99] and [100]. R stands for Residential.

Dataset	*Sampling rate*	*Duration*	*Houses/ Devices*	*Subject*	*Ground truth*	*Country*
Dataport	1Hz to 1min	4+ years	1200+	R, Com, and Indus	Sub-meters	US
REDD	16500 Hz / 1 Hz	Several months	2 / 5	R	Sub-meters	US
BLUED	12000 Hz	1 week	1	R	labels	US
UK-DALE	16000 Hz /1 Hz	2 years	6	R	Sub-meters	UK
PLAID	30000 Hz	5 sec	55	Individual appliances		US
WHITED	44000 Hz	5 sec	9	Individual appliances		Multiple
Tracebase	1 Hz	1 day	158	Individual appliances		Germany
DRED	1 Hz /1 min	6 months	1	R	Sub-meters	Netherlands
AMPds	1 minute	2 years	1	R	Sub meters	Canada
iAWE	1 Hz	73 days	1	R	Sub-meters	India
HES	2 minutes	1 year	251	R	Sub-meters	UK
REFIT	8 sec	2 years	20	R	Sub-meters	UK
ECO	1 sec	8 months	6/45	R	Sub-meters	Switzerland
ACS-Fl	10 sec	2 sessions /1 hour each	0/10	R and office	Sub-meters	Switzerland
GREEND	6 sec	2 years	7/58	R	Sub-meters	Italy and Austria

For a given appliance, a recorded power signal over a given period of time can technically represent signatures. A signature corpus was built by Fisera and Macek [55] through the training stage of control signals based in building management system (BMS). The BMS was considered as a supervisor to aid identify electrical events. Furthermore, the classifier was trained by using steady-state and transient signatures that were restored in corpus; which are corresponded to a time dependent event selected. In low frequency NALM algorithms, however, signature corpora may not be usable, therefore finite state machines (FSM) can be considered as a reliable option (e.g. Hart [19]; Norford & Leeb [53]; Parson et al. [43]).

Unless for continuously variable appliances, the FSM can be applied to represent a number of various appliance types (e.g. microwave, fridge, dishwasher, oven, clothes dryer, etc.). Unlike signature corpora, the FSM has gone beyond a "static set of data". A study was published by Parson et al. [43] , who explained in detail how an FSM can be adjusted whilst the NALM algorithm runs. The rational was to identify a specific model of a given appliance (e,g., Clothes dryer) by using a generic trained FSM of that appliance. Moreover, each generic appliance FSM is loaded only with electrical characteristic values. For low frequency sampling NALM algorithm, the historical data can be also considered as a suitable option of source of data. This type of data source (historical data or periodic data) contains periodic power readings that can be aggregated and characterized by using different methods (e.g. histogram of appliance usage). Even though historical data can be practically applied in other ways, it is fundamentally functioned for testing datasets. The frequency of appliance usage was analyzed by Baranski and Voss [73] [74] [66]. This was frequently subjected to an inherited algorithm that created an FSM for that appliance.

Since labeling data is an important step, as well as low-complexity NALM, we focus on historical submetered data to label data using information from the same dataset used in this work.

2.8 Summary

In this chapter, we highlighted a number of recent papers that demonstrate existing solutions to the energy disaggregation problem. We described few household monitoring techniques to collect real life data with different ranges of sampling rates. High sampling rates lead to unique features but demand high memory capacity which is pricey and impractical. Therefore, in this thesis, we focus on low-rate NALM solutions, where sampling rates are in the range of seconds and minutes.

The sampling rate can also influence the type of features that can be used. Furthermore, we focus on steady-state features as high frequency sampling is required to obtain a high degree of signal uniqueness to capture transient signatures. For example, low-rate NALM approaches can use only steady-state parameters, including active or real power, reactive power, power factor, voltage or current waveform. In fact, newer research approaches use only active power measurements for simplicity and cost, basically, power meters installed in homes can store only active power readings for billing. Another reason is sensors that can be purchased and attached to individual appliances are relatively inexpensive.

Event detection is an important step in NALM. The most popular method used to detect events is edge detection. A pre-set threshold will be constantly compared with the testing data in order to detect raising and falling edges. Researchers have proposed and tested different methods of edge detection from using one fixed number to constantly measuring variance in data. However, in this thesis, we focus on low-complexity methods, therefore, we use multiple thresholds to detect different range of appliances.

We have also reviewed a number of classification methods used for NALM including supervised and unsupervised algorithms. Supervised methods include optimization theories and pattern matching methods. Both methods showed good performance but come with high complexity cost. However, *Support Vector Machines (SVM)* have shown significant performance in classifying appliances especially using low frequency features and high number of existing appliances, which will be the focus of our research.

Unsupervised methods, include HMM-based method and source separation etc, HMM-based method require tunning for pre-set information about appliances in the testing dataset which can be highly complex and impractical. Source separation suffers from low accuracy unless combined with other methods to increase their performance.

We also gave details on available accuracy measurements such as basic accuracy which can be impractical. As we will use in this thesis Precision, Recall and a focus on F-Measure as they include a sense of how many events being falsely correctly or wrongly classified. Also, in additional to accuracy, recently researchers have published datasets to simulate real life household behavior. We have chosen REDD [39], GREEND [40] and REFIT [41] dataset as they come from different environments and sampling rates and different types of appliances.

Chapter 3

Methodology

3.1 Introduction

The Support Vector Machines (SVM)-based classification algorithm is one of the most popular algorithms in artificial intelligence tasks, but it suffers from high computational cost as discussed in the background chapter. Since it is very important to receive near real-time feedback on which appliance is running in a household, the reduction of SVM execution time is the main focus of this chapter. Therefore, inspired by [101]–[104], we propose a combined algorithm of k-means and SVM for NALM to reduce computational complexity.

The disaggregation procedure usually comprises two steps; Data processing and classification. The data processing step comprises of event detection and feature extraction. Classification, on the other hand, is a common artificial intelligence objective with the task to decide to which class each dataset point belongs to.

The focus of the chapter is in developing a new NALM classification method. that can work with various event detection approaches. For completeness, we start with reviewing the adopted event detection method and then proceed with the proposed classification approaches, starting with Trained k-means, followed by Linear and Gaussian SVM and finally we propose our combined approaches. After that, a simple case study of the disaggregation algorithms is demonstrated for better understanding of our proposed methods.

3.2 Data Cleaning

During data monitoring, short spikes and drop in connecting with the server could happen, that can give wrong results in energy disaggregation task. In this thesis, data cleaning was mainly performed by ignoring event windows with less than three samples and consider event windows that start and end with zero values or lost data, which means that extra zero values will not be considered as part of training or testing data. Low values less than 20 Watt were considered to be noise and therefore neglected.

3.3 Event Detection and Feature Extraction

The task of event detection is to detect changes in time-series aggregate load curve due to one or more appliances being switched on/off or changing its state.

Here we closely follow the notion of [105], [106]. Let $\mathbf{M}$ be a set of all known appliances in the house. Let $p(t_i)$ be active power measured at time instance t_i. Without loss of generality, in the following we denote $p(t_i)$ as $p(t_i) = p(iT) = p(i)$, where $T = t_i - t_{i-1}$ is the sampling interval.

The disaggregation task is to find $p_j(i)$ for all j, such that $p(i) = \sum_{j=1}^{M} p_j(i) + n(i)$, where $p_j(i) \geq 0$ is the power load of appliance j and $n(i)$ is the measurement noise. Note that $p_j(i)$ is zero if the appliance is off at time instance iT. Now let W be a set threshold. Then, if $|p_j(i) - p_j(i-1)| \geq W$ then the appliance j has *changed state* at time instant iT.

Threshold W needs to be set low enough so that for all j, if $|p_j(i) - p_j(i-1)| \leq W$ Appliance j did not change its state and, otherwise, it did change its state. W depends on the set of appliances being monitored, and is adapted automatically during the training process based on the minimum state transition that needs to be detected and the maximum variation of the active power within one appliance state across all appliances' states, that is

$$W = \max\{\min_{m \in \mathbf{M}} \mathbf{p_m}, \max_{\mathbf{m} \in \mathbf{M}} |\max(\mathbf{p_m}) - \min(\mathbf{p_m})|\}, \tag{3.1}$$

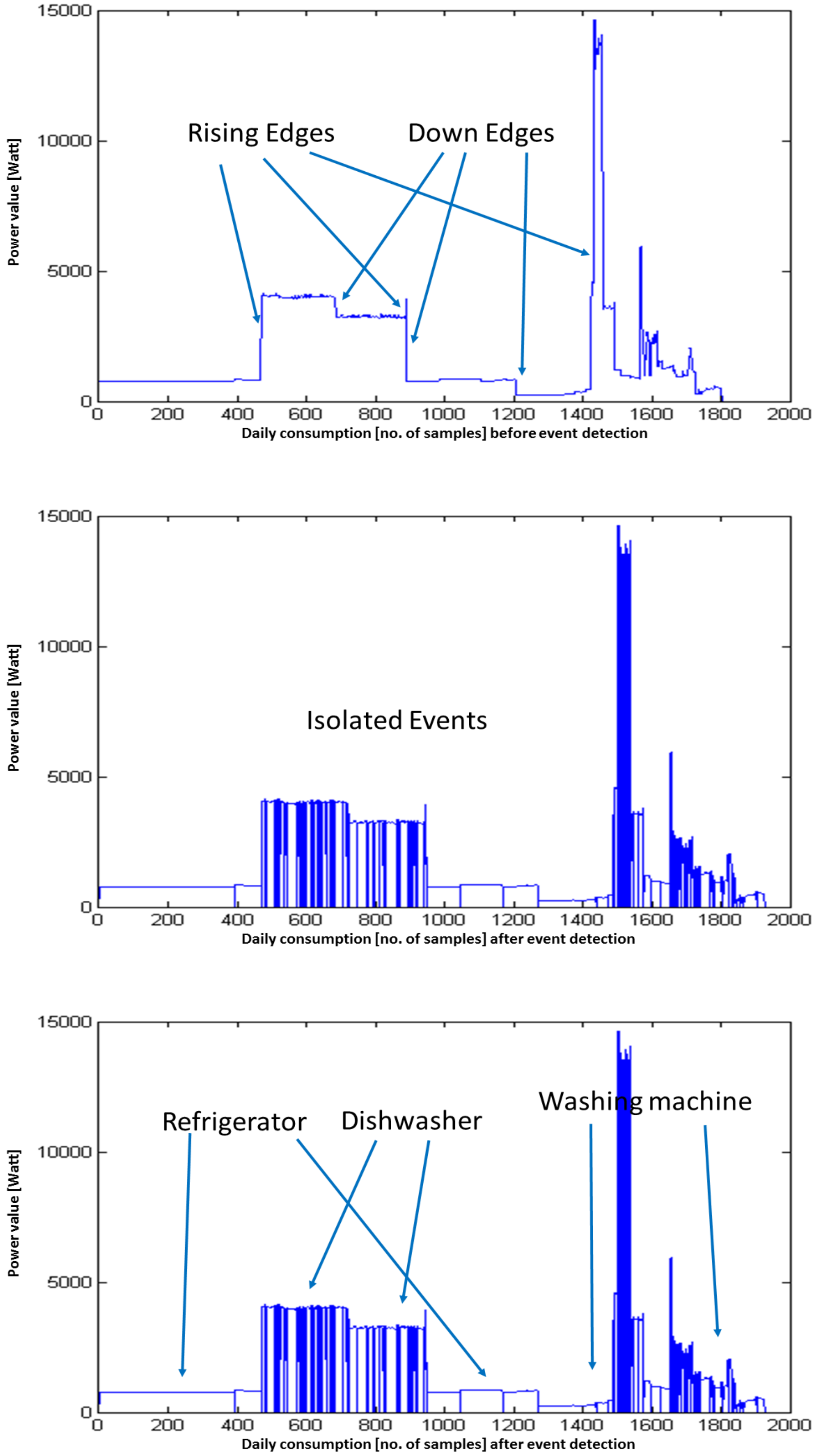


Figure 3.1: Example of Event Detection and Labelling step from REFIT dataset House 17.

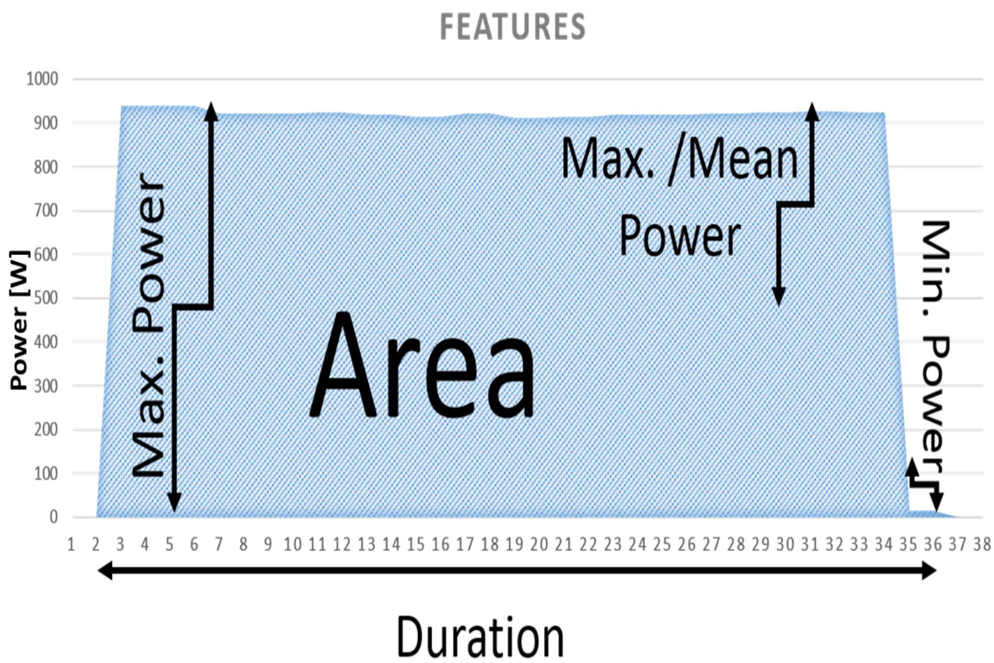


Figure 3.2: Example of Feature Extraction

where $\mathbf{p_m}$ is a vector of active power readings of appliance m.

Note that the value of W depends on the set of available appliances, and is adaptively changed as appliances are being disaggregated and removed from the aggregate load.

An *event* occurs whenever an appliance changes its state. Edge detection is used to detect events by comparing $|p(i) - p(i-1)|$ with W. We say a *window* of the event started at time l_s and ended at l_e if an appliance changed its state at l_s and l_e, and

$$|[p(l_s) - p(l_s - 1)] + [p(l_e) - p(l_e - 1)]| \leq C, \tag{3.2}$$

where C is parameter smaller than W.

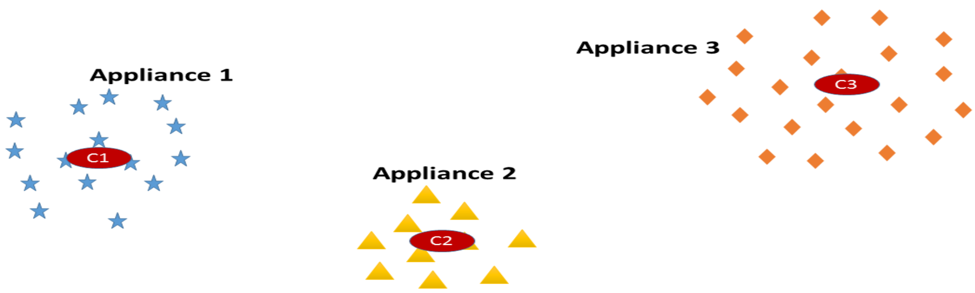
Next, features are calculated and stored from each event window; each window is given a label of which appliance is running as illustrated in Figures 3.1. There are different types of features namely time and phase signatures as reported in [107]. In our experiments, extracted time features include (1) all active power readings in the event window, (2) rising/falling edge magnitude , (3) maximum/minimum active power value in the window, (4) duration of the event, (5) area, calculated as the area of the irregular polygon formed by the active power (Watt) samples in the event window, i.e., the energy of that event window in Joules (see Fig. 3.2). The optimal features to use, for each appliance, will be selected using the training dataset. Extracted features from each detected event are matched to the pre-defined appliance classes using a trained classifier.

3.4 NALM Classification Algorithms

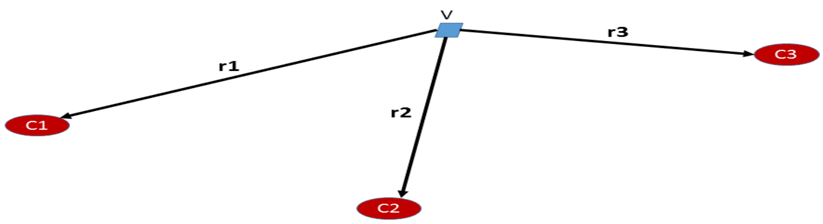
Classification is carried out in two stages: training and testing. Training is performed using a labeled dataset mapped into a finite dimensional space. By a labeled dataset, we mean a collection of event windows with labels indicating which appliance was running. For example, if a microwave was switched on, the resulting event window of active power samples will then be labeled as microwave. Testing then is accomplished by using information from the training step to classify new input events.

3.4.1 Trained k-means

The well-known k-means is a clustering method [108], but here we adapt supervised k-means [109] [110], which we term Trained k-means, to perform supervised clustering similarly to [111]. Trained k-means uses a labeled dataset to classify the input data based on minimum distance classification similar to [112].



(a) Training step



(b) Testing step

Figure 3.3: An example of a Trained k-means separable problem in a 2 dimensional space. C1, C2 and C3 are cluster heads corresponding to Appliances 1, 2 and 3, respectively. r1, r2 and r3 are Euclidean distances measured with a new testing point V.

During training, aggregate events with Appliance A label from the entire training dataset are grouped together, forming the Appliance A class. Like conventional k-means, the centroid of each appliance class is set as its head C. Note that, the number of classes k is always equal to the number of known appliances in the household. When a new testing sample (feature vector) is introduced, it is compared with all heads, and the minimum distance determines the classification outcome.

For better illustration, assume we have three known appliances (Appliance 1, Appliance 2 and Appliance 3) in a house which are needed to be classified. As illustrated in Figures 3.3a and 3.3b, training and testing steps are as follows:

- In training, all appliance events with the same label are grouped together; a class head, is then calculated as the average value of all data points in that class, as in the example shown in Figure 3.3a.
- Testing step is straightforward. When a new event point V is introduced, Euclidean distances (r_1, r_2 and r_3) are then measured between that event and all possible class heads (three heads in this case; C_1, C_2 and C_3)); the minimum distance determines the closest class to that event. The label of that class is then assigned to event V as in the example shown in Figure 3.3b. These training and testing points are effectively features, that could be 1D (power values only), 2D, 3D, etc.

3.4.2 Linear and Gaussian Support Vector Machines

SVM-based algorithms are optimal classifiers in the presence of noise and proven to perform well for NALM applications as reported in [113], [114], [115] and [25]. SVM is a binary classifier, it separates two classes at a time. Training process is accomplished by deciding a proper margin of the largest separation between the nearest training data points. This is done by calculating a weight w_0 between observations, which is a constant for each two points α and z using dot product ($w_0 = \alpha_i z_i$) in a linear decision surface. The smaller the weight the better it separates the two classes [42]. Basically, we test similarity between α and z, and if w_0 was 1 that means α and z are very close. Likewise, if w_0 was 0 that means α and z are further apart. After that, optimization is performed to decide which points to keep as support vectors in order to have a proper decision margin as illustrated in Figure 3.4.

In Figure 3.5, it can be seen that a linear boundary is not clear, therefore, we sometimes need a non-linear solution. Radial Basis Function Kernel SVM, or Gaussian SVM [116] is a popular non-linear type of SVM. It uses Kernel function to evaluate the separation between two classes in a feature space as in Equation 3.3, where σ is a parameter [116].

$$k(\alpha, z) = \exp\left(- \frac{\|(\alpha, z)\|^2}{2\sigma^2} \right) \tag{3.3}$$

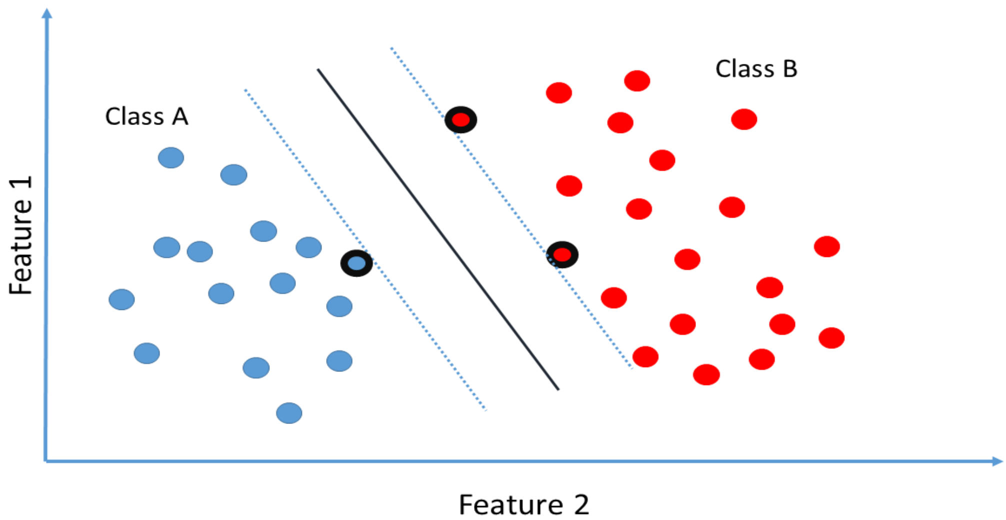


Figure 3.4: An example of a linear SVM separable problem in a 2 dimensional space. Blue points are class A training points. Red points are class B training points. Black circled points are support vectors for each class.

NALM is a multi-class problem, as we normally have more than two appliances (classes) to distinguish between. There are two main strategies for multi-class SVM: (*one-against-all*) and (*one-against-one*) [117]. *One-against-all* strategy consists of constructing one SVM per class, which is trained to distinguish the samples of one class from the samples of all remaining classes. Usually, classification of an unknown pattern is done according to the maximum output among all SVMs. *One-against-one* strategy, also known as "all pairs", consists in constructing one SVM for each pair of classes. Usually, classification of an unknown pattern is done according to the maximum voting , where each SVM votes for one class [117].

As we train SVM classifiers to separate one appliance at a time as reported in [118] and [119], we use *one-against-all* due to its popularity and lower computational time compared to (*one-against-one*). After an appliance has been classified, its contribution is removed, the threshold used for edge detection is adapted, and disaggregation is attempted on the next appliance.

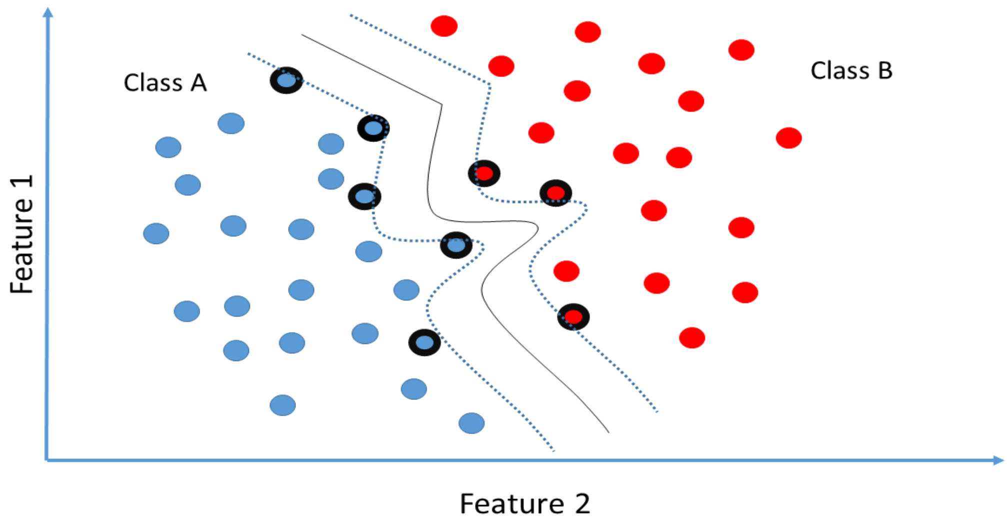


Figure 3.5: An example of a Gaussian SVM separable problem in a 2 dimensional space. Blue points are class A training points. Red points are class B training points. Black circled points are support vectors for each class.

3.5 Proposed Combined Algorithms

To combine Trained k-means and SVM, we first train k-means as explained in Section 3.4.1 using the entire training dataset. As a result, k classes, each corresponding to one appliance, are formed with a centroid as head. Next, all feature vectors falling in Class i that are at an Euclidean distance larger than r from their head, form a subset of feature vectors C_i *that is removed from Class i and used* to train an SVM for Appliance i. r is a pre-set threshold, unique for each house, obtained heuristically, that is used to tradeoff complexity and performance. See Fig. 3.6 for an illustration. Note that, in this way, SVM will be trained using a significantly reduced dataset obtained from the trained k-means classifier, and hence the combined k-means & SVM algorithm complexity will be reduced, compared to SVM classification alone. Algorithm 1 shows the training steps, where $d(x, y)$ denotes the Euclidean distance between vectors x and y.

Algorithm 1 Training: Perform training on the extracted features of the collected dataset.

```
function TRAIN(Labelled training dataset L, |M|, r)
    k=|M|                                          ▷ Number of Appliances
    [Cluster, c]=kmeans(k, L)                      ▷ Call kmeans function
                                   ▷ Returns Cluster distribution and cluster heads c.
    for i = 1 : k do
        C_i = {∅}
        for ∀l ∈ Cluster_i do          ▷ Cluster_i denotes i-th cluster in Cluster
            if d(l, c_i) ≥ r then      ▷ c_i denotes i-th element of k-length vector c
                C_i = C_i ⋃{l}
            end if
        end for
        SVMTrain(C_i)                  ▷ Call conventional SVM training function
    end for
end function
```

Algorithm 2 Testing: Perform testing on the extracted features of the collected dataset.

```
function TEST(Testing dataset, Clusters, c, |M|, r)
    k=|M|                                          ▷ Number of Appliances
    for i = 1 : k do
        C_i = {∅}
        for ∀l ∈ Cluster_i do
            if d(l, c_i) ≥ r then
                C_i = C_i ⋃{l}
            elseClassify sample i to the appliance corresponding to c_i
            end if
        end for
        SVMTest(C_i)                   ▷ Call conventional SVM testing function
    end for
end function
```

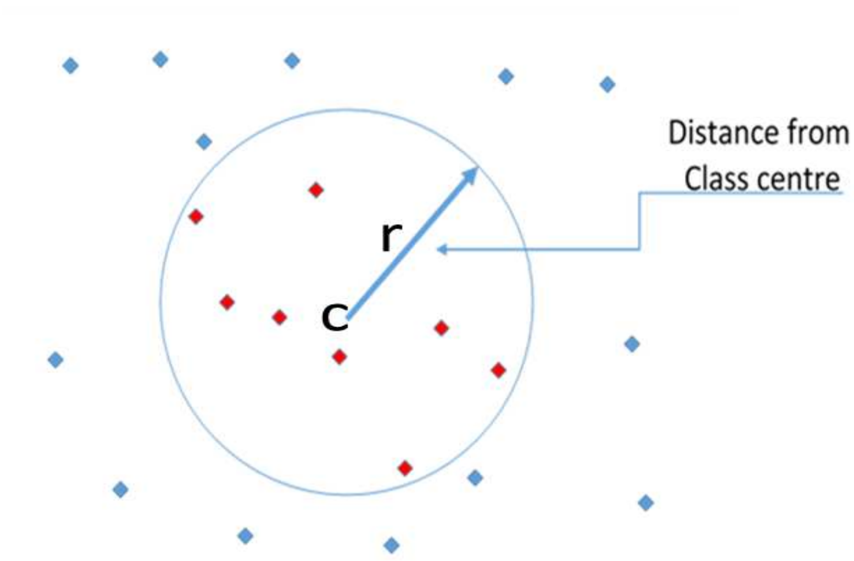


Figure 3.6: Filtering data samples in the proposed algorithm. Red rhomboids inside the circle centred at cluster head* c *will not be fed into the SVM training module.

Testing is straightforward and shown in Algorithm 2. Samples at distance less than r from a cluster head are classified to the appliance corresponding to that cluster head. All other samples are classified using SVM.

More specifically, if the Euclidean distance between a tested sample and any cluster head is smaller than the pre-set threshold, then the sample is classified to the closest cluster head. Otherwise, the sample is input to the SVM classifier.

The proposed combined method has low execution time, since many samples will be classified rapidly using k-means, and only a small amount of samples that are far away from their heads, will be fed to the SVM classifier. However, the proposed algorithm maintains high performance, since SVM improves classification for samples that would most likely be incorrectly classified using the trained k-means.

3.6 Illustration of Disaggregation Algorithms

In this section, we illustrate the proposed algorithm using a simple case study on House 5 of the REFIT dataset. Three appliances were chosen, as they are frequently used in daily activities, which are: tumble dryer, television and toaster. We used two days for training and ten days for testing. First, events are detected based on pre-set thresholds, different features are calculated and stored forming different (2D, 3D, 4D and 5D) features. Then, we label training events depending on which appliance was running. Note that we also label testing events for verification purposes. Next, we feed training feature combinations along with their label-vectors to the classifiers to perform training and testing steps.

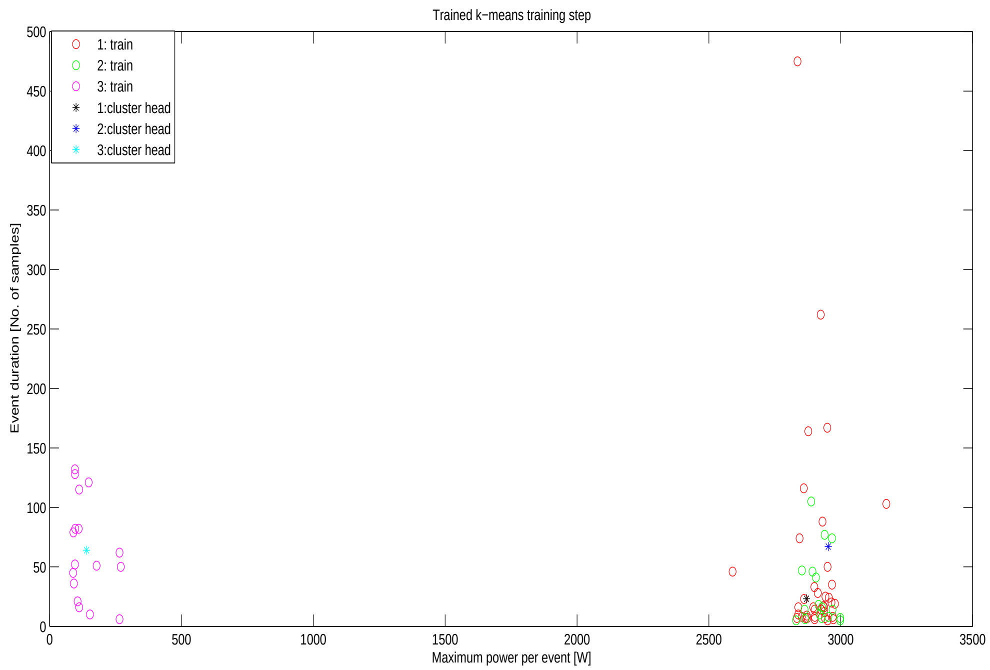


Figure 3.7: Training step of Trained k-means. Circles are training points of each appliance, red are for the tumble dryer, green are for the TV and purple are for the toaster. Stars are cluster heads which are then stored for the testing step. x-axis is maximum power value per event. y-axis is event duration [No. of samples].

3.6.1 Classification training and testing steps

3.6.2 Trained k-means

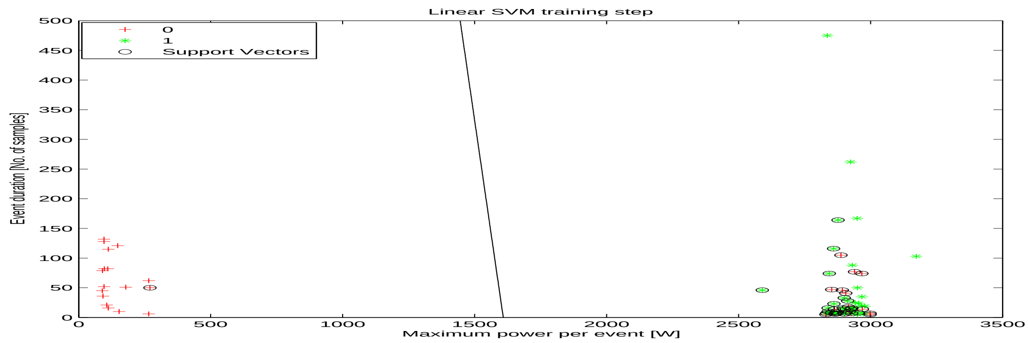
Trained k-means uses training points of the same label to form a cluster head using mean values as shown in Figure 3.7. Circles are training points of each appliance, red circles are for the tumble dryer, green circles are for the TV and purple circles are for the toaster. Stars are cluster heads which are then stored for testing. In testing, features will be extracted from every new testing point and will be compared to all possible cluster heads using Euclidean distance. The label of the closest head will be assigned to the testing event.

3.6.3 Linear and Gaussian Support Vector Machines

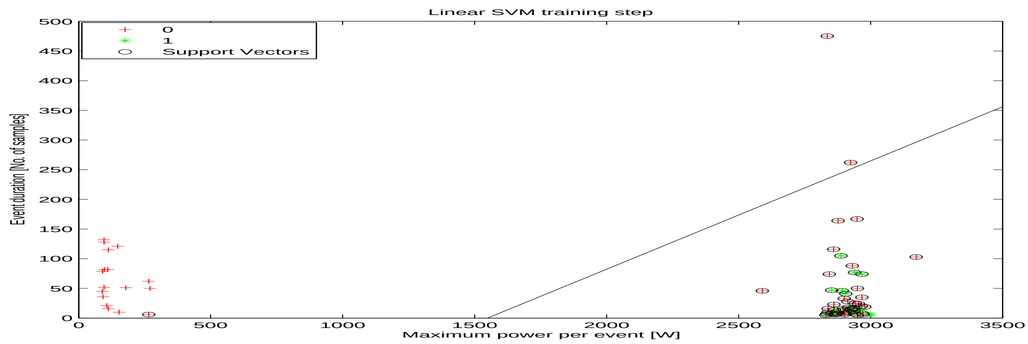
Linear and Gaussian SVMs usually classify between two classes only but here we use *one-against-all* multi-class SVM, which trains and tests two appliances at a time as shown in Figures 3.8a, 3.8b and 3.8c for Linear SVM, and Figures 3.9a, 3.9b and 3.9c for Gaussian SVM. Training points in green are the current class, red points represent every other appliance and black circles are support vectors used to separate the two classes. The first step is to train and detect appliance 1 which is tumble dryer as in Figures 3.8a and 3.9a then testing points of tumble dryer are removed from the training dataset. The second step is to train and detect appliance 2 which is TV as shown in Figures 3.8b and 3.9b then testing points of TV are removed from the training dataset. Finally, we train and detect appliance 3 which is toaster as in Figures 3.8c and 3.9c.

3.6.4 Linear and Gaussian combined algorithms

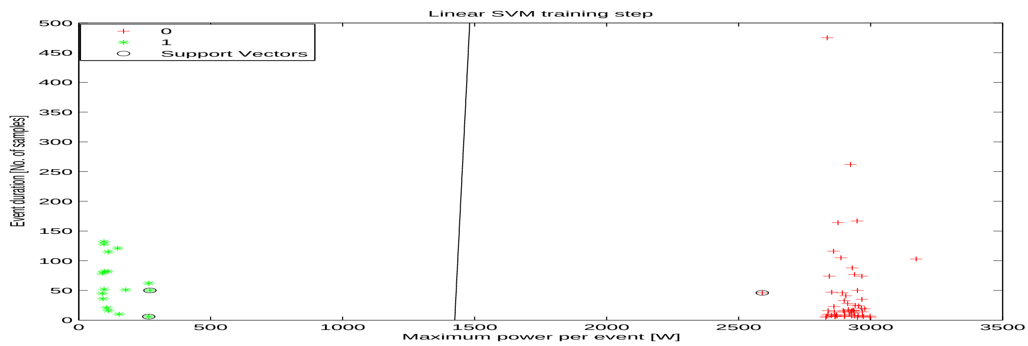
Linear and Gaussian combined algorithms train and test in three steps. First, Trained k-means uses all training and testing points to detect all possible appliances as illustrated in Figures 3.10a, 3.11a. Secondly, we apply a radius on all cluster heads as explained in Figure 3.6 which is fixed to all known appliances within the same house. All training and testing points fall within that threshold will be removed then and the remaining of training and testing points will be fed to Linear or Gaussian SVM classifiers. Finally, multi-class Linear and Gaussian SVMs will be trained and testing



(a) Tumble dryer training



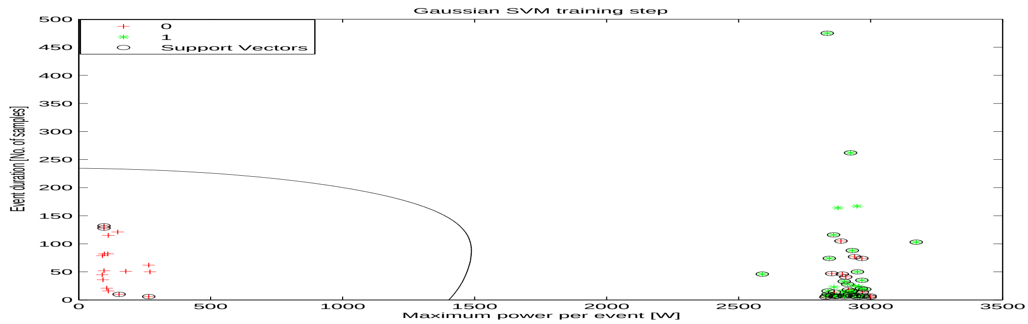
(b) TV training



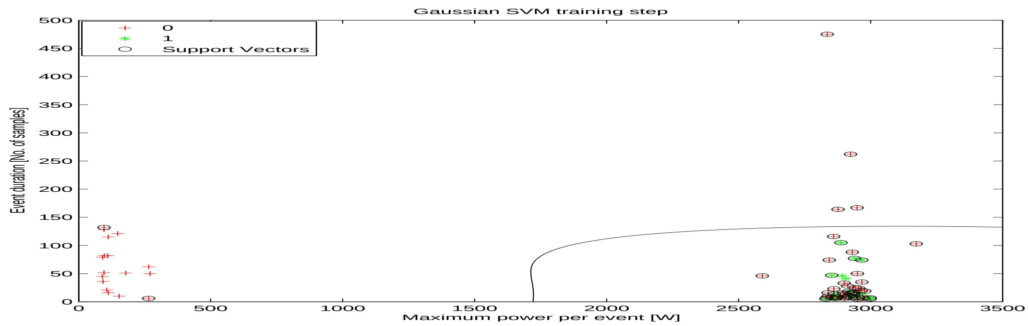
(c) Toaster training

Figure 3.8: Training steps of Linear SVM. Training points in green are the current class, red points are every other appliance, black circles are support vectors used to separate the two classes. x-axis is maximum power value per event. y-axis is event duration [No. of samples].

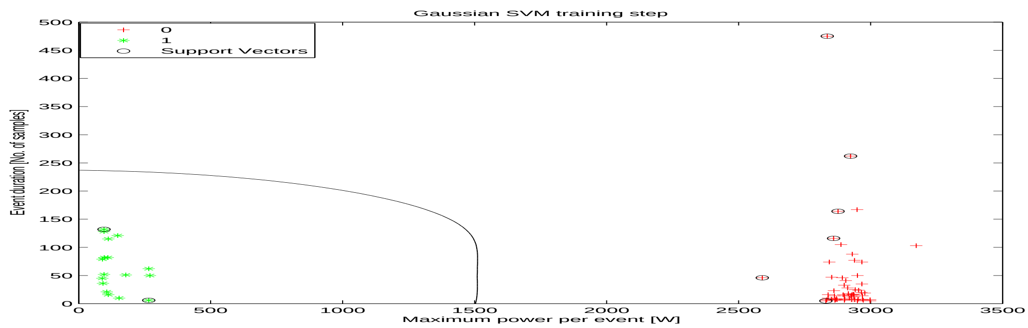
will be performed as explained before which is shown in Figures 3.10b, 3.10c and 3.10d for Linear combined algorithm, and Figures 3.11b, 3.11c and 3.11d for Gaussian combined algorithm. Training points in green are for the current class, red points represent every other appliance, black circles are support vectors used to separate the



(a) Tumble dryer training



(b) TV training



(c) Toaster training

Figure 3.9: Training steps of Gaussian SVM. Training points in green are the current class, red points are every other appliance, black circles are support vectors used to separate the two classes. x-axis is maximum power value per event. y-axis is event duration [No. of samples].

two classes. The first step is to train to detect appliance 1 which is tumble dryer as in Figures 3.10b and 3.11b. The second step is to train to detect appliance 2 which is TV as in Figures 3.10c and 3.11c. The final step is to train to detect appliance 3 which is toaster as in Figures 3.10d and 3.11d.

3.6.5 Disaggregation case study results

The outcome of the classifiers is a one dimensional vector of expected labels, where each label corresponds to the testing event in the same order of their appearance in the testing dataset. After that, we use confusion matrix to compare correct labels that we have generated in the training step with the obtained labels during testing to calculate True Positive, False Positive and False Negative as illustrated in Figure 3.12. Then, we can use our accuracy measures explained in Section 2.6.1.3. We also monitor speed of execution of both training and testing steps separately in terms of seconds as a measure of time complexity. Note that *MATLAB 2013a* was used to generate most of the experiments in this thesis; other simulators might result in a faster or slower performance.

3.6.5.1 Accuracy

From Table 3.1, as it is a simple case study, it can be seen that all algorithms have performed well in detecting tumble dryer and toaster using most of feature combinations but performed poorly in detecting TV due to its training events overlapped with the tumble dryer using all feature combinations.

It can be seen from Figure 3.13 that Trained k-means is significantly faster than all other algorithms in training and testing steps with training execution time of 0.0062 seconds and testing execution time of 0.0931 seconds. Linear and Gaussian SVMs have a very high training execution times with 3.72 and 3.99 seconds respectively. Linear and Gaussian combined algorithms show a lower execution times in both training and testing steps compared to that of Linear and Gaussian SVM algorithms.

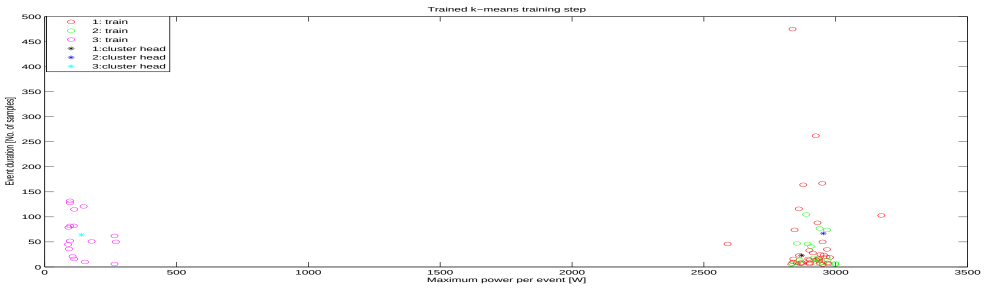
3.7 Summary

In this chapter, we have explained in detail our novel methodology which combines Trained k-means and SVM. Trained k-means is used for its practicality to speed our classifier which filters easily events, the remaining of training and testing points are then classified by Linear and Gaussian SVMs. This method balances between testing time and accuracy which will be demonstrated in the next chapter. Next, we will test

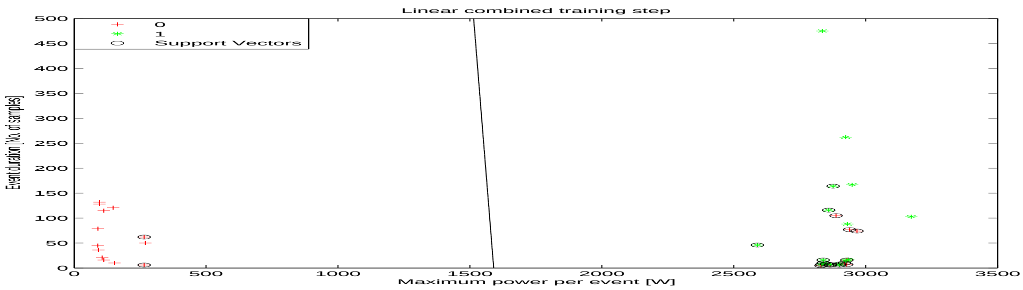
Table 3.1: A comparison between Trained k-means, Linear SVM, Gaussian SVM, Linear combined algorithm and Gaussian combined algorithm using Precision, Recall and F_m after disaggregation using REFIT dataset House 5.

Method		*tumble dryer* (%)	*television* (%)	*toaster* (%)
Trained k-means	Pr	94.64	0	100
	Re	100	0	100
	F_m	97.24	0	100
Linear SVM	Pr	94.64	0.95	100
	Re	100	66.6	100
	F_m	97.24	1.87	100
Gaussian SVM	Pr	94.64	1.92	100
	Re	100	33.3	100
	F_m	97.24	3.63	100
Linear combined	Pr	94.64	0.483	100
	Re	100	33.3	100
	F_m	97.24	0.95	100
Gaussian combined	Pr	94.64	0.47	100
	Re	100	33.3	100
	F_m	97.24	0.93	100

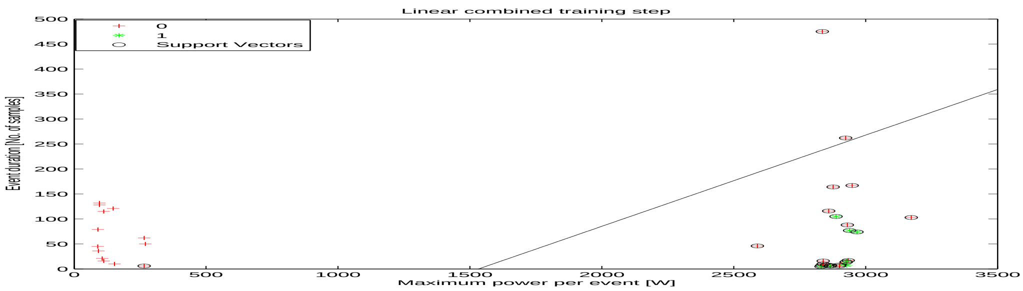
this methodology in supervised and unsupervised approaches and compare it with using only k-means and SVMs separately as well as the state of the art HMM.



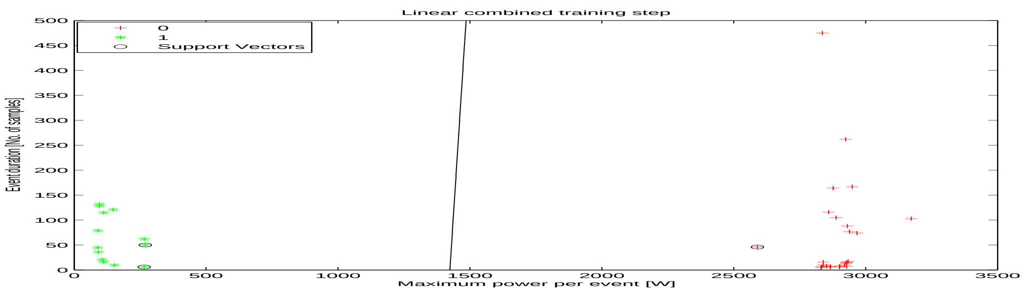
(a) Trained k-means training



(b) Tumble dryer training

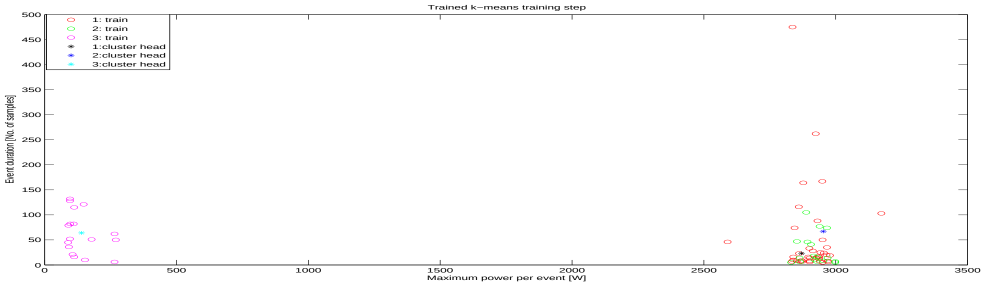


(c) TV training

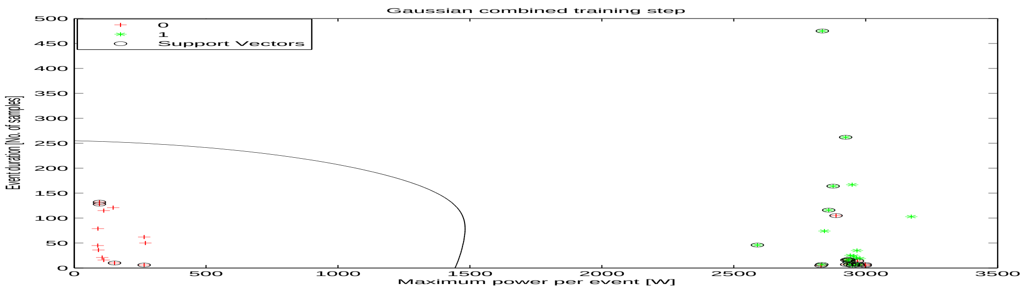


(d) Toaster training

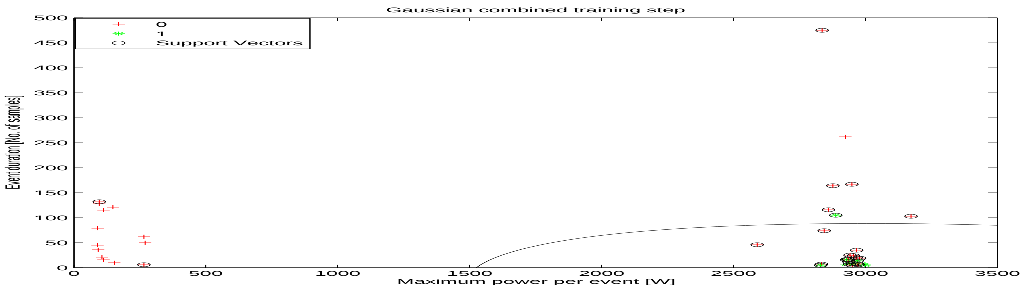
Figure 3.10: Training steps of Linear combined algorithm. Training points in green are the current class, red points are every other appliance, black circles are support vectors used to separate the two classes. x-axis is maximum power value per event. y-axis is event duration [No. of samples].



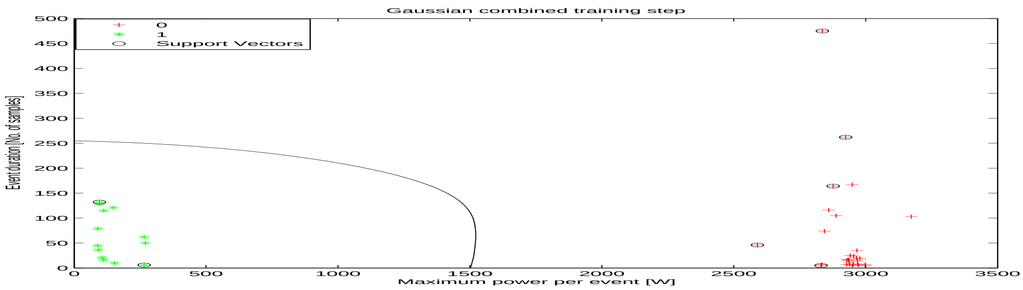
(a) Trained k-means training



(b) Tumble dryer training



(c) TV training



(d) Toaster training

Figure 3.11: Training step of Gaussian combined algorithm. Training points in green are the current class, red points are every other appliance, black circles are support vectors used to separate the two classes. x-axis is maximum power value per event. y-axis is event duration [No. of samples].

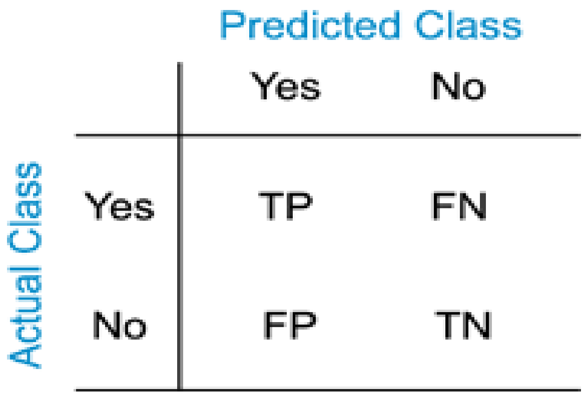


Figure 3.12: True Positive, False Positive and False Negative in the Confusion Matrix form.

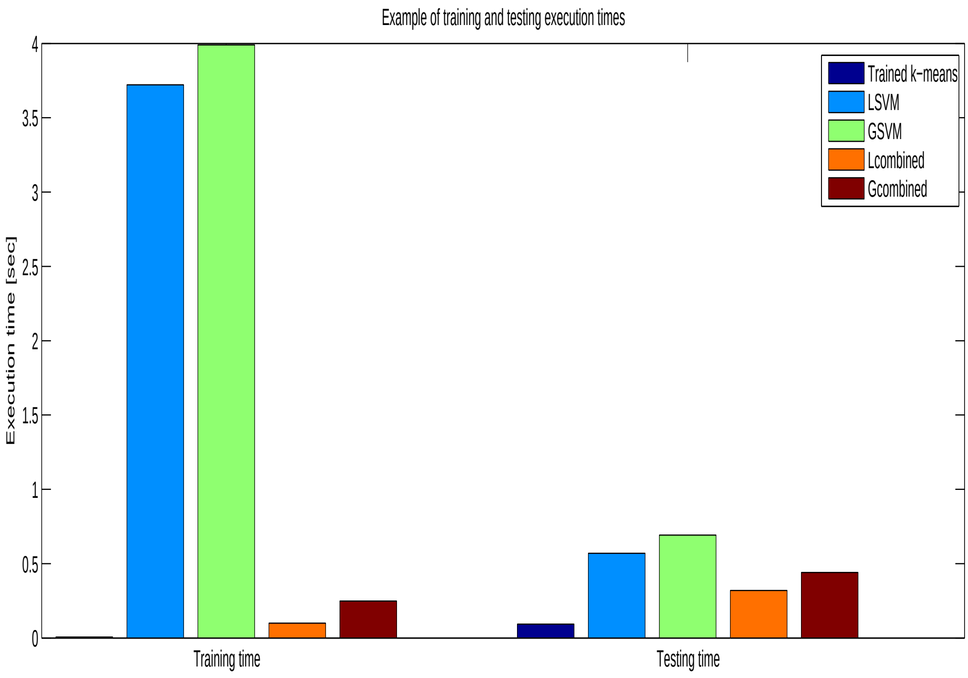


Figure 3.13: Training and Testing execution times by Trained k-means, Linear SVM, Gaussian SVM, Linear combined algorithm and Gaussian combined algorithm using REFIT dataset House 5.

Chapter 4

Supervised Method; Performance and Robustness

4.1 Introduction

In the methodology chapter 3, we have proposed real-time low-complexity combined algorithms to disaggregate total household energy consumption into its individual readings. In this chapter, and based on our published work in [106], we test performance of our Linear and Gaussian combined algorithms using different simple feature combinations; we also test their reliability by reducing training datasets and introducing errors in training label-vectors in comparison with Trained k-means, Linear Support Vector Machines and Gaussian Support Vector Machines. Finally, we benchmark with HMM performance in all experiments.

4.2 Novel Contributions

Novel contributions of this chapter are:

- Novel real-time low-complexity Linear and Gaussian kmeans-SVM-based NALM method for low sampling rate data;
- Innovative appliance-based selection of extracted features that maximize performance;

- Detailed experimental evaluation using different training sizes and errors in labeling the training datasets;
- Performance comparison using three households from the open source database from the USA [39] with 1min sampling rate;

4.3 Performance and Robustness

Testing performance of our Linear and Gaussian SVM-based approaches can be achieved by detecting different types of appliances. The two major metrics that are considered here are accuracy and complexity. Accuracy is measured by the number of correctly detected appliance events as explained in Section 2.6.1.3; the complexity is measured by execution time (in seconds) for training and testing simulations for classifications form start to finish. Linear combined and Gaussian combined performances are compared with Trained k-means, Linear SVM and Gaussian SVM. All algorithms are tested with the same training and testing periods.

To assess the testing robustness of the algorithms we test how reliably an algorithm adapts to real life issues such as missed data and labeling errors. One way of testing robustness is by reducing training datasets. A proper amount of training events of all known appliances is needed to be present in the training sets. Labeling appliances for creating public datasets, is sometimes done manually by occupiers of tested houses, which can result in labeling errors. Therefore, another way to test robustness is to insert errors in the training labeling step. These errors are added randomly, that is, by placing wrong labels in random positions in our training label-vector by wrong labels.

4.4 Results and Discussion

We use House 1, House 2 and House 6 from the publicly available REDD dataset [39] down-sampled to 1min resolution. The training size was varied in the experiments, and testing is always performed on four weeks worth of data.

All experiments were run on an HP Pavilion 15 Notebook PC with 8GB RAM, 1TB Hard drive and AMD A10 with 2.2 GHz Radeon HD dual Graphics processor (quad core) using *MATLAB 2013a*.

4.4.1 Feature Selection

Using different feature combinations has a significant impact on the performance of our approaches. Table 4.1 shows few selected feature combinations of 2-dimensional, 3-dimensional , 4-dimensional combinations and all features (5-dimensional) combination. The same event detection thresholds and training set were used here for all simulations. Marked with bold typeface are the best performing features for each appliance. One can see that the algorithms respond differently to different types of features. For example, *Max power, Min power and Max/Mean* were better for Gaussian combined algorithm in Houses 1 and 2 with total F-Measure of 75% and 86.30% respectively, but were outperformed by using *Max power and Min power* only in House 6 with 97%.

Likewise, *Area and Duration* gave worse results for Linear combined algorithm in Houses 1 and 2 with total F-Measure 66.45%. Yet, in House 6 it has better results of 91.47%. Interestingly, feature combination has the same impact on Trained k-means and Linear and Gaussian SVMs which can be seen in Appendix A were same feature combinations can affect classifiers differently.

We use all possible feature combinations to detect known appliances using all tested algorithms, and the best result is then chosen. Tables 4.2, 4.3 and 4.4 show the best feature combinations used to distinguish between known appliances in REDD dataset Houses 1, 2 and 6 respectively. It can be noticed from all tables that *maximum power value* often help detect refrigerators correctly in all three houses. The remaining appliances respond better to *area* and *maximum divided by minimum power ratio*. However, in practical, area slows down classifiers when used to detect different appliances. It can also be noticed from the three tables that higher consumer appliances *(washer dryer and dishwasher)* are often distinguishable using *duration of events* due to their long operation cycles. However, 2 dimensional and 3 dimensional feature combinations give better performance by all tested algorithms compared to 4 dimensional and 5 dimensional, mostly, in all appliances, this can simply mean that the higher dimensional datasets are harder to disaggregate by the classifiers.

Table 4.1: Comparison between selected features using F-Measure for REDD data Houses. Max= maximum power value. Min= minimum power value. dur= duration of an event. ratio= maximum power value over mean power value ratio.

House number	*Feature combination*	*Linear Combined* (%)	*Gaussian Combined* (%)
	Max. & Min.	70.72	69.46
House 1	*Area. & Dur.*	66.45	48.57
	Max.,Dur. & Area	69.14	67.08
	Max.,Min. & ratio	70.88	**75**
	Min.,Dur.,Area & ratio	**71.83**	59.96
	All features	69.4	60.28
	Max. & Min.	83.04	83.66
House 2	*Area. & Dur.*	66.45	78.51
	Max.,Dur. & Area	85.42	86.18
	Max.,Min. & ratio	83.41	**86.30**
	Min.,Dur.,Area & ratio	**85.67**	67.71
	All features	83.16	85.80
	Max. & Min.	86.99	**97**
House 6	*Area. & Dur.*	**91.47**	86.78
	Max.,Dur. & Area	91.25	96.37
	Max.,Min. & ratio	90.61	96.37
	Min.,Dur.,Area & ratio	89.97	95.52
	All features	87.84	95.94

4.4.2 Algorithms Performance

The three REDD dataset houses used are House 1, 2 and 6; we chose five appliances in each house. In House 1, we train the algorithms with five known appliances: refrigerator, microwave, toaster, dishwasher, washer dryer. In House 2, we trained the algorithms with: refrigerator, stove, microwave, toaster, and dishwasher. In House 6, we have: refrigerator, stove, microwave, toaster, air conditioner. All remaining appliances were considered to be ”unknown” and hence they contribute to noise. Unknown appliances were neglected due to changing their signature or being combined with other appliances in the same monitor.

Table 4.2: Feature combinations used in detecting all appliances for REDD data House 1. NA= Not Available. Max= maximum power value. Min= minimum power value. dur= duration of an event. ratio= maximum power value over mean power value ratio.

Method	*Refrigerator*	*Microwave*	*Toaster*	*Dishwasher*	*Washer Dryer*
Trained k-means	max & dur	max, min & ratio	max, min & ratio	max & area	max & ratio
Linear SVM	max & area	max, min & ratio	dur & ratio	max & ratio	max, min ,dur & ratio
Gaussian SVM	max, min & ratio	max, & min	area & dur	max, min area & ratio	max & dur
L Combined	max & area	max, min & ratio	all features	min & ratio	max & dur
G Combined	max, min & ratio	max & dur	max, min area & dur	max & dur	max, min & area

Table 4.3: Feature combinations used in detecting all appliances for REDD data House 2. NA= Not Available. Max= maximum power value. Min= minimum power value. dur= duration of an event. ratio= maximum power value over mean power value ratio.

Method	*Refrigerator*	*Stove*	*Microwave*	*Toaster*	*Dishwasher*
Trained k-means	max & dur	NA	NA	max & dur	NA
Linear SVM	max & area	all features	NA	max & area	max, dur & area
Gaussian SVM	max, min & area	max & ratio	area & ratio	max & area	area & min
L Combined	max & dur	area & min	area & dur	max, dur area & ratio	max dur area
G Combined	max & dur	NA	area & dur	max & dur	min, dur & area

Table 4.4: Feature combinations used in detecting all appliances for REDD data House 6. NA= Not Available. Max= maximum power value. Min= minimum power value. dur= duration of an event. ratio= maximum power value over mean power value ratio.

Method	*Refrigerator*	*Stove*	*Microwave*	*Toaster*	*Air conditioner*
Trained k-means	max & dur	NA	NA	NA	max & dur
Linear SVM	max & area	max & area	dur area & ratio	NA	max, area & ratio
Gaussian SVM	max & min	max & ratio	dur & ratio	NA	max & min
L Combined	max & ratio	max & ratio	max, min & ratio	NA	max & ratio
G Combined	max, min area & dur	max & dur	max, min & dur	NA	max, min area & dur

4.4.2.1 Time Complexity

Table 4.5 and Figure 4.1 show training time and testing time results in seconds obtained for REDD dataset houses 1, 2 and 6 for Trained k-means, Linear SVM, Gaussian SVM, Linear combined algorithm and Gaussian combined algorithm. Execution time is measured after the data processing step which indicates only the time for algorithms to run training and testing process. It is obvious that trained k-means has the shortest execution time that is less than 0.3 sec in both training and testing in all three houses. Linear SVM has the highest training times of 0.945, 1.198 and 0.698 seconds respectively, while Gaussian SVM has the highest testing times of 1.283, 1.523 and 0.82 seconds respectively. It can be seen that Linear and Gaussian combined algorithms vary between Trained k-means and SVMs which are faster than SVMs but slower than Trained k-means. Next we test accuracy for our proposed combined algorithms in comparison with Trained k-means, Linear support vector machines and Gaussian vector machines.

Table 4.5: Comparison between Trained kmeans, Linear SVM, Gaussian SVM, Linear combined algorithm and Gaussian combined algorithm using Execution time for REDD data Houses 1, 2 and 6.

Method	House 1		House 2		House 6	
	Train (sec)	*Test* (sec)	*Train* (sec)	*Test* (sec)	*Train* (sec)	*Test* (sec)
Trained k-means	0.155	0.152	0.291	0.291	0.239	0.139
Linear SVM	0.945	0.723	1.198	0.803	0.698	0.562
Gaussian SVM	0.729	1.283	0.801	1.523	0.579	0.82
Linear Combined	0.298	0.388	0.352	0.553	0.301	0.334
Gaussian Combined	0.484	0.442	0.42	0.488	0.47	0.477

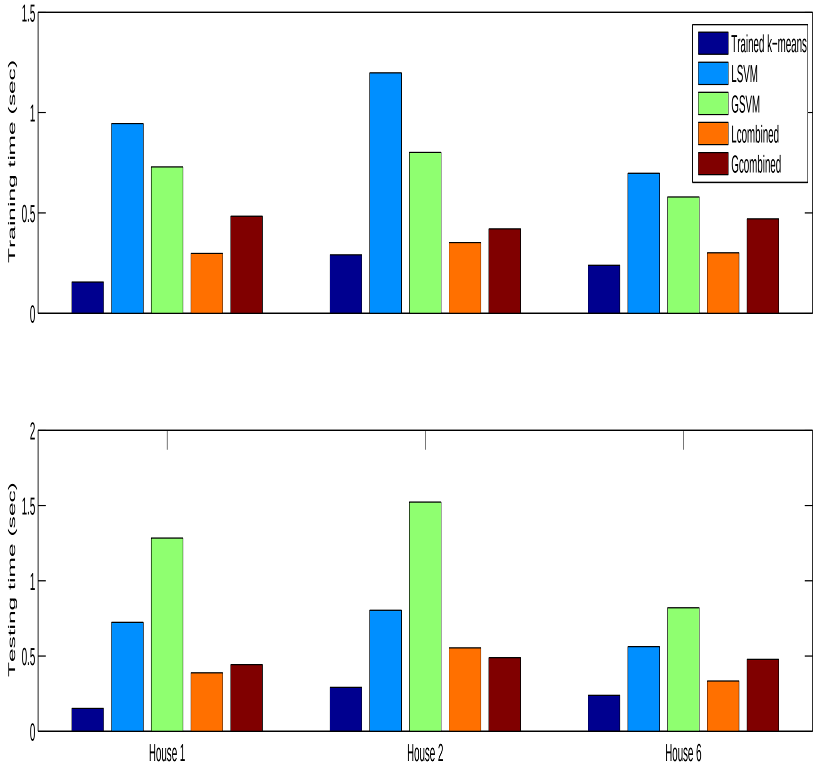


Figure 4.1: Upper figure shows training execution times in seconds and lower figure shows testing execution times after disaggregation by Trained k-means, Linear and Gaussian SVMs and Linear and Gaussian combined method for Houses 1,2 and 6 in REDD dataset.

4.4.2.2 Accuracy

We have used the evaluation metrics of precision (PR), recall (RE) and F-Measure (FM) for each appliance explained in Equations 2.4, 2.5 and 2.6 respectively. We also use total precision , total recall and total F-Measure measurements that use total events such as total True positive events from all known appliances in each tested house which are defined as :

$$TotalPR = \sum_{n=1}^{N} TP_n / (\sum_{n=1}^{N} TP_n + \sum_{n=1}^{N} FP_n) \tag{4.1}$$

$$TotalRE = \sum_{n=1}^{N} TP_n / (\sum_{n=1}^{N} TP_n + \sum_{n=1}^{N} FN_n) \tag{4.2}$$

$$TotalF_M = 2 * ((totalPR) * (totalRE))/((totalPR) + (totalRE)), \tag{4.3}$$

where N is the total number of known appliances in a tested house. TP_n, FP_n and FN_n are True positive events, false positive events and false negative events per appliance.

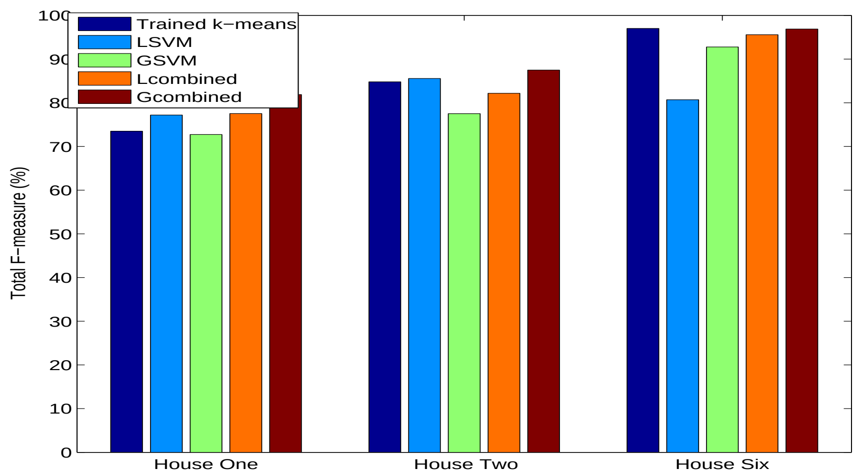


Figure 4.2: Total F-Measure after disaggregation by Trained k-means, Linear and Gaussian SVMs and Linear and Gaussian combined method for Houses 1, 2 and 6 in REDD dataset.

Figure 4.2 shows total F-Measure after disaggregation of REDD dataset houses 1,2 and 6 using Trained k-means, Linear and Gaussian SVMs and Linear and Gaussian combined algorithms. The best result of each known appliance from all feature combinations was chosen as discussed in Section 4.4.1 that were tested. All five algorithms always use the same edge detection and feature extraction method explained previously.

All algorithms show high total F-Measure of above 70% in all tested houses. In Houses 1 and 2, Gaussian Combined algorithm has the best total F-Measure among other algorithms with 81.88% and 87.49% respectively but is slightly worse than Trained k-means in House 6 with total F-Measure of 96.87%. Linear SVM was performed 3% and 8% better than Gaussian SVM in Houses 1 and 2 respectively but 12% less in House 6. Trained k-means accomplished the highest total F-Measure compared to other algorithms in House 6 with 97.01% and a good performance on Houses 1 and 2 with 73.49% and 84.79% respectively. However, total F-Measure gives false impression that Trained k-means can behave better than SVMs and combined algorithms. Next, we look into each house in detail.

Figures 4.3 to 4.6 show a comparison between Trained k-means, Linear SVM, Gaussian SVM, Linear combined algorithm and Gaussian combined algorithm in REDD Houses 1, 2 and 6 respectively. It can be seen that all algorithms performed well in detecting refrigerator as expected due to its high number of training and testing events. Trained k-means has lower F-Measure as it was effected by high False Positive of 132 events compared to 82, 91, 80 and 72 events by Linear SVM, Gaussian SVM, Linear combined and Gaussian combined respectively (see Figure 4.4). In dishwasher results, Trained k-means has a top 100% Precision but much lower Recall and F-Measure which can be explained by looking at Figure 4.4, as it can be seen that Trained k-means has a low detection rate of dishwasher but zero False Positive events gave wrong impression. By looking at toaster accuracy measures, it can be seen that Gaussian combined algorithm gave 100% Precision because of its zero False Positive but its F-Measure result fell behind Trained k-means as it has higher False Negative of 25 events compared to 3 only.

By looking at Figure 4.5 for House 2 results, it can be seen that all algorithms performed well in detecting refrigerator with Precision and Recall of above 91% and F-Measure of 92.58% or better. Gaussian combined algorithm did not manage to correctly disaggregate any stove events but Linear combined F-Measure outperformed all other algorithms in detecting stove with Precision of 66.66%, Recall of 33.33% and F-Measure of 44.44% using proper feature combination. The proposed combined algorithms have clearly exceeded other algorithms in classifying microwave and toaster. Interestingly, Linear combined algorithm had a similar performance with F-Measure 29.26% to that by Linear SVM with F-Measure 26.06% in detecting dishwasher and Gaussian combined had same performance by Gaussian SVM with Precision of 50%,

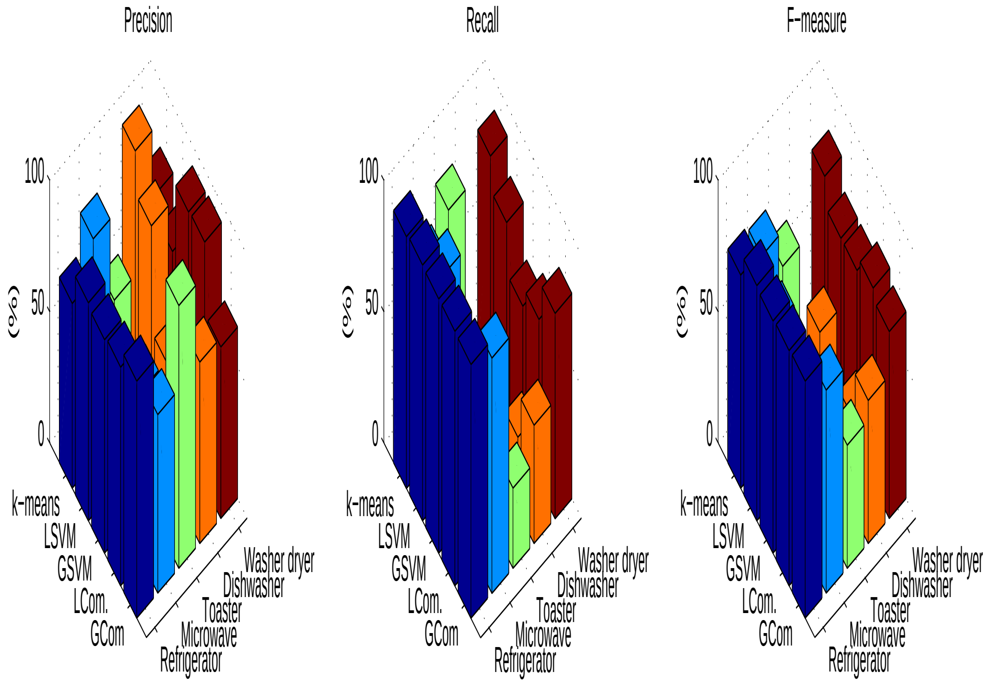


Figure 4.3: Precision, Recall and F-Measure for each appliance, after disaggregation by Trained k-means, Linear and Gaussian SVMs and Linear and Gaussian combined method for House One in REDD dataset.

Recall of 21.42% and F-Measure 30%. Trained k-means, on the other hand, fell behind in detecting all tested appliances as expected except for refrigerator which has highest number of events due to its operating nature.

Figure 4.6 shows a comparison between all five tested algorithms in House 6. Refrigerator and air conditioner have the highest number of events in the training set and 405 and 58 events respectively in the testing set, which made all algorithms have a significantly high Precision, Recall and F-Measure results. 90% or above for refrigerator for Precision, Recall and F-Measure. Linear SVM had slightly lower F-Measure result with only 30% when identifying air conditioner compared to above 87% roughly by other tested algorithms due to its high FP number from refrigerator events using *maximum power, area and max/mean ratio*. Trained k-means was not able to distinguish correctly any event for stove, microwave and toaster which have

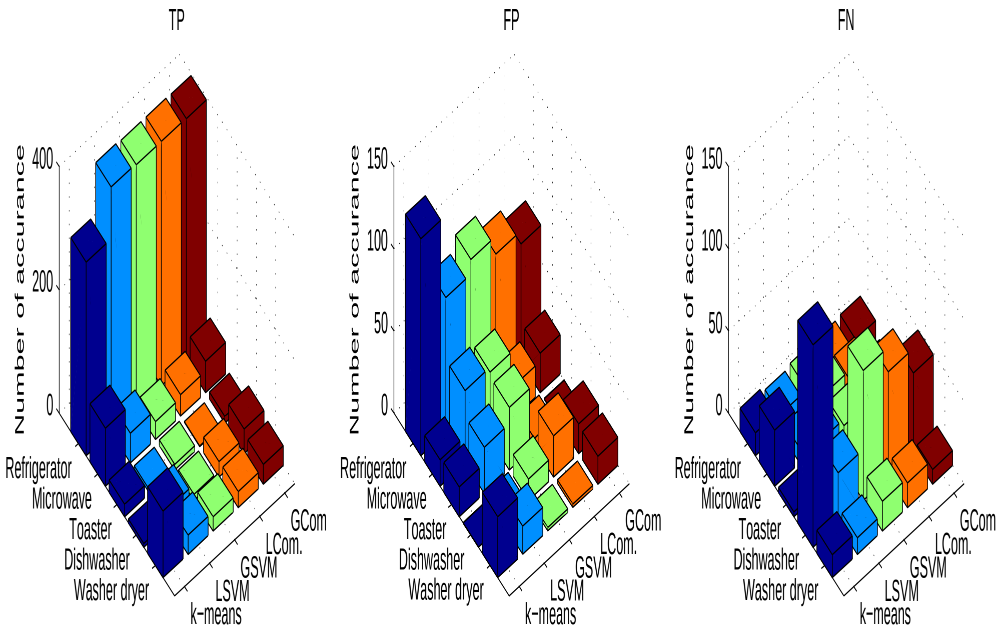


Figure 4.4: True Positive, False Positive and False Negative values for each appliance, after disaggregation by Trained k-means, Linear and Gaussian SVMs and Linear and Gaussian combined method for House One in REDD dataset.

much less non-overlapped events in the training set with less than 10 testing events each. In fact, all five algorithms failed to detect toaster correctly using all possible feature combinations.

4.4.2.3 Performance summary

It is clear that Linear and Gaussian combined algorithms have provided a competitive performance compared to other tested algorithms in terms of complexity and accuracy in detecting most of present appliances in REDD dataset Houses 1, 2 and 6. Both combined algorithms provided a trade-off between speed of processing and quality of disaggregation using different sets of features that simplifies data and maximizes

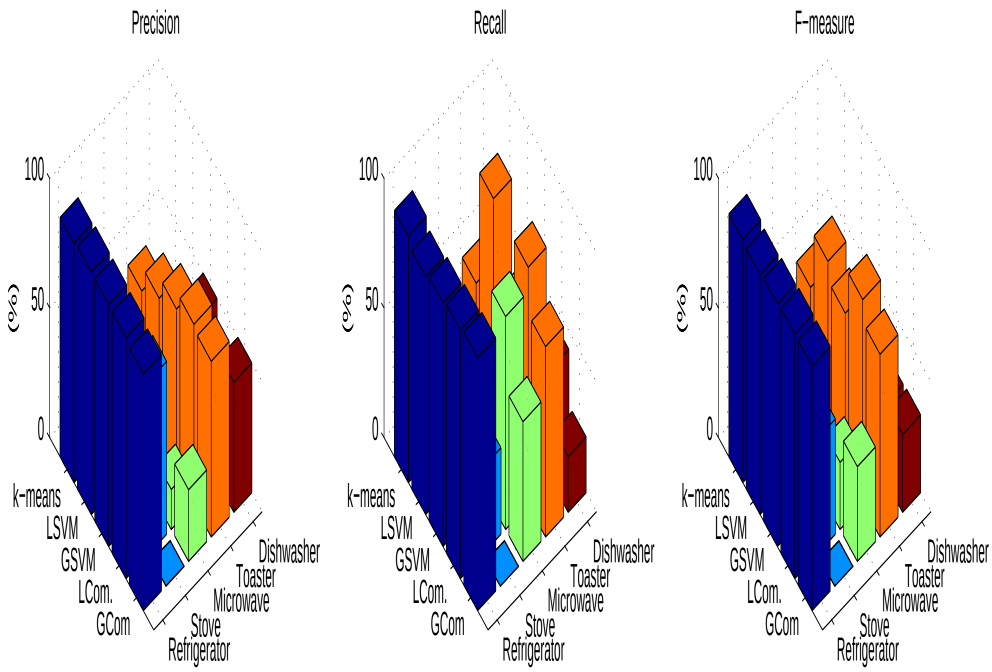


Figure 4.5: Precision, Recall and F-Measure for each appliance, after disaggregation by Trained k-means, Linear and Gaussian SVMs and Linear and Gaussian combined method for House Two in REDD dataset.

performance. Next, we test robustness of our approaches along with Trained k-means, LSVM and GSVM.

4.4.3 Algorithms Robustness

In this section, we test the robustness of the proposed methods to the reduction of the training set size. We test different training sizes: 6000 samples (roughly 6 days), 5000 samples (roughly 5 days), 4000 samples (roughly 4 days), 3000 samples (roughly 3 days) and 2000 samples (roughly 2 days). We also introduced errors in label-vector starting with 5% error rate then increase gradually until 20% error rate.

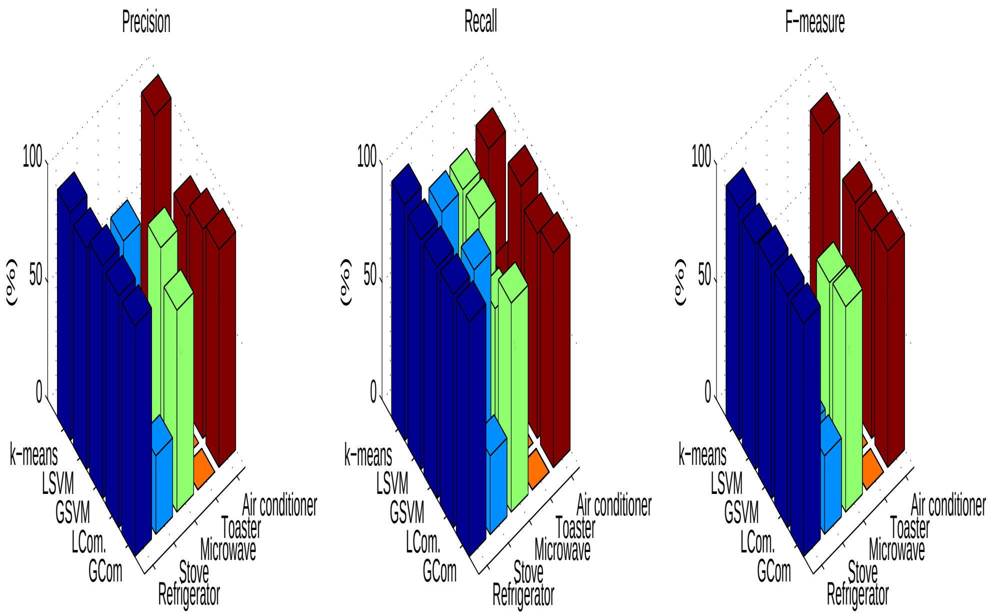


Figure 4.6: Precision, Recall and F-Measure for each appliance, after disaggregation by Trained k-means, Linear and Gaussian SVMs and Linear and Gaussian combined method for House Six in REDD dataset.

4.4.3.1 Reduction of training-set size

Figures 4.7 to 4.10 show results while reducing different training sizes for all five algorithms. Figure 4.7 shows results of testing step execution times after reducing training dataset size. Disaggregation is achieved by choosing the best results after testing all possible feature combinations. Therefore, testing execution times are averaged per house. It can be noticed here that Trained k-means is fairly steady in response to training different sizes as it takes roughly 0.153, 0.266 and 0.136 seconds on average to test all existing appliances in Houses 1, 2 and 6 respectively. Linear SVM has a response time of roughly 0.688 second in testing Houses 1 and 2. Thus, it decreases slowly in testing House 6 from 0.5 to 0.39 seconds. Gaussian SVM has the highest testing execution time that exceeds 1 sec on average among all other algorithms , and is also the most sensitive approach which dropped in House 1 experiments from 1.2

at 5000 samples to 1 second at 4000 and dropped by half at 2000 samples with 0.677 second. In House 6, GSVM had execution time of 0.876 and 0.938 seconds at 6000 and 5000 samples respectively which then decreased to 0.54 second roughly at 4000, 3000 and 2000 samples. On the other hand, Linear and Gaussian combined algorithms show a very steady decrease in testing running times after decreasing training datasets in all REDD houses. The exact results in detail are available in Appendix A.

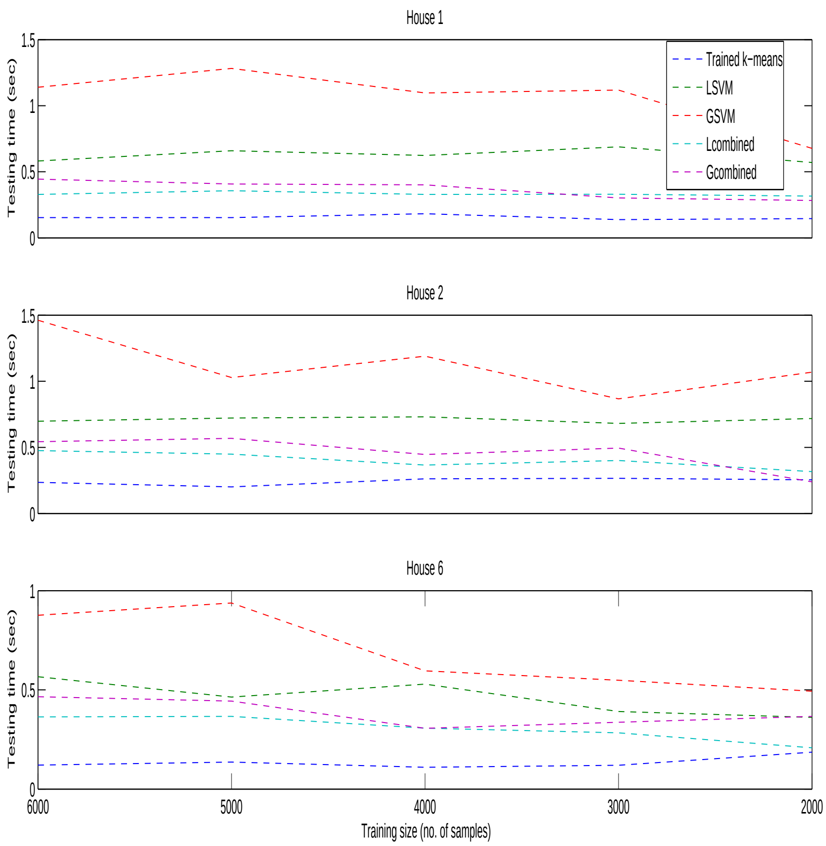


Figure 4.7: Testing time of reduced training size after disaggregation of each appliance by Trained k-means, Linear and Gaussian SVMs and Linear and Gaussian combined methods for Houses 1, 2 and 6 in the REDD dataset.

Figure 4.8 shows F-Measure results of reducing training size using House 1 after disaggregation of each appliance by all five algorithms. Refrigerator shows very steady performance by all tested algorithms even with decreased training due to its significantly hight number of events in training dataset. All five algorithms managed to classify microwave with above 63% F-Measure roughly. Though, Linear combined had a lower result at 5000 and 3000 samples with 20% less F-Measure. Gaussian combined also had a lower F-Measure of 20% approximately at 5000 samples. Toaster had a zero value by all algorithms in all different training sizes due to its sensitivity. Trained k-means was not successful in detecting dishwasher. Linear SVM and Gaussian combined algorithms have better performance at 4000 samples compared to other training sizes. Gaussian SVM and Linear combined algorithms have slightly better performance than other three algorithms in detecting dishwasher with 20% to 40% F-Measure each.

Washer dryer was hard to be correctly disaggregated by Trained k-means, Linear SVM and Linear combined algorithms. Gaussian combined had a slightly lower F-Measure than Gaussian SVM. However, all five algorithms were quite robust in detecting refrigerator and microwave. Linear SVM, Gaussian SVM, Linear combined and Gaussian combined algorithm were robust in detecting dishwasher and washer dryer. All algorithms were more sensitive in detecting toaster. Note that appliances can have better performance with lower training due to better quality of training.

House 2 F-Measure results of reducing training size after disaggregation of each appliance by all five algorithms are present in Figure 4.9. Refrigerator showed very steady performance by all tested algorithms with above 90% F-Measure approximately. Stove and microwave detection were not as successful which can be seen since only Gaussian Combined had 40% F-Measure at 6000 samples whereas rest of other algorithms gave zero or close to zero values when different training sizes were tested. Toaster results varied in F-Measure by our tested algorithms from 45% to 78%. Linear combined algorithm accomplished better results at 3000 and 2000.

Figure 4.10 shows F-Measure results of when reducing the training size after disaggregation of each appliance by all five algorithms in REDD dataset House 6. Refrigerator and air conditioner were not affected much by reducing training set size due to their significant presence in training; other appliances were not as steady as they have much less data in the training set. Looking at air conditioner results, it can be seen that Trained k-means had a major drop in performance at 4000 samples with 18.18% F-Measure, zero value at 2000 samples and 86% F-Measure roughly with all other training sizes. Linear SVM had a better result at 4000 samples compared to other training sizes with 73.78% F-Measure. Gaussian SVM had firmly fixed performance for up to 2000 samples with F-Measure between 71% to 87.7%. Linear combined algorithm gave better results at 3000 and 4000 samples with 87.7% and

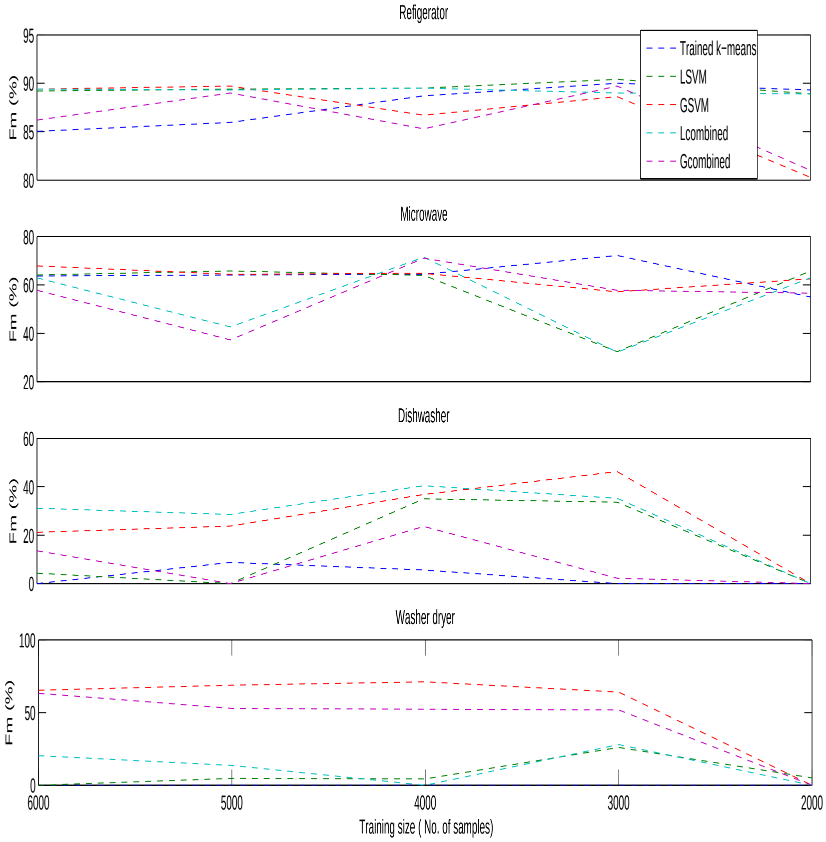


Figure 4.8: F-Measure of reduced training size after disaggregation of each appliance by k-means, Linear and Gaussian SVMs and Linear and Gaussian combined method for House 1 in the REDD dataset.

88.8% respectively. Gaussian combined algorithm had a better F-Measure at 5000 and 4000 samples with 92.45% and 82.14% respectively.

4.4.3.2 Insertion of labeling errors

Figures 4.11 to 4.13 show another technique of testing robustness of an algorithm by inserting wrong labels in the training label-vector as every training dataset has a corresponding vector of labels to help separating classes.

Figure 4.11 shows F-Measure results of error insertion using House 1 after disaggregation of each appliance by all five algorithms. Refrigerator, microwave and washer dryer performance were very robust to different error rates due to their proper training events in the training dataset. All five algorithms managed to detect refrigerator

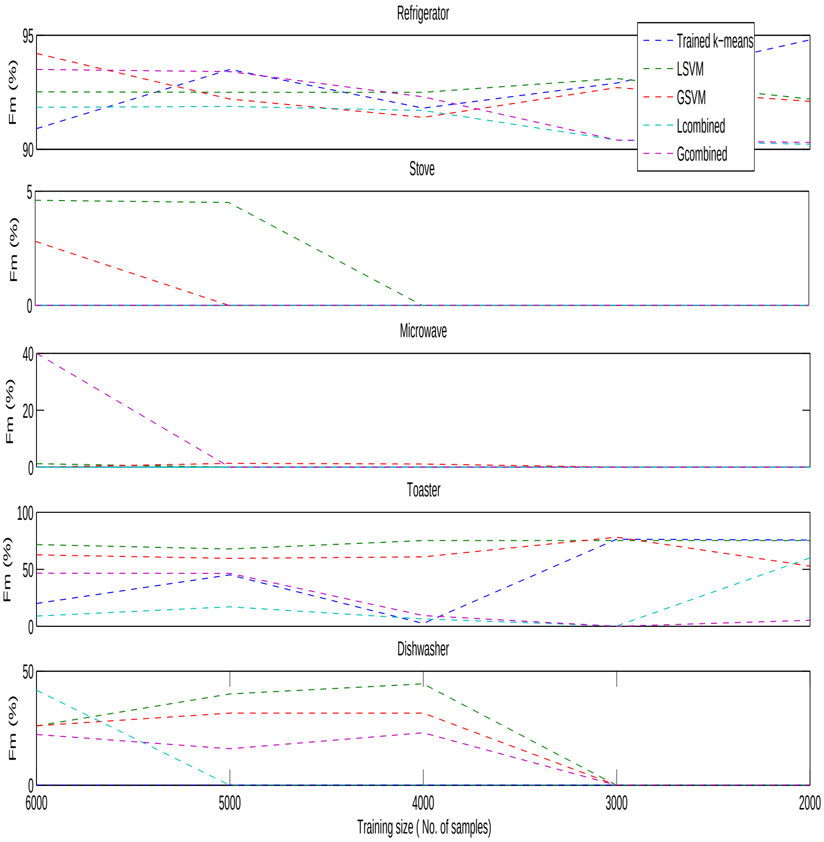


Figure 4.9: F-Measure of reduced training size after disaggregation of each appliance by k-means, Linear and Gaussian SVMs and Linear and Gaussian combined method for House 2 in the REDD dataset.

clearly with F-Measure that is up to 90% roughly. Although, microwave was a challenge for Trained k-means as its performance dropped dramatically at 15% error rate almost half of F-Measure value. All tested algorithms show very poor performance in finding toaster and dishwasher. Trained k-means, Linear SVM, Linear and Gaussian combined algorithms have zero or close to zero value with toaster. Thus, Gaussian SVM gave a slightly better F-Measure value with roughly 36% at 5% and 15% error rates. In the case of detecting washer dryer, Linear combined gave a high and very steady performance accomplishing roughly 80% F-Measure compared to a top of 39%, 65%, 69% and 34% with Trained k-means, linear SVM, Gaussian SVM and Gaussian combined algorithms respectively. Note that appliances are affected by each other in the training step, which might result in higher F-Measure results for some appliances corresponding to higher error rates due to errors being introduced randomly.

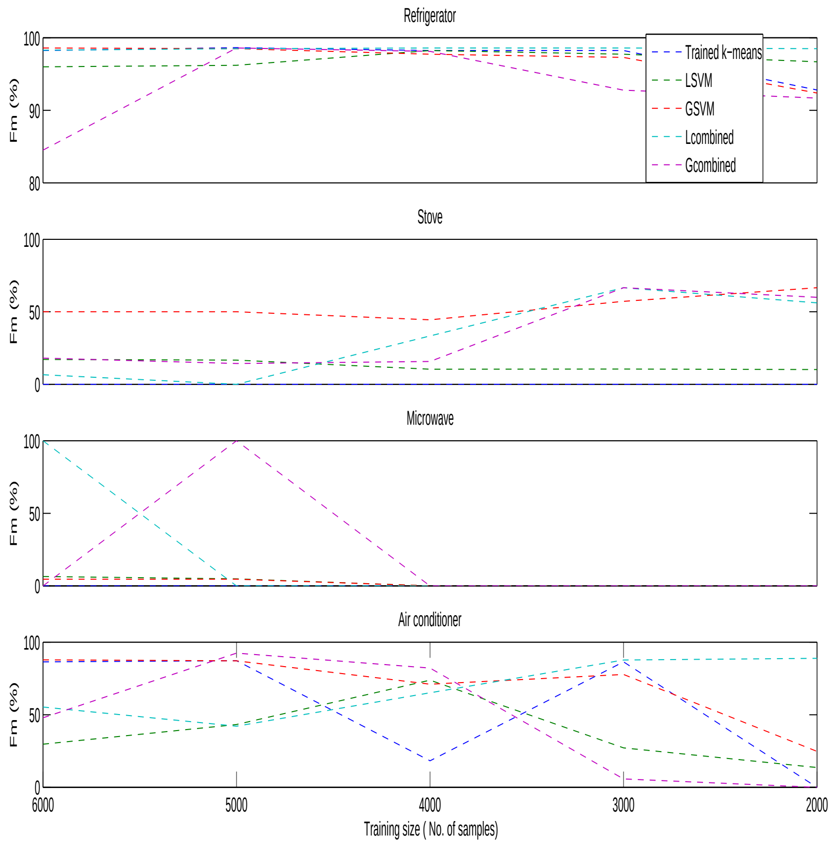


Figure 4.10: F-Measure of reduced training size after disaggregation of each appliance by k-means, Linear and Gaussian SVMs and Linear and Gaussian combined method for House 6 in the REDD dataset.

Figure 4.12 shows F-Measure results of error insertion using House 2 after disaggregation of each appliance by all five algorithms. Refrigerator show very steady performance by all tested algorithms as expected. Detecting stove and microwave were not as successful, since all algorithms show very poor performance that is close to zero to disaggregate stove events. Linear and Gaussian combined algorithms outperformed other algorithms to correctly distinguish microwave events with up to 51% roughly. Washer dryer was hard to be correctly disaggregated by Trained k-means, Gaussian SVM and Gaussian combined algorithms. Thus, Linear SVM and Linear combined have a slightly better F-Measure values than other algorithms with roughly 45% each at 5% and 10% error rates for Linear SVM and at 15% and 20% error rates.

Figure 4.13 shows F-Measure results of error insertion using House 6 after disaggregation of each appliance by all five algorithms. Refrigerator and air conditioner

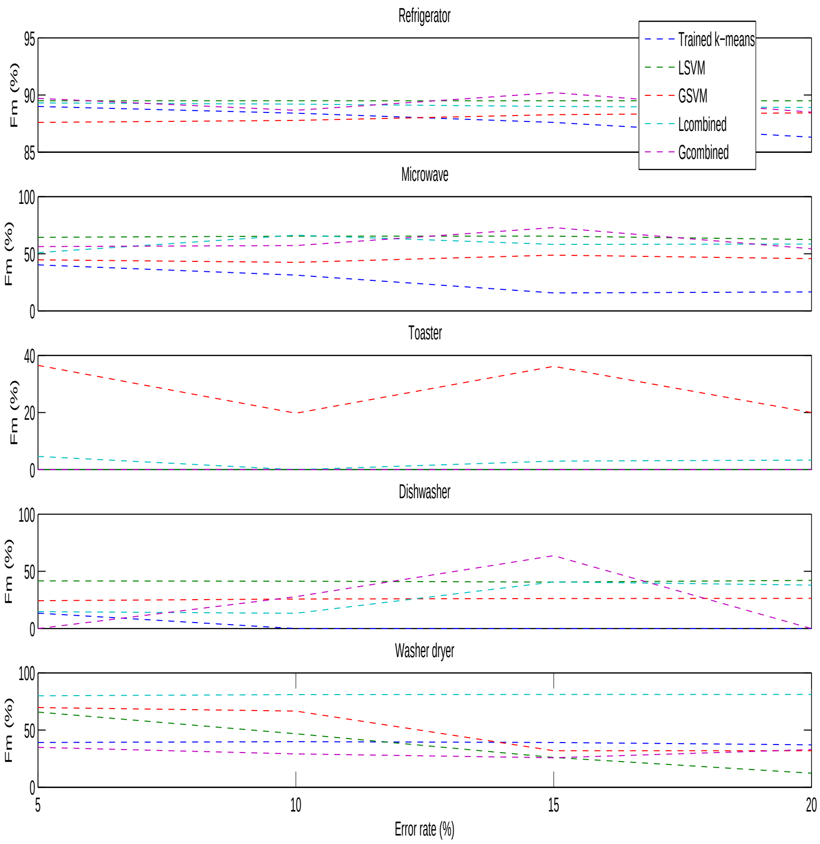


Figure 4.11: F-Measure of increased error rate after disaggregation of each appliance by k-means, Linear and Gaussian SVMs and Linear and Gaussian combined method for House 1 in the REDD dataset.

have quite robust performance in all algorithms due to their high presence in training datasets even with high error rate in label-vector. Stove, microwave and toaster have significantly lower performance as the amount of their training events are less than ten testing non overlapped events each. In fact, none of our tested algorithms have correctly detected any toaster events due to its short operating cycle in training and testing steps. However, Linear and Gaussian combined algorithms accomplished better results with microwave with up to 100% F-Measure at 10% error rate using Linear combined and a steady 66.66% F-Measure with Gaussian combined algorithm.

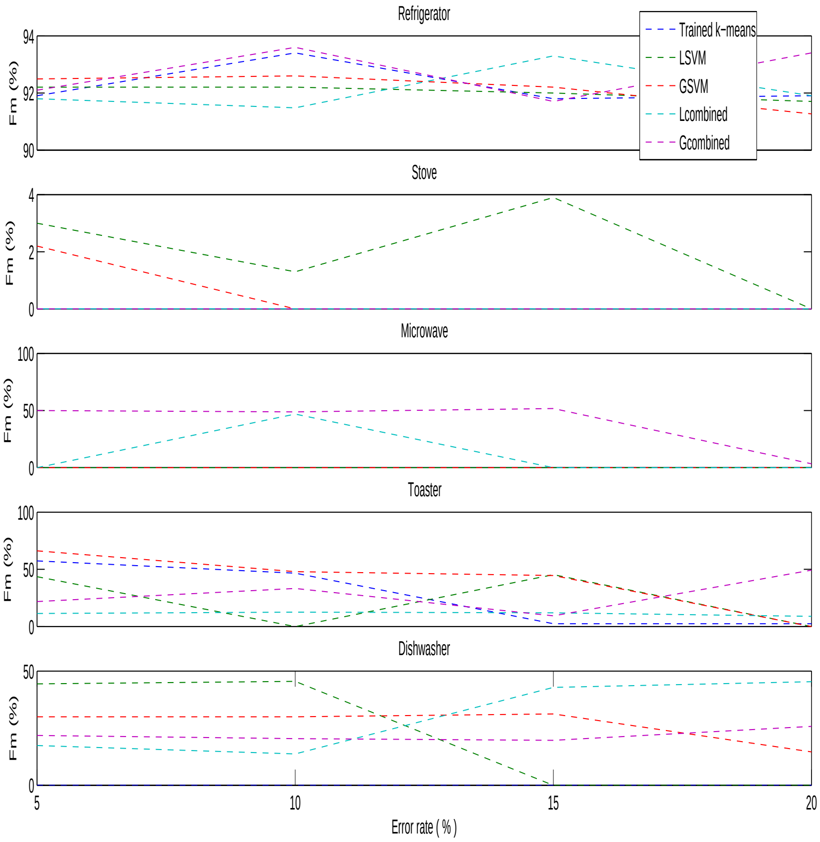


Figure 4.12: F-Measure of increased error rate after disaggregation of each appliance by k-means, Linear and Gaussian SVMs and Linear and Gaussian combined method for House 2 in the REDD dataset.

4.4.3.3 Robustness summary

The Combined algorithms are not sensitive to the variation of the training size nor increased error rate. Note that the SVM-based and proposed method can have slightly better performance for smaller training sets due to better quality of the training data. And it can also give better F-Measure results with higher error rate as some appliances are more sensitive to errors than others. In average over all three houses, the combined methods outperform all other approaches for training sizes of 2000 and 6000 in most cases and comes after Trained k-means for the training size of 5000 and 3000.

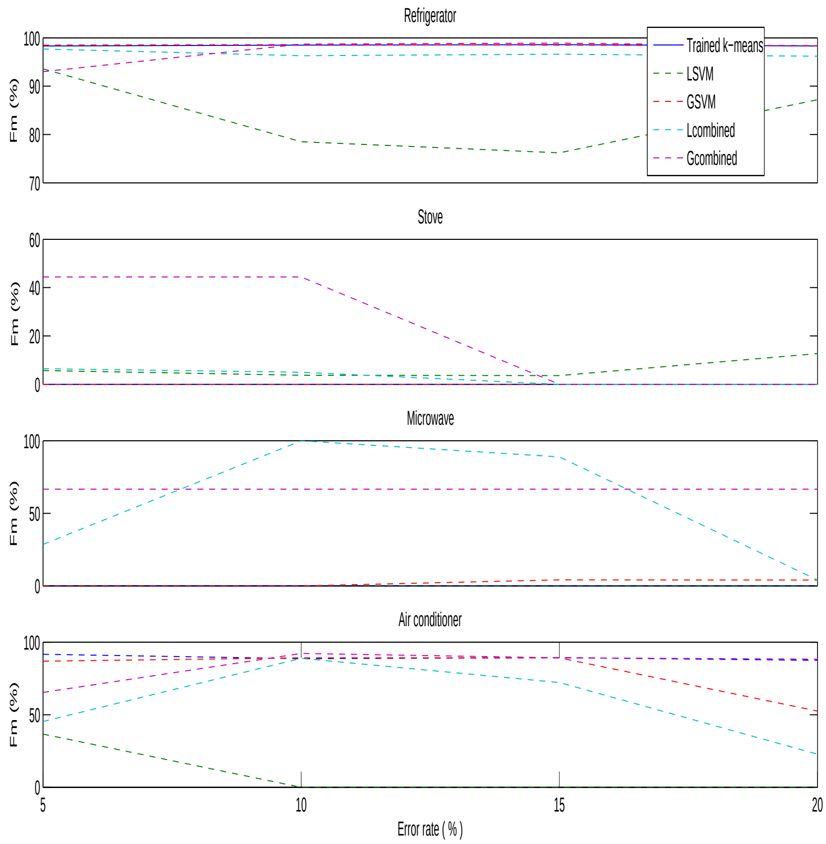


Figure 4.13: F-Measure of increased error rate after disaggregation of each appliance by k-means, Linear and Gaussian SVMs and Linear and Gaussian combined method for House 6 in the REDD dataset.

4.5 Benchmark

In this section, we compare the proposed Linear and Gaussian approaches with the state-of-the-art HMM of [43], which was designed for low-sampling (1 min) rates. For each dataset, Linear combined algorithm, Gaussian combined algorithm and HMM were always tested using the same amount and periods of data for training (7000 samples or roughly one week) and testing (four weeks). The HMM-based method [43] requires prior initialization of the model using expert knowledge (state variances, mean value for each state and state transition probabilities), which was carried out in our experiments either using the information provided by the authors of [43], or were modeled during experiments. Then, we gradually reduce training and insert errors in the label-vector to test robustness as explained in Section 4.4.3.

4.5.1 Performance

Table 4.6 shows total F-Measure after disaggregation of REDD dataset houses 1, 2 and 6 using Linear combined algorithm, Gaussian combined algorithm and the state-of-the-art HMM. All tested algorithms always use the same edge detection and feature extraction method explained previously. All algorithms show high total F-Measure of above 70% in all tested houses.

From the results for all tested houses, it can be seen clearly that train and test execution times of Linear and Gaussian combined algorithms are significantly lower than HMM train and test times. In House 1, Linear and Gaussian combined algorithms need less than 1 second in total for training and testing compared to roughly 50 seconds to train and test HMM algorithm. However, Total F-Measure for all three algorithms need similar 77.52%, 81.88% and 77.06% respectively. In House 2, Linear and Gaussian combined algorithms have less than 0.4 second for training and less than 0.6 seconds for testing compared to roughly 22.767 and 18.088 seconds to train and test HMM algorithm respectively. However, total F-Measure for all three algorithms was above 82%.

Table 4.6: Comparison between Linear combined algorithm, Gaussian combined algorithm and HMM using Execution time and total F-Measure for REDD data Houses. L=Linear. G=Gaussian.

Method	H 1			H 2			H 6		
	Train (sec)	*Test* (sec)	*Fm* (%)	*Train* (sec)	*Test* (sec)	*Fm* (%)	*Train* (sec)	*Test* (sec)	*Fm* (%)
L Combined	0.298	0.388	77.52	0.352	0.553	82.17	0.301	0.334	95.58
G Combined	0.484	0.442	**81.88**	0.42	0.488	**87.49**	0.47	0.477	**96.87**
HMM	28.317	22.903	77.06	22.767	18.088	82.38	30.22	16.189	72.82

Tables 4.7, 4.8 and 4.9 show a detailed comparison of REDD dataset Houses 1, 2 and 6 between our combined algorithms and HMM. Table 4.7 shows Precision, Recall and F-Measure in House 1. All three algorithms have similar results for all five tested appliances except that HMM gave poorer performance in detecting toaster and washer dryer with zero value for both appliances. Gaussian combined algorithm showed better Precision, Recall and F-Measure in detecting toaster with 100%, 30.55% and 46.80% respectively.

House 2 results are presented as Table 4.8 of Precision, Recall and F-Measure of Linear combined, Gaussian combined and HMM. All three algorithms have similar results for all five tested appliances except that Gaussian combined algorithm gave poorer performance in detecting stove. HMM has better F-Measure results in detecting microwave with 10% and 20% higher performance. Thus, combined algorithms have higher F-Measure to disaggregate dishwasher with roughly 30% each compared to 12.32%. House 6 results are present in Table 4.9. It can be seen that all algorithms have very close results in classification of refrigerator and microwave and none of them have correctly detected any toaster events. Linear and Gaussian combined algorithms gave much higher performance in recognizing air conditioner events using a proper feature combination. HMM had also lower F-Measure with zero value in stove detection compared to roughly 33.3% using rest of algorithms.

Table 4.7: Comparison between Linear combined algorithm, Gaussian combined algorithm and HMM using F-Measure for REDD data House 1. L=Linear. G=Gaussian.

Method	L Combined			G Combined			HMM		
	Pr (%)	Re (%)	F_m (%)	Pr (%)	Re (%)	F_m (%)	Pr (%)	Re (%)	F_m (%)
Refrigerator	83.19	96.82	89.49	90.18	96.57	**90.18**	90	77.16	83.12
Microwave	64.81	61.40	63.06	68	89.47	**77.27**	79.31	60.53	68.66
Toaster	16.66	2.77	4.76	100	30.55	**46.80**	0	0	0
Dishwasher	50	28.08	35.97	68.96	44.94	54.42	44.63	86.17	**58.80**
Washer Dryer	92.85	63.41	**75.36**	65.30	78.04	71.11	0	0	0

Table 4.8: Comparison between Linear combined algorithm, Gaussian combined algorithm and HMM using F-Measure for REDD data House 2. L=Linear. G=Gaussian.

Method	L Combined			G Combined			HMM		
	Pr (%)	Re (%)	F_m (%)	Pr (%)	Re (%)	F_m (%)	Pr (%)	Re (%)	F_m (%)
Refrigerator	92.87	95.88	**94.35**	91.07	97.50	94.17	87.45	87.93	87.69
Stove	66.66	33.33	44.44	0	0	0	38.10	66.67	**48.48**
Microwave	15.38	82.05	25.91	27.63	53.84	36.52	35.71	58.14	**44.25**
Toaster	69.62	91.66	**79.13**	67.69	73.33	70.40	50	92.45	64.90
Dishwasher	22.22	42.85	29.26	50	21.42	**30**	33.33	7.56	12.32

4.5.2 Robustness test

Tables 4.10 to 4.15 show a detailed comparison of REDD dataset Houses 1, 2 and 6 in robustness between our combined algorithms and HMM. We test robustness of our

Table 4.9: Comparison between Linear combined algorithm, Gaussian combined algorithm and HMM using F-Measure for REDD data House 6. L=Linear. G=Gaussian.

Method	L Combined			G Combined			HMM		
	Pr (%)	*Re* (%)	F_m (%)	*Pr* (%)	*Re* (%)	F_m (%)	*Pr* (%)	*Re* (%)	F_m (%)
Refrigerator	98.28	99.01	98.65	98.06	99.75	**98.90**	69.20	96.13	80.47
Stove	20	100	**33.33**	33.33	33.33	**33.33**	0	0	0
Microwave	100	74.07	85.10	85.71	88.88	87.27	100	100	**100**
Toaster	0	0	0	0	0	0	0	0	0
Air conditioner	88.88	86.95	87.91	92.72	91.07	**91.89**	0.43	100	0.85

combined algorithm in comparison with HMM similar to Section 4.4.3 by reducing the training dataset size and replacing correct labels with wrong ones in the training label-vector. Testing robustness in House 1 is presented in Tables 4.10 and 4.11. It can be seen that all algorithms are quite robust in disaggregating refrigerator, microwave and dishwasher up to 2000 samples and 20% error rate, but less successful in detecting toaster as all gave zero or close to zero values in different training sizes and error rates. The combined algorithms showed better performance to correctly distinguish washer dryer appliance.

In House 2, robustness results are presented in Tables 4.12 and 4.13. Both tables show that all algorithms are quite robust in disaggregating refrigerator, toaster and dishwasher up to 2000 samples and 20% error rate. Gaussian combined algorithm gave better performance in detecting microwave correctly with higher error rates than Linear combined algorithm and HMM. Although, HMM managed to detect stove with 48.48% F-Measure.

Tables 4.14 and 4.15 show results of House 6. It is very clear that Linear and Gaussian combined algorithms have slightly higher results in detecting refrigerator and stove in all different training sizes with more than 10% better F-Measure in reduced training experiments and 44% better in error insertion experiments by Gaussian combined algorithm. A much better performance was made by our combined algorithms in detecting air conditioner with up to 88.88% F-Measure compared to less than 1% F-Measure only by HMM in both tables. However, HMM managed to successfully disaggregate Microwave at 6000 samples and 5% error rate with 100% F-Measure.

Table 4.10: Comparison between Linear combined algorithm, Gaussian combined algorithm and HMM using F-Measure of reduced training size for REDD data House 1. L=Linear. G=Gaussian.

Method		*Refrigerator*	*Microwave*	*Toaster*	*Dishwasher*	*Washer Dryer*
	Train size	(%)	(%)	(%)	(%)	(%)
L Combined	6000	89.4	62.9	0	31.08	20.3
	4000	89.5	71.5	0	40.4	0
	2000	88.9	62.9	0	0	0
G Combined	6000	86.2	57.8	0	13.5	63.3
	4000	85.3	71.1	0	23.6	52.8
	2000	81	56.6	0	0	0
HMM	6000	84.03	68.66	1.11	58.8	0
	4000	84.03	52.99	1.11	53.01	0
	2000	83.29	52.99	1.11	53.01	0

Table 4.11: Comparison between Linear combined algorithm, Gaussian combined algorithm and HMM using F-Measure of Error insertion size for REDD data House 1. L=Linear. G=Gaussian.

Method		*Refrigerator*	*Microwave*	*Toaster*	*Dishwasher*	*Washer Dryer*
	Error rate	(%)	(%)	(%)	(%)	(%)
L Combined	5%	89.3	50.9	4.6	14.8	80
	15%	89	58.2	3	40.8	81.1
	20%	88.9	58.5	3.3	37.9	81.1
G Combined	5%	89.7	56.3	0	0	34.9
	15%	90.2	72.8	0	63.7	25.7
	20%	88.5	54.4	0	0	32.9
HMM	5%	81.49	48.57	0	58.8	0
	15%	81.49	65.35	0	0	0
	20%	81.34	65.35	0	0	0

4.5.3 Benchmark Summary

Experimental results using REDD data Houses 1, 2 and 6 demonstrate the competitiveness of the proposed solutions with respect to a state-of-the-art HMM-based approach. Indeed, the proposed approaches show similar performance to that of HMM, with up to 18 and 13 times lower execution time for testing and training, respectively. Tests, conducted by reducing the training size and introducing errors in the training dataset, showed high robustness of the proposed approaches, that are capable of performing successful disaggregation using only two days of training data and up to 20% of errors in the training set.

Table 4.12: Comparison between Linear combined algorithm, Gaussian combined algorithm and HMM using F-Measure of reduced training size for REDD data House 2. L=Linear. G=Gaussian.

Method		*Refrigerator*	*Stove*	*Microwave*	*Toaster*	*Dishwasher*
	Train size	(%)	(%)	(%)	(%)	(%)
L Combined	6000	91.84	0	0	8.9	41.6
	4000	91.7	0	0	6.4	40
	2000	90.22	0	0	60	0
G Combined	6000	93.5	0	40	46.5	22.2
	4000	92.3	0	0	9.4	23
	2000	90.3	0	0	5.2	0
HMM	6000	87.69	48.48	44.25	64.90	12.32
	4000	83.4	0	44.25	0	0
	2000	83.55	0	44.25	0	0

Table 4.13: Comparison between Linear combined algorithm, Gaussian combined algorithm and HMM using F-Measure of Error insertion size for REDD data House 2. L=Linear. G=Gaussian.

Method		*Refrigerator*	*Stove*	*Microwave*	*Toaster*	*Dishwasher*
	Error rate	(%)	(%)	(%)	(%)	(%)
L Combined	5%	91.8	0	0	11.4	17.39
	15%	93.3	0	0	11.9	42.8
	20%	91.9	0	0	8.9	45.3
G Combined	5%	92.1	0	50.6	35.1	21.8
	15%	91.7	0	51.7	9.3	19.66
	20%	93.4	0	3.3	46.2	25.8
HMM	5%	83.42	48.48	44.25	64.90	12.32
	15%	83.42	48.48	44.25	46.97	12.32
	20%	83.55	18.65	0	46.97	12.32

4.6 Chapter Summary and Conclusion

Designing accurate NALM algorithms for low sampling data is challenging. In this chapter we tested our proposed Linear and Gaussian low-complexity combined algorithms based on combining Trained k-means and Linear and Gaussian Support Vector Machines. Appliances from a range of REDD dataset houses with roughly 1 minute sampling rate with 5 appliances each are used to evaluate performance and robustness. The two combined algorithms using house-specific training data are accurate even when the training period is short until up to two days only for training and training errors are present with up to 20% error rate. The combined algorithms

Table 4.14: Comparison between Linear combined algorithm, Gaussian combined algorithm and HMM using F-Measure of reduced training size for REDD data House 6. L=Linear. G=Gaussian.

Method		*Refrigerator*	*Stove*	*Microwave*	*Air conditioner*
	Train size	(%)	(%)	(%)	(%)
L Combined	6000	98.3	6.6	100	55.3
	4000	98.6	33.3	0	65.1
	2000	98.5	57.1	0	88.8
G Combined	6000	84.5	18	0	47.9
	4000	98.14	15.7	0	82.14
	2000	91.7	60	0	0
HMM	6000	80.47	0	100	0.85
	4000	77.44	0	0	0
	2000	77.44	0	0	0

Table 4.15: Comparison between Linear combined algorithm, Gaussian combined algorithm and HMM using F-Measure of Error insertion size for REDD data House 6. L=Linear. G=Gaussian.

Method		*Refrigerator*	*Stove*	*Microwave*	*Air conditioner*
	Error rate	(%)	(%)	(%)	(%)
L Combined	5%	97.7	6.4	28.5	45.3
	15%	96.6	0	88.8	72.2
	20%	96.2	0	4	22.8
G Combined	5%	93	44.4	66.6	65.3
	15%	98.9	0	66.6	89.2
	20%	98.3	0	66.6	88.3
HMM	5%	80.47	0	100	0.85
	15%	80.47	0	0	0.85
	20%	79.60	0	0	0.85

also showed a competitive performance to state-of-the-art Hidden Markov Models and Support Vector Machines. A set of different feature combinations is used to maximize performance. Next, we create a database of signatures to eliminate the training step.

Chapter 5

Appliance Database Creation and Clustering

5.1 Introduction

To make our combined algorithms practical and reduce or remove household occupier effort in maintaining a time-diary, a database of appliance signatures is created in order to train the classifier models. The designed database is a compact collection of appliance power load signatures which are used to develop statistical features, such as mean, variance and auto-correlation for each appliance that are then used for load disaggregation. The database is populated using open source datasets from Austria and Italy [40], and UK [41].

Similar attempts have recently been reported in [120], where a database of signatures was introduced, called Plug-Level Appliance Dataset (PLAID), accommodated by roughly 200 appliances from US datasets sampled at a high resolution of 30 kHz. The database is made publicly available for current and voltage readings with over 1kHz sampling rate.

Load signatures contain useful information with respect to operational characteristics of the loads in an electrical circuit, which can be used to study similarities and differences in signatures between appliances. By doing so, one can predict whether a feature or a load characteristic collected in one house can successfully help disaggregate an appliance of another house.

First, in this chapter, we compare different statistical models of appliances and then we create a database of signatures that uses Gaussian modeling as previously published in [132]. In Section 5.5, we propose two simple methods to classify household appliances into different groups using mean-shift and k-means algorithms. In Section 5.6, results of the classification methods will be presented followed by summary and conclusion section.

5.2 Novel Contributions

Novel contributions of this chapter are:

- A generic database of appliance load profiles populated from 34 houses in UK and Europe containing over 130 appliance signatures.
- A clustering approach using Mean-shift algorithm.
- A load hierarchical tree using unsupervised k-means algorithm.
- A detailed investigation of the signature database of appliance load profiles by clustering all houses from the open source databases from Italy and Austria (GREEND dataset) [40] and two UK (REFIT dataset) [41] houses into groups and sub-groups.

5.2.1 Appliance Modeling Validation

All domestic appliances are designed to work within a certain active power range, which can often be found in the appliance instruction manual. However, in practice, the consumed power will deviate due to electrical noise, interference, aging etc.

We test modeling suitability of three common statistical models for representing power signature: Gaussian, Laplace and Log-normal using these equations, respectively:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} exp^{-(\frac{x-\mu^2}{2\sigma^2})} \tag{5.1}$$

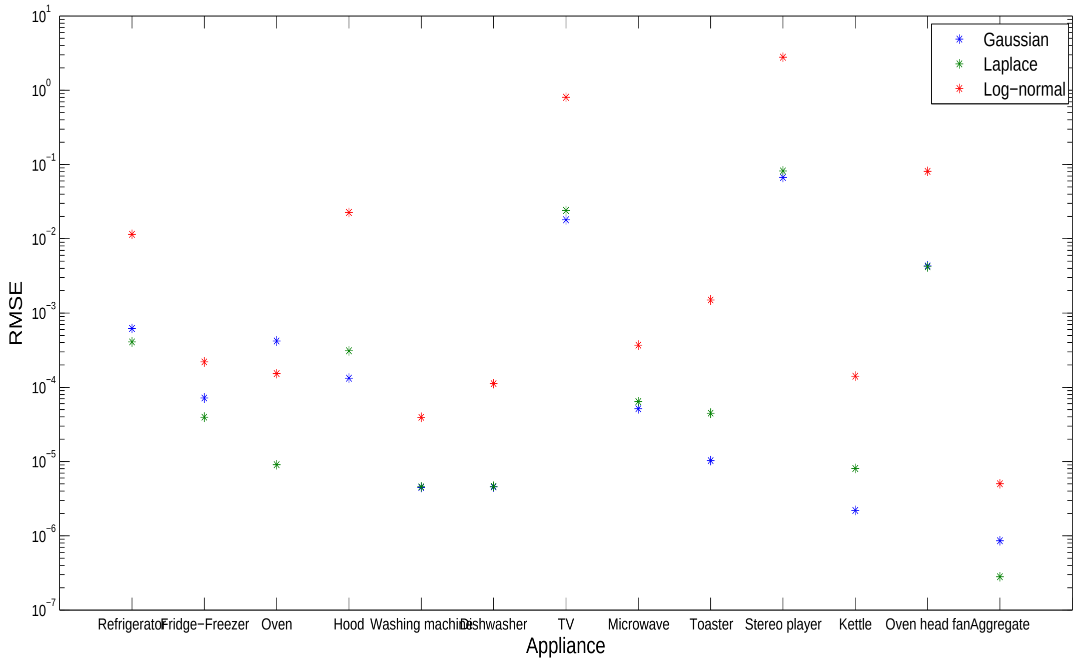


Figure 5.1: Root-Mean-Square Error of Gaussian mixture, Laplace and Log-normal statistical models for 13 appliances from REFIT and GREEND datasets, and the REFIT aggregate meter reading, all shown on a log-scale.

where σ is the standard deviation, μ is the mean for Gaussian Probability Distribution Function and x is the profit bin which is the histograms of the real distribution curve.

$$f(x) = \frac{1}{2b} exp(-\frac{|x-\mu|}{b}) \tag{5.2}$$

where σ is the standard deviation, μ is the mean for Laplace Probability Distribution Function and x is the profit bin and b is the diversity parameter which is higher than 1.

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} exp^{(-\frac{(ln(x-\mu))^2}{2\sigma^2})} \tag{5.3}$$

where σ is the standard deviation, μ is the mean for lognormal Probability Distribution Function and x is the profit bin which is the histograms of the real distribution curve.

Figure 5.1 presents the Root-Mean-Square Error (RMSE) for each of the three distribution models, using selected appliances from GREEND and REFIT datasets. The values are obtained by averaging in time and across different houses using:

$$RMSE = \sqrt{\left(\frac{1}{N}\left(\sum_{i=1}^{N}(y_i - \bar{y}_i)\right)^2\right)} \tag{5.4}$$

where N is the total number of observations, that is, active power values for the modelled appliance. $(y_i - \bar{y}_i)$ is the difference between estimated power reading at time instance i and model estimated value at the same time instance.

From the figure, it can be seen that the Gaussian mixture model is the best fit for all appliances, especially for high loads such as kettle and toaster. As expected, the Gaussian mixture model is also the best for the aggregate readings, since it is a sum of nearly independent processes. This validates our approach of using the Gaussian mixture model (red line in Figure 5.2) to build the distribution model of all appliances. We also note that RMSE is insignificantly small except for low-consumers such as TV and Stereo Player due to their long operation cycles. These findings are similar to those reported in [89].

5.3 Signature Database Creation

To capture the electrical behavior of an appliance, we present in Figure 5.2 probability density function (pdf) of active power for several domestic appliances. These results are obtained using one month of continuous monitoring in five houses. Because we are targeting low sampling rates (under 1Hz) we focus only on steady state operation removing automatically in the process of data cleaning transient values from each appliance operation. It can be seen from the figure that the consumed active power follows a similar distribution trend for all appliances: pdf rapidly decreases from one to three peaks with few isolated impulses. In addition, and similarly to [89], and validated in the previous section, we model each appliance power load profile using a Gaussian mixture model, obtained via curve fitting, and store mean, variance and the first two correlation coefficients calculated as:

$$R(\tau) = (\sum (y_t - \mu)(y_{t+\tau} - \mu))/\sigma^2, \tag{5.5}$$

where μ is the mean power value, σ^2 is the variance and τ is the sample lag.

Some appliances, so called *multi-state appliances*, such as washing machine and dishwasher, have several operating states, for example, washing machine usually contains three cycles: washing, rinse and spin. The number of Gaussian components in the Gaussian mixture model is set to the number of operating states. Standby modes are neglected using proper thresholds for each appliance.

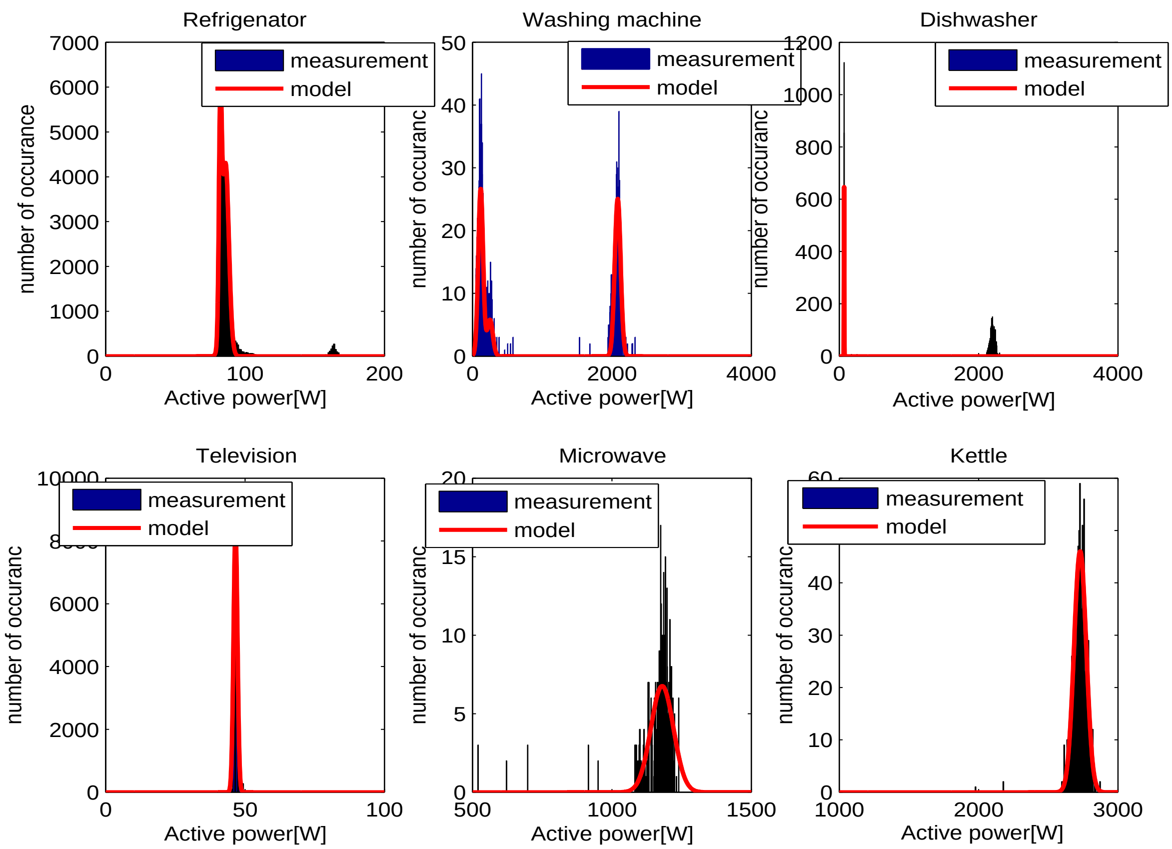


Figure 5.2: Gaussian mixture probability density distribution models of six REFIT dataset appliances.

To replace NALM training, we create a database of appliance load profiles. Over 100 different appliance-load profiles are collected from GREEND and REFIT datasets with appliances ranging from standard kitchen appliances, such as kettle, toaster, microwave, white appliances, washing machines, dishwashers, to electronics, TV, ADSL modem, radios and PCs. We stored one complete working cycle of each appliance at the acquired sampling rate. A working cycle is stored when an appliance is ON until it changes its states to OFF. Working cycle of some appliances that run all the time such as refrigerator and freezer is stored when appliance is ON until it changes it states to standby mode.

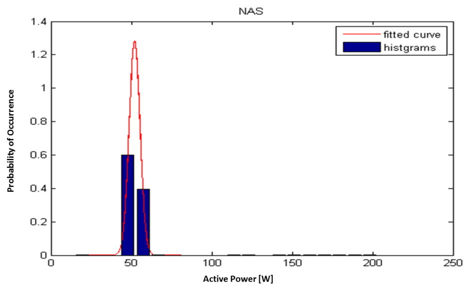
Each input to the database contains the statistical parameters, appliance type, model (where available), the origin (dataset including the measuring sampling rate), and one full-cycle signature. The database is organized into CSV files, where each file corresponds to one appliance type.

5.4 Load-profiles of Selected Household Appliances

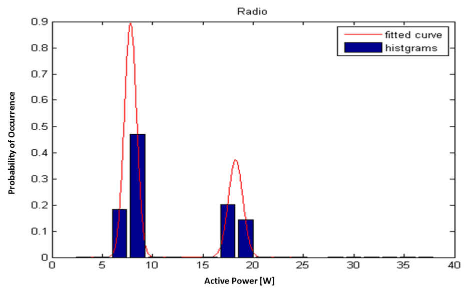
5.4.1 GREEND Dataset

Figures 5.3, 5.4 and 5.5 show the Gaussian mixture distribution curve along with histograms, generated using true data, for a randomly selected sets of appliances from GREEND dataset. They are split into three sets for clarity. The first set has 4 appliances which are: House 2 Network Access Storage (NAS), House 1 Radio, House 3 Hair drier and House 5 Stove. The second set has 4 appliances which are: House 0 Kitchen lamp, House 1 Microwave, House 0 Vacuum cleaner and House 2 Tumble dryer. The third set with 4 appliances which are: House 0 Radio, House 7 Hood, House 2 Bread maker and House 7 ADSL modem. It can be seen from these three sets of figures that the Gaussian mixture model and true curves fit well together, and this is also validated in Tables 5.1, 5.2 and 5.3 which show that RMSE values for those appliances are relatively small.

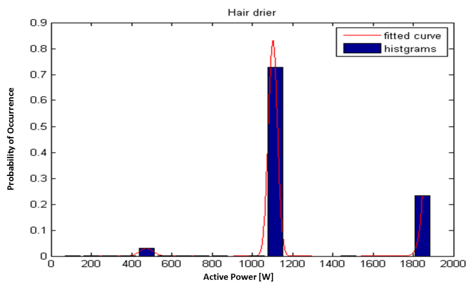
Figures B.1, B.2 from Appendix B and Figure 5.6 show Gaussian mixture distribution models for different Refrigerators from GREEND dataset Houses 0, 1, 3, 4, 5 and 7, Televisions from GREEND dataset Houses 0, 2, 3, 4, and 5, and Coffee makers from GREEND dataset Houses 0, 2 and 3. It is obvious from these figures that energy consumptions of these different appliance types are very different. Tables B.1, B.2 and 5.4 show that regardless of the type, each appliance is modeled well resulting in relatively small RMSE.



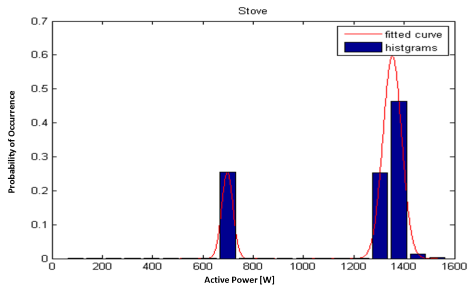
(a) H2 NAS



(b) H1 Radio



(c) H3 Hair drier



(d) H5 Stove

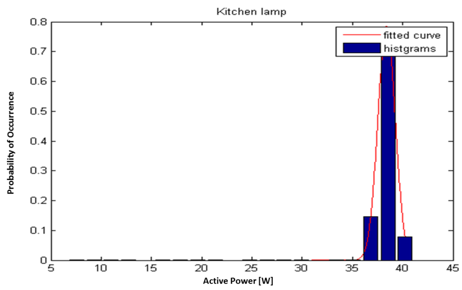
Figure 5.3: Pdf for four different appliances from the GREEND dataset. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

Table 5.1: RMSE, mean, variance, $1st$ ***order correlation coefficient and*** $2nd$ ***order correlation coefficient for different appliances of different GREEND houses. H denotes House number.***

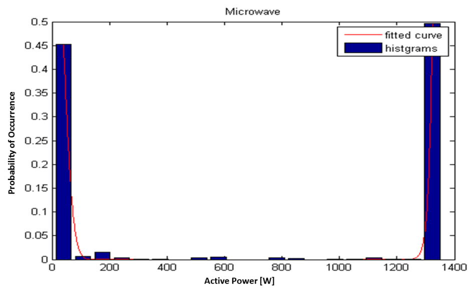
Appliance	Mean value [W]	Variance	RMSE	1st Cor.	2nd Cor.
H2 NAS	52.56	9.54	7.90 E-04	0.0106	0.0251
H1 Radio	19.08	0	3.6 E-04	-0.0487	0.1707
H3 Hair drier	527 1117.1 1867.9	195.84 13.68 11.8	7.74 E-04	0.1208	-0.0284
H5 Stove	677.7 1353.1	58.2 26.98	1.2 E-03	0.0682	-0.0144

5.4.2 REFIT Dataset

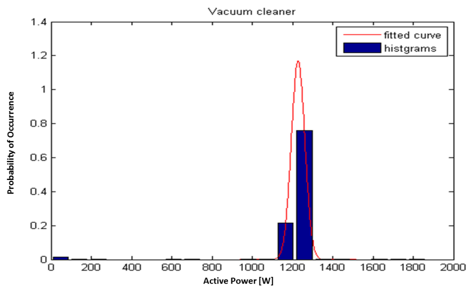
Figures 5.7, 5.8 and 5.9 show the Gaussian distribution curve along with histograms, generated using true data, for a randomly selected sets of appliances from REFIT dataset. They are split into three sets for clarity. The first set has 4 appliances which are: House 7 Fridge, House 3 Television, House 5 kettle and House 7 Toaster. The second set has 4 appliances which are: House 1 Tumble dryer, H5 Combination



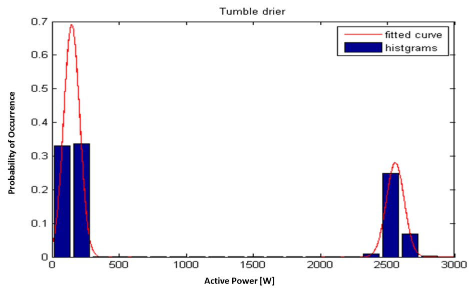
(a) H0 Kitchen lamp



(b) H1 Microwave



(c) H0 Vacuum cleaner

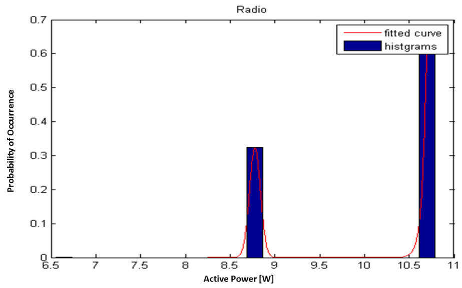


(d) H2 Tumble drier

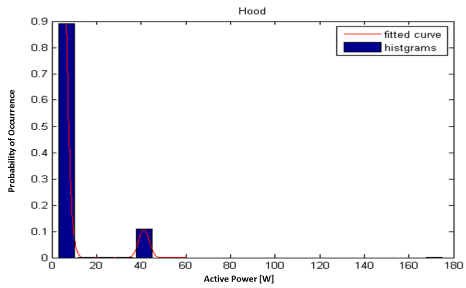
Figure 5.4: Pdf for four different appliances from the GREEND dataset. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

Table 5.2: RMSE, mean, variance, 1st order correlation coefficient and 2nd order correlation coefficient for different appliances of different GREEND houses. H denotes House number.

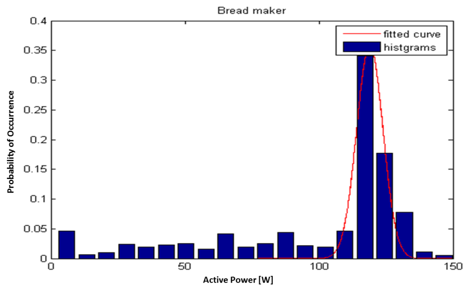
Appliance	Mean value [W]	Variance	RMSE	1st Cor.	2nd Cor.
H0 Kitchen lamp	38.81	1.04	1.2 E-04	-0.027	-0.0092
H1 Microwave	62.1 1316.6	74.19 62.42	4.9 E-03	0.0611	-0.0485
H0 Vacuum cleaner	1208	164.98	0.0038	0.1005	-0.0529
H2 Tumble drier	87.1 2558.2	94.7 72.21	5.94 E-04	0.0013	-0.0479



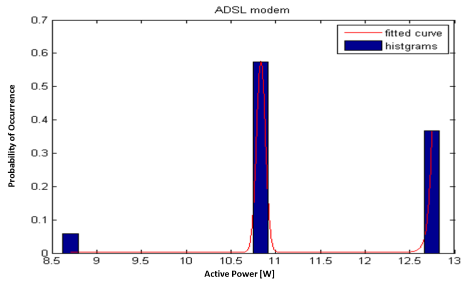
(a) H0 Radio



(b) H7 Hood



(c) H2 Bread maker



(d) H7 ADSL modem

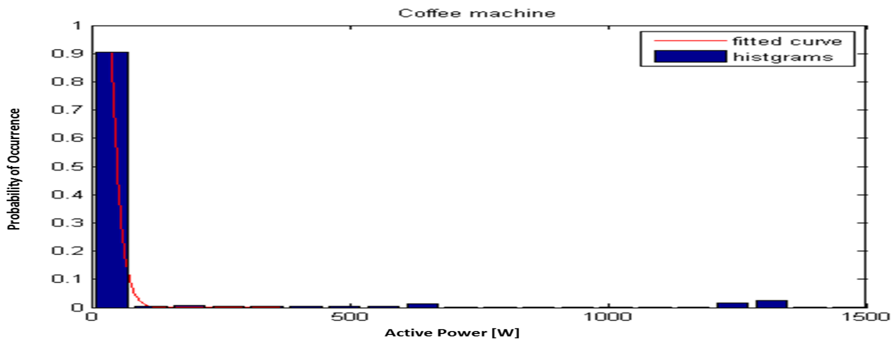
Figure 5.5: Pdf for four different appliances from the GREEND dataset. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

Table 5.3: RMSE, mean, variance, 1st order correlation coefficient and 2nd order correlation coefficient for different appliances of different GREEND houses. H denotes House number.

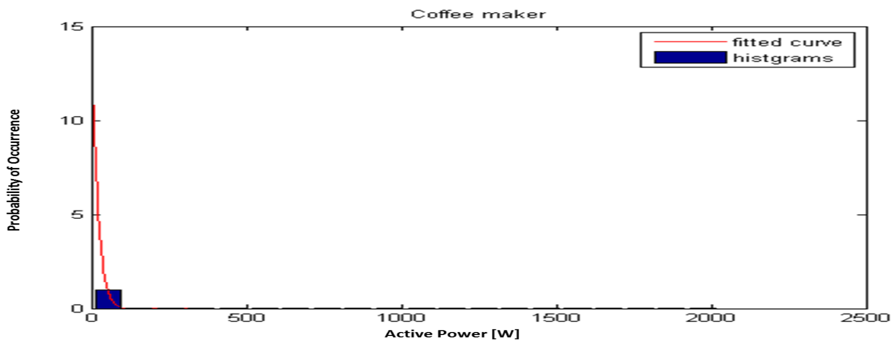
Appliance	Mean value [W]	Variance	RMSE	1st Cor.	2nd Cor.
H0 Radio	8.67 10.8	0 0.1325	0.0017	-0.3196	-0.2253
H7 Hood	6.58 173.14	12.41 1.94	1.33 E-04	0.03	-0.0023
H2 Bread maker	98.83	36.26	2.97 E-02	-0.0072	-0.265
H7 ADSL modem	2.19 16.87	0 8.56	1.33 E-02	0.1909	0.1247

Table 5.4: RMSE, mean [W], variance, 1st order correlation coefficient and 2nd order correlation coefficient for different Coffee makers of different GREEND houses.

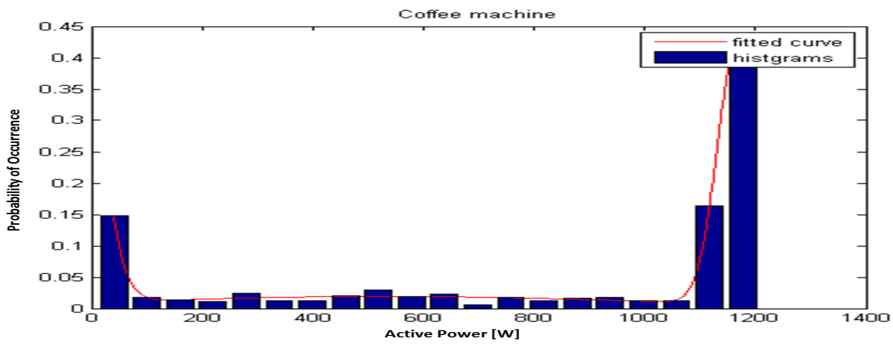
Appliance	Mean value	Variance	RMSE	1st Cor.	2nd Cor.
H0 Coffee machine	83.11	283.37	8.4 E-03	-0.0103	-0.1678
H2 Coffee machine	21.44	150.01	1.7 E-03	-0.1272	-0.1235
H3 Coffee machine	49.7 531.2 1155.1	1.07 312.55 25.9	6.1 E-03	0.0359	-0.2965



(a) House 0



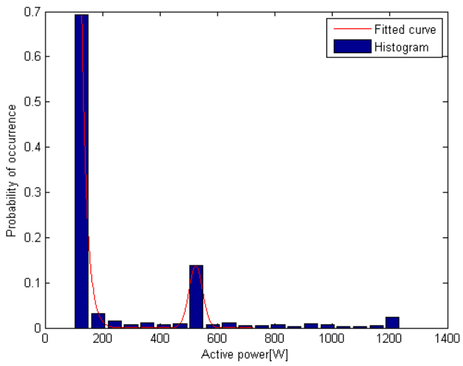
(b) House 2



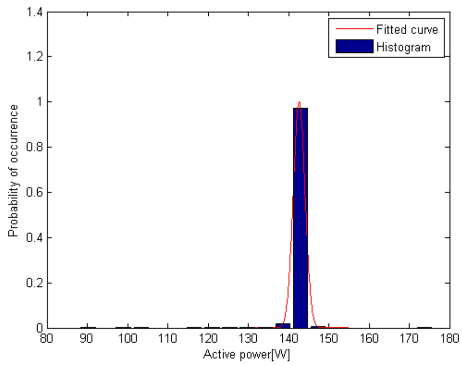
(c) House 3

Figure 5.6: Different distributions of different Coffee Makers types from the GREEND dataset. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

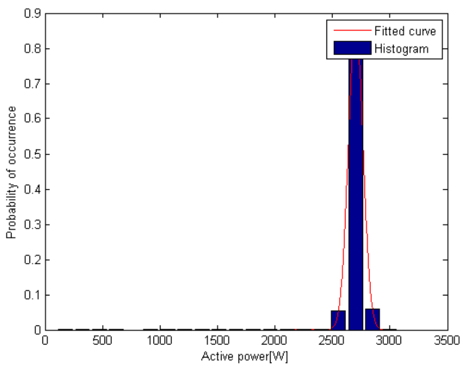
microwave, House 6 Toaster and House 7 Tumble dryer. The third set is with 4 appliances which are: House 1 Chest freezer, House 1 Electric heater, House 7 Kettle and House 6 freezer. It can be seen from these three sets of figures that the Gaussian mixture model and true curves fit well together, and this is also validated in Tables 5.5, 5.6 and 5.7 which show that RMSE modeling values for those appliances are relatively small. However, these errors are slightly higher than those obtained by GREEND dataset due to noise that is present in REFIT dataset.



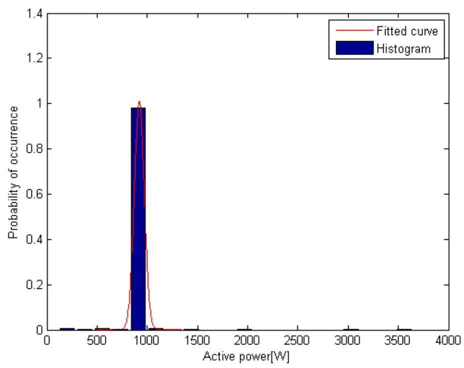
(a) H0 Refrigerator



(b) H3 Television

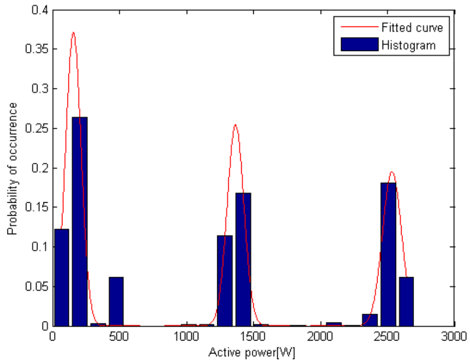


(c) H5 Kettle

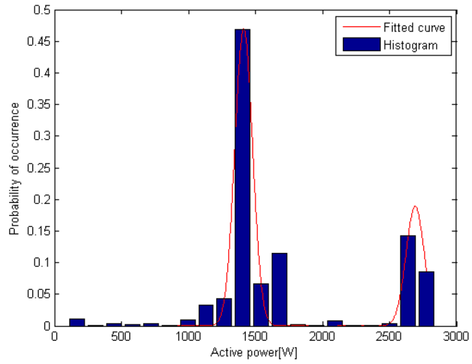


(d) H7 Toaster

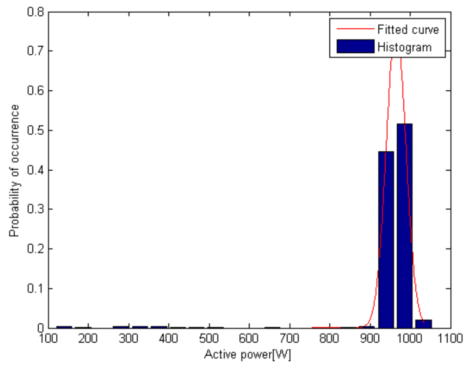
Figure 5.7: Pdf for four different appliances from the REFIT dataset. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].



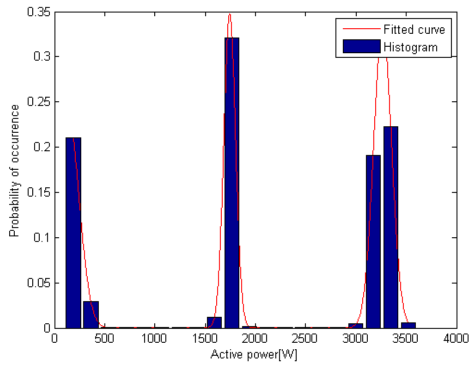
(a) H1 Tumble dryer



(b) H5 Microwave



(c) H6 Toaster

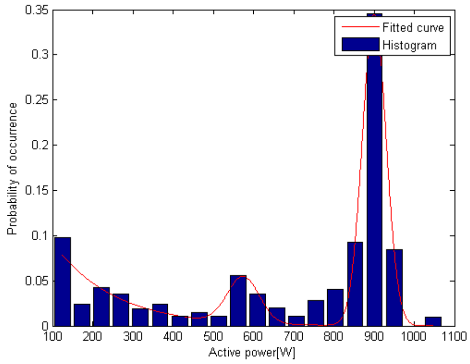


(d) H7 Tumble dryer

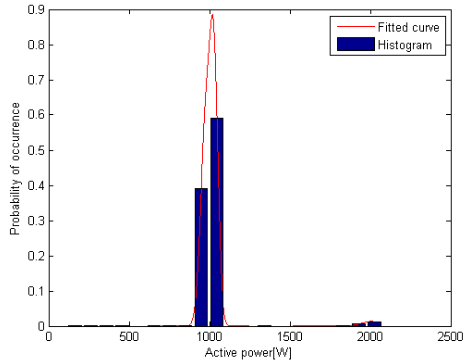
Figure 5.8: Pdf for four different appliances from the REFIT dataset. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

Table 5.5: RMSE, mean, variance, 1***st order correlation coefficient and*** 2***nd order correlation coefficient for different appliances of different REFIT houses. H denotes House number.***

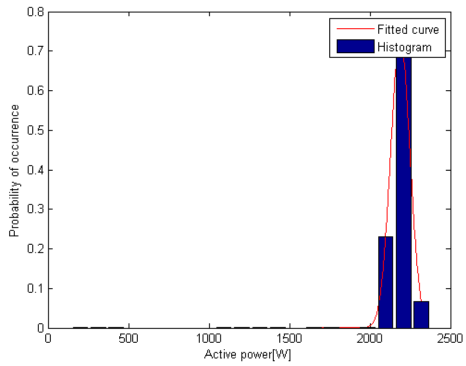
Appliance	Mean value [W]	Variance	RMSE	1st Cor.	2nd Cor.
H7 Refrigerator	103.4 506.75	1.6 308.07	0.0017	-0.8196	0.0253
H3 Television	142.7	2.046	3.02 E-04	0.763	-0.3023
H5 Kettle	2690	193.68	0.089	0.7122	-0.2675
H7 Toaster	914.54	135.91	0.0019	-0.4929	-0.7227



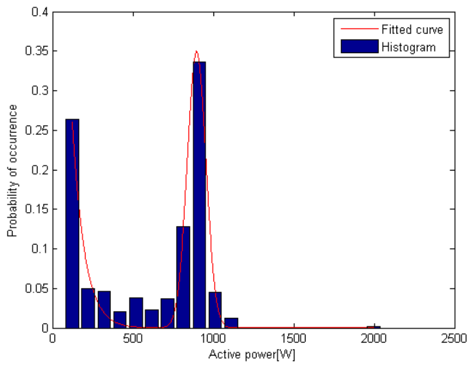
(a) H1 Chest freezer



(b) H1 Electric heater



(c) H7 Kettle



(d) H6 Freezer

Figure 5.9: Pdf for four different appliances from the REFIT dataset. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

Table 5.6: RMSE, mean, variance, 1st order correlation coefficient and 2nd order correlation coefficient for different appliances of different REFIT houses. H denotes House number.

Appliance	Mean value [W]	Variance	RMSE	1st Cor.	2nd Cor.
H1 Tumble dryer	121.1	46.29	0.0187	0.8796	-0.8203
	1214.8	349.64			
	2539.4	57.26			
H5 Microwave	1412	93.65	3.21 E-02	0.873	0.5223
	2692	96.35			
H6 Toaster	955.06	88.53	0.0025	-0.9002	0.965
H7 Tumble dryer	247.8	29.17	1.33 E-02	0.9923	-0.5247
	1754.9	51.83			
	3259.6	76.21			

Table 5.7: RMSE, mean, variance, 1st order correlation coefficient and 2nd order correlation coefficient for different appliances of different REFIT houses. H denotes House number.

Appliance	Mean value [W]	Variance	RMSE	1st Cor.	2nd Cor.
H1 Chest freezer	175.26 627.92 903.28	61.44 198.33 23.09	0.0196	0.8176	-0.5223
H1 Electric heater	570.3 1001.2 1994.5	529.26 16.08 36.7	8.06 E-05	0.0356	0.9023
H7 Kettle	2182.2	136.34	7.18 E-04	-0.0672	-0.865
H6 Freezer	241.18 895.35	228.58 70.18	0.0181	-0.2309	0.9247

5.5 Database Clustering

Grouping appliances could provide better understanding of appliance electric behavior, which would help predict if NALM algorithms could successfully disaggregate appliances or not. In this section, we introduce two methods to cluster appliances and appliance-states. The first method uses mean-shift algorithm to group appliances into different categories; while mean-shift does not require defining the number of groups, a bandwidth value has to be determined. The second method, uses the Genetic k-means to split appliances into two groups to form a Hierarchical tree with root and leaves. This clustering can be done on datasets or separate houses, big data or smaller sets of appliances. By doing that, one can study similarities and dissimilarities between load-profiles before applying disaggregation. In other words, clustering could predict if an appliance feature/signature is unique enough for NALM algorithms to correctly classify it or not.

5.5.1 Mean-shift Clustering

Appliance clustering can be performed using active power and reactive power readings or features extracted from their loads. Here, we use mean power value and variance of Gaussian signatures extracted from the signature database.

We use Mean-shift algorithm [121] to cluster possible appliances and appliance-

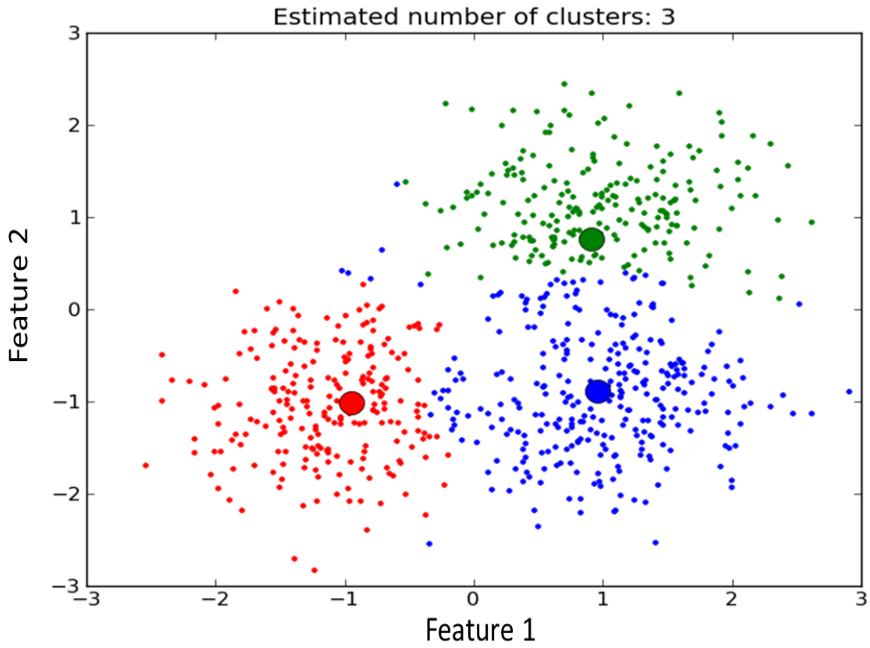


Figure 5.10: An example of mean-shift classification into three clusters. Big circles represent cluster heads and each colour represents a cluster.

states in GREEND and REFIT datasets based on their Gaussian load signature. In comparison to the classic k-means clustering approach, there are no assumptions on the number of modes or clusters that are needed [122]. As exemplified in Figure 5.10, we cluster all objects into possible groups and then we re-cluster any relatively big groups into sub-groups and so on.

5.5.2 Hierarchical Tree Clustering

In the hierarchical tree clustering, the objects that are needed to be clustered are organized into a tree. The tree shows how the objects (appliances) are related to each other, and the groups of the objects can be determined from the tree. Figure 5.11 shows a tree for clustering 12 objects i.e., appliances, together as an example. We start from the top and cluster down appliances into smaller groups in every step. Each node represents the number of appliances to be clustered.

In [123], a similar method was introduced using different features; based on voltage and current (V-I) trajectory, which then were compared to the taxonomies based on traditional power metrics and eigenvectors. It was found that the groups of appliances in the taxonomy based on V-I trajectory were well-separated and had engineering meanings. The authors used Euclidean distance between every pair of appliances or appliance-states and linked the closest two pairs together forming a leaf, then connected every two pairs of appliances to finally form the root of the tree. By doing so, it was proven that appliances from the same type, in terms of their operation purpose, do not perform in the same way, and their signatures can be significantly different. Their method, suffers from high complexity especially with high number of appliances.

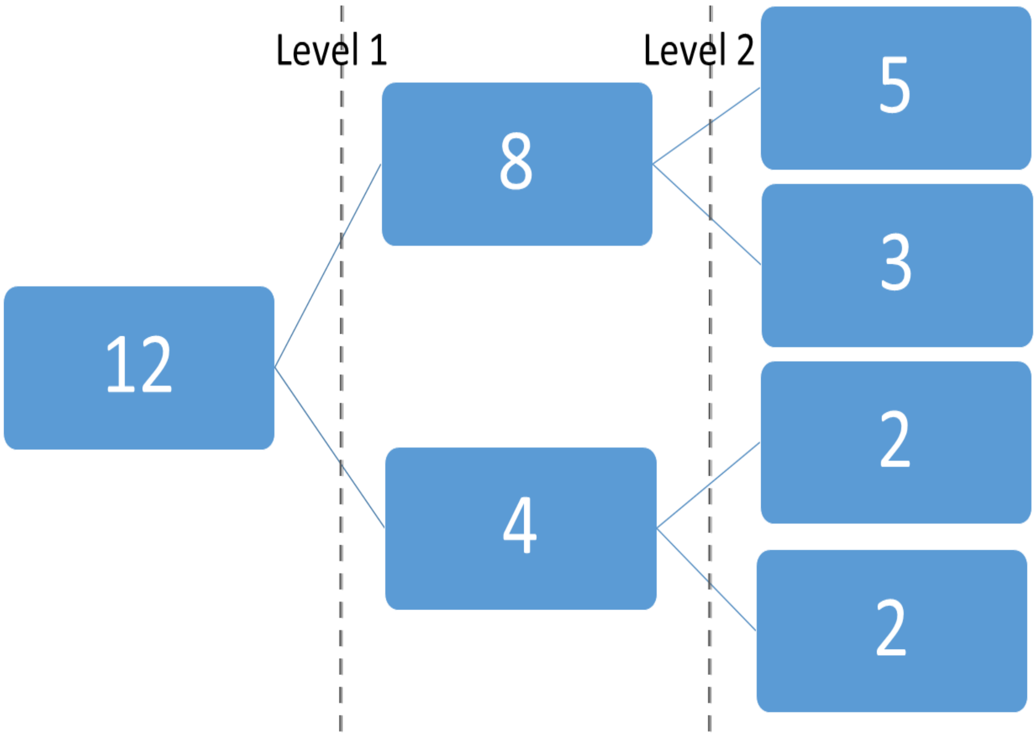


Figure 5.11: An example of tree classification using 12 objects. Each node shows the number of objects to be clustered.

The details about how the tree is formed, in our approach, and how objects are grouped together are:

- *Step 1*:
  - Consider all appliances together as a parent node in a tree. Split objects i.e., appliances, into two groups (clusters) using k-means clustering with k=2. Every new cluster is treated as a parent node and can be split into two clusters again until we reach a relatively small number of objects in every child node.

- *Step 2*:
  - Measure the Euclidean distance between the center of every child cluster and the center of every parent cluster. The distance between a pair of objects indicates how similar the objects are. Smaller distance between the child node to the parent cluster means that the objects in child cluster are similar to objects in the parent cluster. Here, an object is an appliance or an appliance-state.
- *Step 3*:
  - Link all nodes together to create the tree as shown in Figure 5.11; in this example we have 12 objects connected together as a tree and clustered down into two groups then every group was split into two groups. Basically, we can keep splitting clusters until we reach a relatively small size cluster at every end.
- *Step 4*:
  - In this step, we cut the tree at a certain hight to create a number of groups. Practically, distance between parent node and the two children nodes is roughly the same as illustrated in Figure 5.12. It can be seen from the figure that $d1$ is roughly equal to $d2$. $d1$ and $d2$ are distances between child 1 and child 2 centers (c1 and c2) to the parent cluster center (C). Therefore, we cut the clustering tree into a suitable *level* instead of a *hight*. A level indicates how many times we split each cluster from top to end, and *step 2* is no longer needed. The main factor that controls cutting at a suitable level is the number of groups that we need. If we cut the tree at a very high level we will have a small number of groups with a large number of objects in every group that are more dissimilar, and if we cut at a very low level we will have a large number of groups with a small number of objects that are more similar. In Figure 5.11, if we cut at level 1, two groups will be formed with 8 and 4 objects respectively. If we cut at level 2, four groups will be formed with 5, 3, 2 and 2 objects respectively.

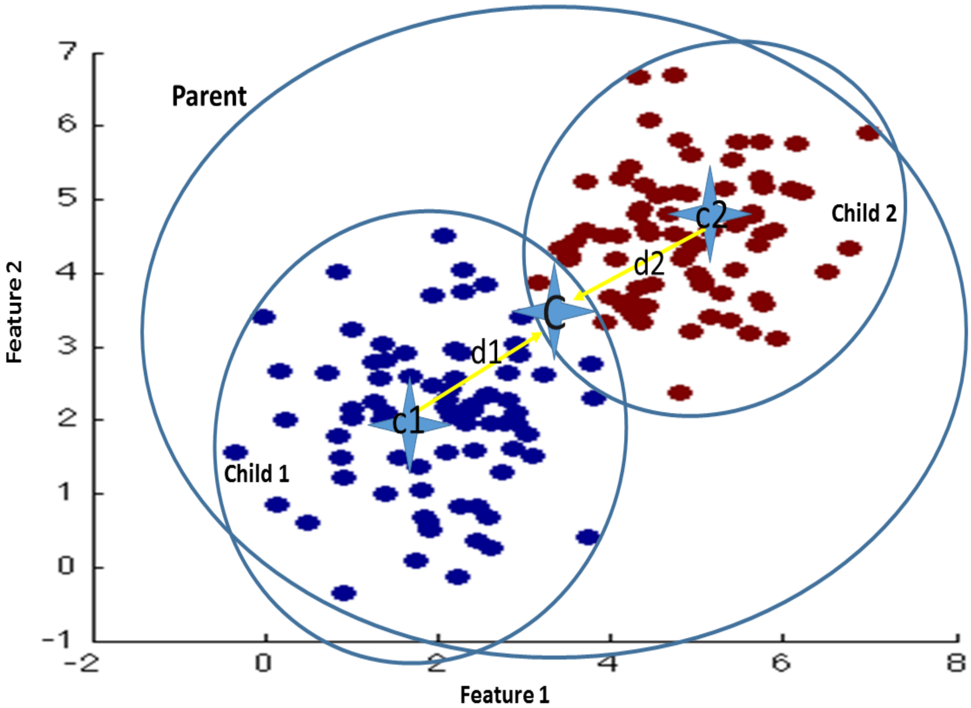


Figure 5.12: Parent cluster split into 2 clusters using k-means (k=2). $d1$ ***and*** $d2$ ***are distances between child 1 and child 2 centers (c1 and c2) to the parent cluster center (C).***

5.6 Results and Discussion

In this section, we use all houses from GREEND dataset and REFIT dataset Houses 2 and 3 to test our clustering methods. From these experiments, we try to group similar appliances together based on their Gaussian distribution characteristics. We also predict, from the clusters, which appliances would be distinguishable from similar types of appliances.

5.6.1 GREEND Dataset Results

5.6.1.1 Mean-shift Clustering

Figures 5.13 and Table B.17 (Appendix B) show the outcome from clustering all appliances and appliance-states in GREEND dataset into groups and sub-groups by mean-shift clustering method. Figure 5.13 (upper figure) shows all appliances and appliance-states clustered which formed 9 groups. Some groups have high number of appliances and appliance-states with roughly 63 objects in one of the groups and few have lower number of appliances and appliance-states with only one appliance-state in another group. Then we re-clustered group 1 since it has relatively high number of appliances and appliance-states, into smaller groups forming 6 sub-groups as shown in Figure 5.13 (middle figure) and Table B.18. Sub-group 1.2 was then re-clustered into smaller sub-groups as it has roughly 44 objects forming another 5 groups as shown in Figure 5.13 (lower figure) and Table B.19.

From Tables B.17, B.18, and B.19 it can be seen that many different appliances were clustered into the same groups such as group 6. Moreover, similar appliances were clustered into different groups such as televisions and refrigerators. It can also be noticed that group 1.2.3 has different types and states of radios from GREEND dataset Houses 0 and 1 along with House 5 computer with scanner and printer operating states and also all three operating states of ADSL modem of House 7, which could means that all of these appliances have similar Gaussian load characteristics despite the difference in their operating purpose.

Group 2 shows that some high consumption appliances have similar consumption behavior such as dishwashers and washing machines high states from Houses 0, 1, 2, 4 and 5. In the same group we have kettle from House 0, hair dryer high state from House 3 and House 5 iron state 4, which indicates that all these appliances show operate similarly. On the other hand, House 2 tumble dryer state 2 was clustered alone in group 4 as it has unique consumption load.

5.6.1.2 Tree Clustering

Figures 5.14 and 5.15 and Table B.20 show the outcome from clustering GREEND dataset into groups and sub-groups by tree clustering method using k-means algorithm. Figure 5.14 shows the classification tree cut into a suitable level (dashed line) which formed 8 groups. Some groups have high number of appliances and appliance-state with up to 47 objects in one of the nodes while few nodes have lower number of appliances and appliance-states with only one appliance-state in one of the nodes. Then we re-cluster one of the nodes which has 40 appliance-states (with red circle) into 10 sub-groups as shown in Figure 5.15. Re-cut the tree into a lower level again (dashed line in 5.15) to form more sub-groups.

From Table B.20, it can be seen that many similar appliances were clustered into different groups such as televisions and refrigerators. Furthermore, different appliances were clustered into same groups such as group 3.2. It can also be noticed that group 3.4 has different types and states of radios from GREEND dataset Houses 0 and 1 along with House 5 computer with scanner and printer both operating states and also all operating states of House 7 ADSL modem, as it might mean that all of these appliances have similar load characteristics. Likewise, group 3.9 has low mean power appliances such as House 0 TV and House 2 NAS.

Group 7 shows a good relation between some high consumption appliances such as dishwashers and washing machines high states from Houses 0, 1, 2, 4 and 5. In the same group we have House 3 kettle, House 3 hair dryer high state and House 5 iron state 3, which means that they all share similar Gaussian models. House 2 tumble dryer state 2 was clustered alone in group 8 as it showed unique consumption load.

5.6.1.3 GREEND Dataset Results Summary

From both methods (mean-shift and tree clustering), we can predict that appliances fall into the same category can have similar characteristics which could make them hard to be classified by an NALM algorithm without training. It can be seen from

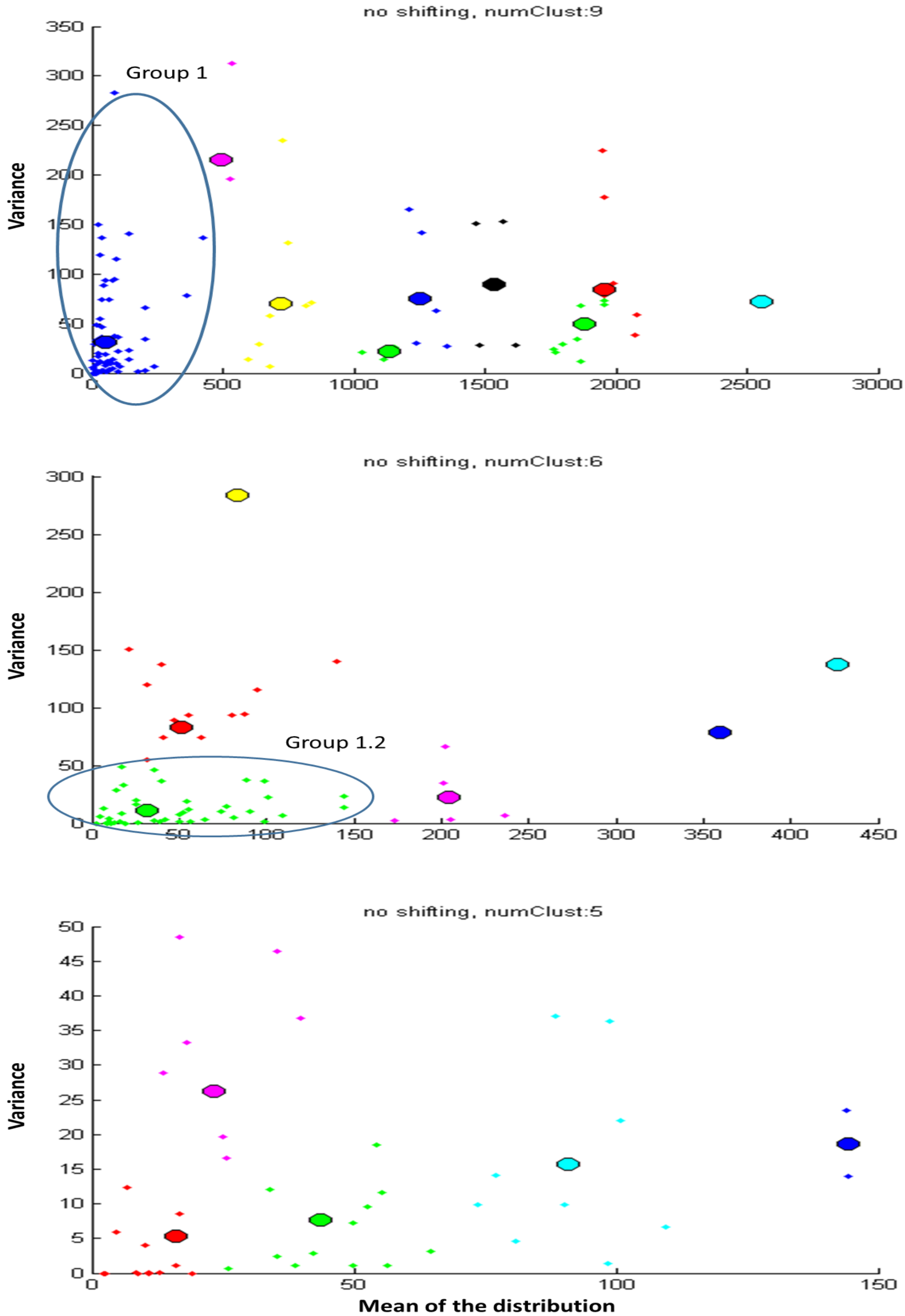


Figure 5.13: Clustering of GREEND dataset into groups and sub-groups using mean-shift algorithm. Upper figure has 9 groups. Middle figure has 6 groups. Lower figure has 5 groups. x-axes is Power in [W]

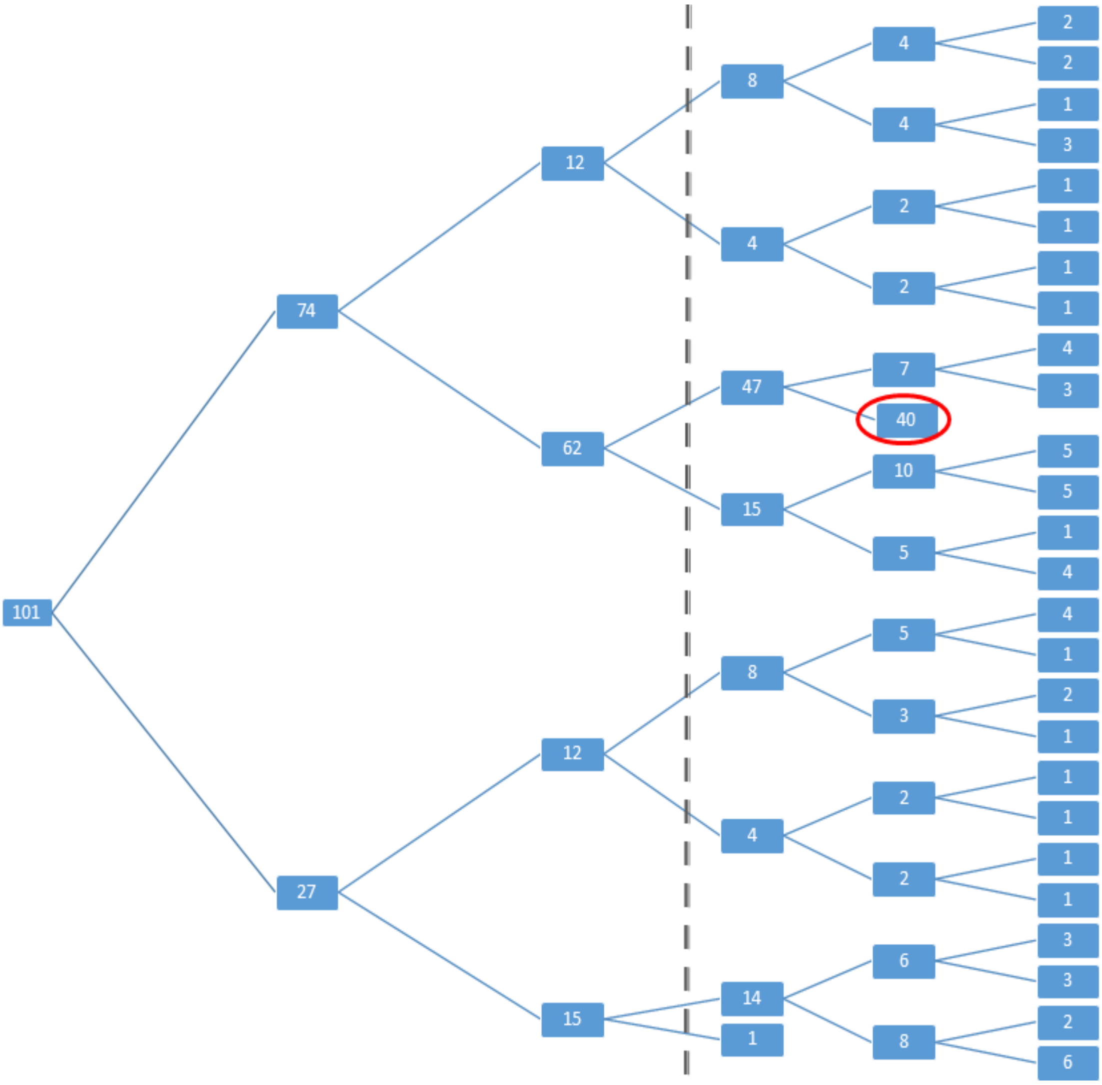


Figure 5.14: Classification for GREEND dataset using tree clustering-part 1. Nodes show number of appliances to be clustered. Dashed line is a suitable level to cut the tree. Red circled node will continue in Part 2, in Figure 5.15.

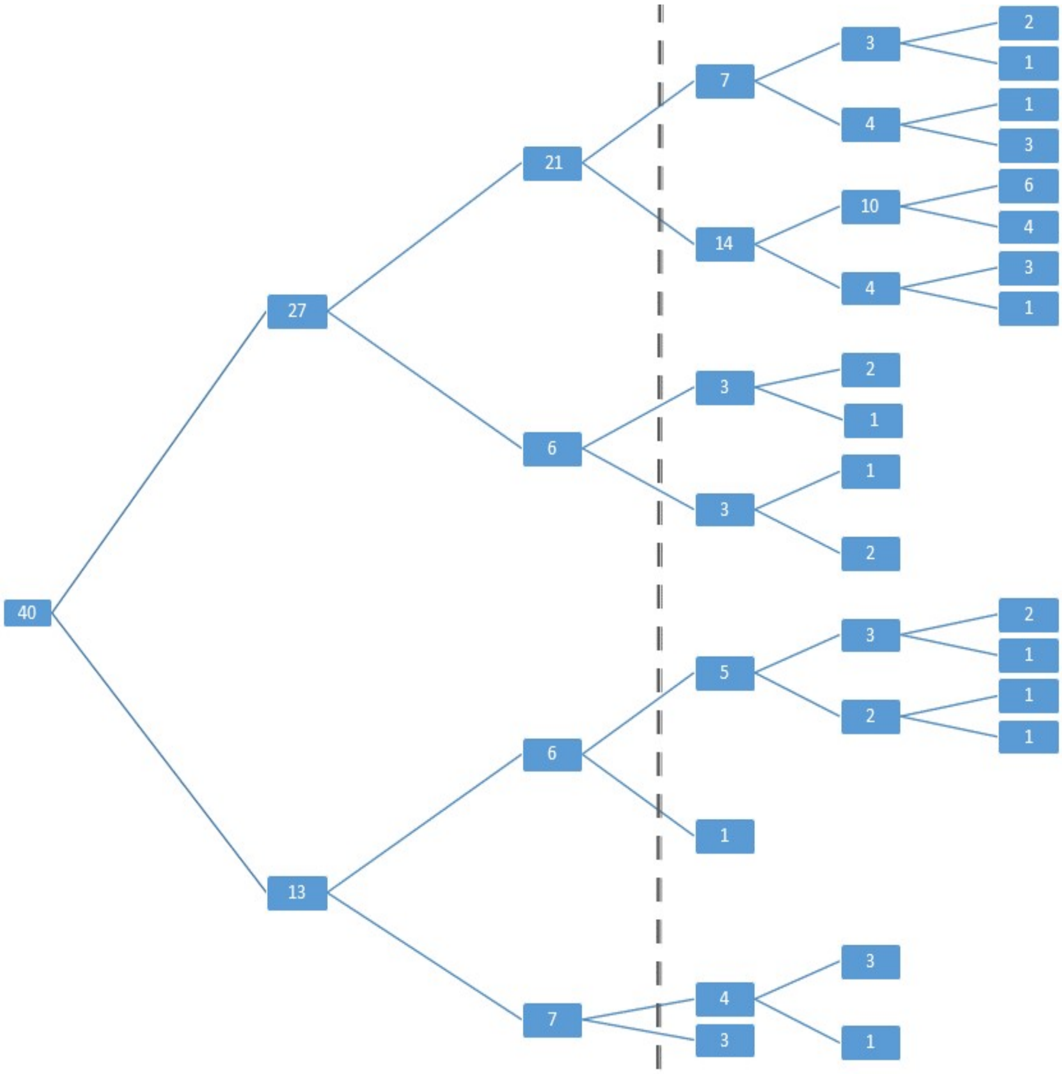


Figure 5.15: Classification for GREEND dataset using tree clustering-part 2. Nodes show number of appliances to be clustered. Dashed line is a suitable level to cut the tree.

Tables B.17 and B.20 that most of appliances first states were clustered into one group due to their mean and variance values being close, that could possibly mean that low states are hard to classify. However, usually in disaggregation tasks, low states are neglected. It can be seen that most washing machine states across all houses are grouped together, which could mean that most washing machines in GREEND dataset show similar behavior in terms of their Gaussian estimation curve. Televisions, do not belong to one group but, all of them are in the low to medium state groups.

We can see from both methods that tumble dryer second state is in its own group with no other appliances which means that it could be easily disaggregated due to its signature being unique. The same can be said about coffee machine and fridge freezer second states. However, the first state of fridge-freezer and freezer states are in the same groups which can lead for them being misclassified. Dishwasher and washing machine second states were categorized into the same groups which means that it is hard to classify them unless other appliances in the same houses were different in terms of their signatures. We can also predict that dishwasher in Houses 2 and 3 can be hard to classify if washing machine second state is present as they are clustered in the same group but the first states can be correctly classified. Washing machine in Houses 1 and 2 can be mixed with the microwave and tumble dryer second states but the first states of the appliances are different, which can help disaggregate them correctly.

5.6.1.4 REFIT Dataset Results Summary

From both methods, Houses 2 clustering shows that toaster will easily be misclassified with other appliances, as its states being spread across most of the groups. Also microwave and kettle will be mixed with toaster third and fourth states. Second states of dishwasher and washing machine are in the same groups, which could mean that it will be hard to correctly classify them.

In House 3, tumble dryer, dishwasher and washing machine second states are clustered in the same groups with both methods, which means that it will be hard to correctly classify them unless other operation states were significantly different. However, the tumble dryer first state is also similar to the washing machine first state.

5.7 Chapter Summary and Conclusion

In this chapter, we created a database of signatures that is based on Gaussian Distribution Estimation. The database includes appliances modeled from GREEND and REFIT datasets. We also gave a deep insight into GREEND dataset different appliances and Houses 2 and 3 REFIT dataset and studied similarities and dissimilarities between all different types of appliances. We clustered those appliances into groups and sub-groups using two simple, yet effective techniques: First, we used mean-shift clustering algorithm to section all appliances and then sub-sectioned any relatively big groups into smaller sets of groups. In the second method, we created a hierarchical tree using k-means algorithm, then cut the tree into a proper level to form different groups and sub-groups.

Both methods gave similar classification results for GREEND and REFIT datasets. For the GREEND dataset, it can be noticed from some groups such as group 1.2.3 in mean-shift method is similar to group 3.4 in the tree method. Likewise, group 7 in mean-shift and group 2 in tree method are almost the same. The tumble dryer second state in House 2 was clustered in a separate group in both methods. For the REFIT dataset, both methods gave similar prediction about toaster as it can easily be misclassified with other appliances. Tumble dryer, dishwasher and washing machine could be mixed as their second states have similar Gaussian characteristics.

Mean-shift and Tree clustering methods gave predictions on which appliances will be hard to classify using or algorithms and which appliances show unique signatures. In the next chapter, we use Gaussian signatures to test NALM algorithms and test clustering predictions.

Chapter 6

Unsupervised Methods and Performance

6.1 Introduction

Our proposed disaggregation algorithms were tested using training data sets from specific REDD dataset houses in Chapter 4, and then a signature database was created in Chapter 5. This chapter is based on published material in [132], where the signatures database is used to develop novel approaches that require no training from the household, and hence no input from the household occupier which is considered a supervised approach. Unsupervised approaches require no input from consumers which is more practical as explained in Section 2.5. Here, we propose and test Unsupervised methods using our Linear and Gaussian combined algorithms, compare performance with Trained k-means, Linear SVM and Gaussian SVM. Finally, we benchmark with the HMM-based approach.

6.2 Novel Contributions

Novel contributions of this chapter are:

- A low-complexity NALM approach that uses the developed database for training, irrespective of the house, and hence does not require customer input; using

House-agnostic training data and compared with House-specific training data approach.

- Detailed experimental evaluation using open source database from Italy and Austria (GREEND dataset) [40].
- Detailed experimental evaluation using open source database from UK (REFIT dataset) [41].
- Comparison with state-of-the-art (HMM)-based method.

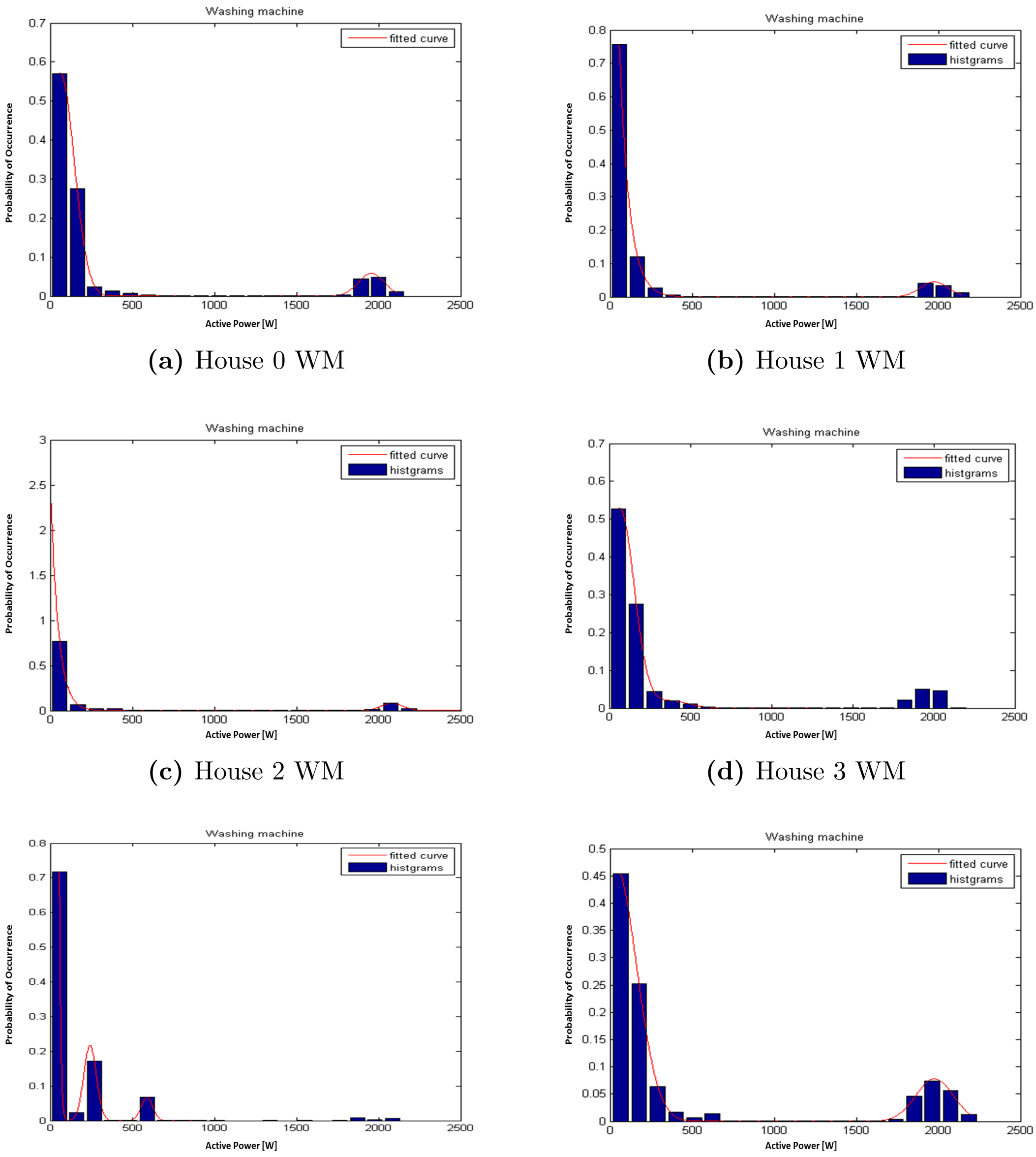
6.3 Unsupervised methods

In this section, we introduce two new *Unsupervised* NALM methods. These methods use the signatures generated by Gaussian statistical curve that we introduced in Chapter 5. The first method, called General Modeling, uses a *universal* model that can possibly fit all appliances of the same type, such as, dishwashers and washing machines. The second method, called Gaussian Unsupervised Method, draws data samples from the created database assuming Gaussian load distribution model. Then, one can use features such as maximum and minimum power, duration of an operation state or the area under the curve to maximize the classification algorithms performance.

6.3.1 Unsupervised General Approach

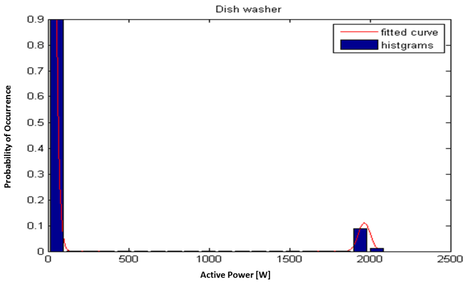
In this section, we turn to providing a general statistical model for one type of appliances. A general statistical model can be generated simply by selecting same type of appliances in different houses provided in a dataset. Then, we develop a Gaussian mixture model as explained in Section 5.3 with samples from different appliances of the same type. For example, in order to provide a general model for toaster, different toasters data samples can be modeled together, using Gaussian statistical model.

While this kind of generalization will not work for TVs, refrigerators and coffee makers, as discussed in Section 5.3 as similar appliances of these types showed majorly different operation behavior, all tested washing machines and dishwashers have similar signatures. Figures 6.1, 6.2, 6.3 and 6.4 show the Gaussian mixture distribution model for the washing machine and dishwasher obtained using the data from all GREEND houses and Houses 1 to 7 in REFIT dataset. It can be seen that an efficient general model can be formed that represents well different appliance brands as shown in Tables 6.1, 6.2, 6.3 and 6.4 since the root-mean-square error (RMSE) obtained in this way is still small across all tested GREEND houses. Some other appliances were modeled in Appendix B using REFIT dataset.

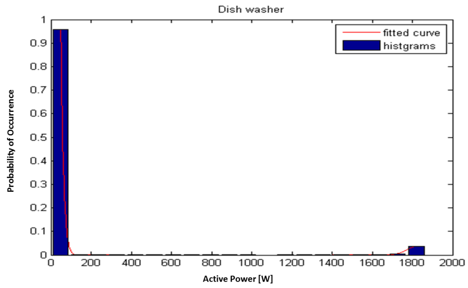


(a) House 0 WM
(b) House 1 WM
(c) House 2 WM
(d) House 3 WM
(e) House 4 WM
(f) WM General Model

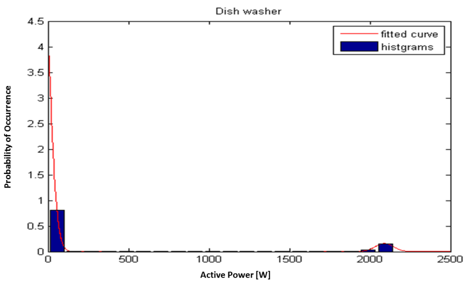
Figure 6.1: Different distributions of different Washing machines types from the GREEND dataset. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].



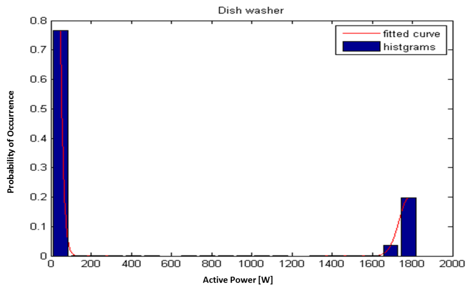
(a) House 0 DW



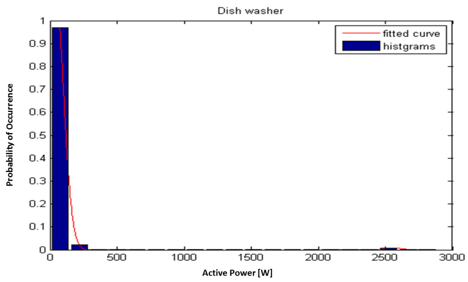
(b) House 1 DW



(c) House 2 DW



(d) House 3 DW



(e) DW General Model

Figure 6.2: Different distributions of different Coffee Makers types from the GREEND dataset. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

The first proposed unsupervised method, called Unsupervised Training-less General Model approach, uses only mean and variance of the generated general Gaussian mixture model, without any sampling or training. Mean and variance are used as classification features for k-means and Linear SVM, Gaussian SVM, Linear combined and Gaussian combined algorithms. This is a very simplistic approach that is not expected to perform well, due to statistical similarities of appliance signatures. Testing data goes through event detection and each event is treated as a single state appliance with a mean and variance values which then are matched to mean and variance of the general model.

Table 6.1: RMSE, mean, variance, 1st order correlation coefficient and 2nd order correlation coefficient for different washing machines of different GREEND houses. G denotes the general model.

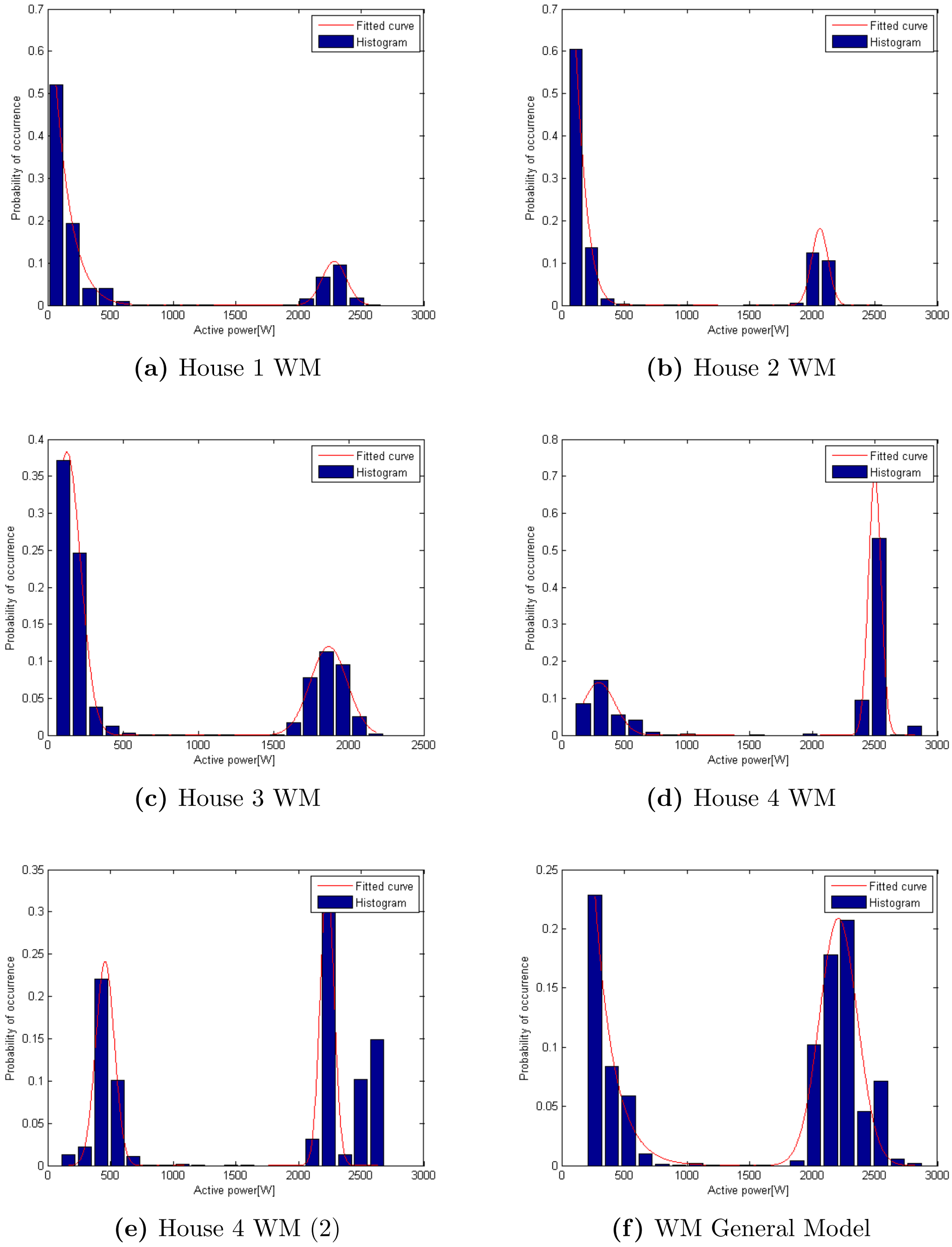
Appliance	Mean value	Variance	RMSE	1st Cor.	2nd Cor.
H0 WM	80.1 1955.6	93.8 73.07	4.2 E-3	0.0751	-0.1059
H1 WM	40.4 1991.7	73.97 90.92	3.8 E-3	0.022	-0.1126
H2 WM	47.2 2081	88.84 58.3694	7.20 E-03	0.082	0.0317
H3 WM	94.7 1957.8	115.59 69.51	1.91 E-2	0.0609	-0.0838
H4 WM	54.9 597.4 1946.1	93.71 13.77 224	3.3 E-3	0.0298	-0.2105
G WM	139.2 2009.3	119.93 90.45	3.2 E-3		

Table 6.2: RMSE, mean, variance, 1st order correlation coefficient and 2nd order correlation coefficient for different dishwashers of different GREEND houses. G denotes the general model.

Appliance	Mean value	Variance	RMSE	1st Cor.	2nd Cor.
H0 DW	77 1953.3	14.05 77.8	6.17 E-05	0.1925	0.1226
H1 DW	13.7 1796	28.9 29.54	4.5 E-05	-0.042	-0.1111
H2 DW	18.1 2071.3	33.19 38.79	9.97 E-05	-0.066	-0.1186
H3 DW	39.9 1760.7	36.8 24.1	1.15 E-04	0.2213	0.1674
G DW	48 2480	42.56 368.2	6.7 E-03		

6.3.2 Gaussian Unsupervised Method

The second Unsupervised proposed approach is called Gaussian Unsupervised method. This approach uses features derived from the Gaussian distribution models and draws random samples from the Gaussian distribution to train our classification algorithms;



(a) House 1 WM (b) House 2 WM

(c) House 3 WM (d) House 4 WM

(e) House 4 WM (2) (f) WM General Model

Figure 6.3: Different distributions of different washing machine types from the REFIT dataset. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

Trained k-means, Linear SVM, Gaussian SVM, Linear combined and Gaussian combined algorithm. Features such as *Maximum power value*, *Minimum power value*, *Duration* and so forth are calculated similar to Supervised method explained in Chapter 3. Different dimensional feature combinations are then fed to classifiers in the train-

Table 6.3: RMSE, mean, variance, 1***st order correlation coefficient and*** 2***nd order correlation coefficient for different washing machines of different REFIT houses. GM denotes the general model.***

House	Mean value	Variance	RMSE	1st Cor.	2nd Cor.
H1	116.8 2274.7	131.91 99.01	0.0085	-0.0701	-0.0059
H2	135.9	71.53	8.63 E-04	0.082	0.2206
H3	164.4 1865.2	79.50 110.27	0.0046	-0.002	-0.0765
H4	344.6 2513.1	170.84 77.56	0.00117	0.0699	0.0068
H4	448.3 2358.5	121.09 182.5181	3.3 E-3	-0.0754	-0.0105
GM	225 2209	56 213.3	0.0206		

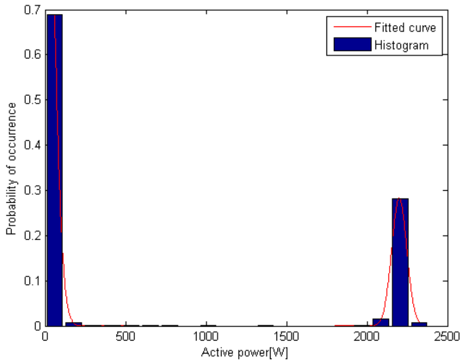
Table 6.4: RMSE, mean, variance, 1***st order correlation coefficient and*** 2***nd order correlation coefficient for different dishwashers of different REFIT houses. GM denotes the general model.***

House	Mean value	Variance	RMSE	1st Cor.	2nd Cor.
H2	64.6 2203.5	21.96 35.63	4.56 E-05	0.1755	0.0359
H3	130.6 2098.3	8.87 72.41	0.002	0.0022	0.0226
H5	2297.5	106.36	2.70 E-03	0.082	0.0317
H6	65.4 2179.9	11.02 46.11	5.64E-05	0.0699	0.3838
GM	152 2141	419.5 152.7	0.032		

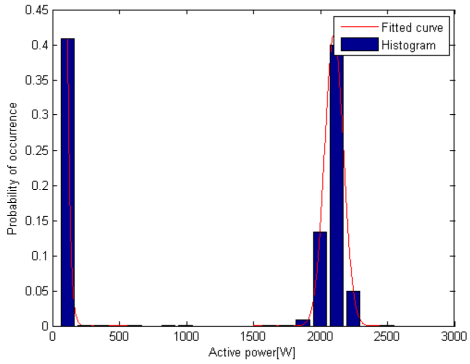
ing and testing steps. Testing data, goes through event detection and each event is treated as a single state appliance and therefore same features are calculated per event and multi-dimensional data are created and matched to features derived from Gaussian distribution estimation.

6.4 Illustration of The Unsupervised Methods

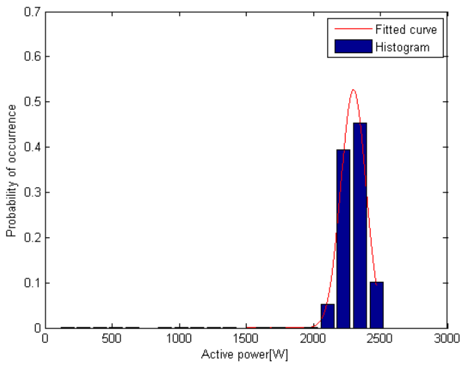
Next, we demonstrate the two proposed methods on a simple case study using Houses 5 and 17 of REFIT dataset. Three appliances were chosen, as they are usually involved



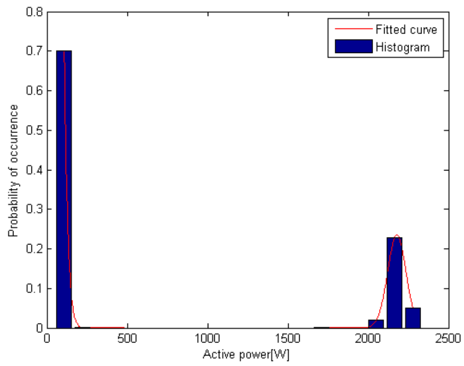
(a) House 2 DW



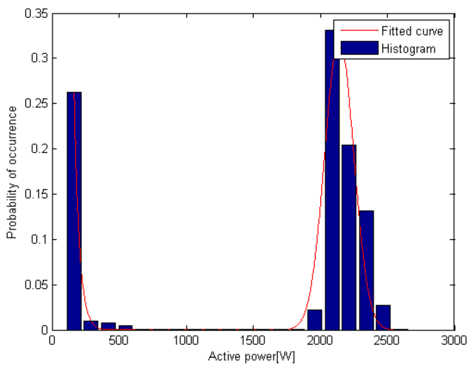
(b) House 3 DW



(c) House 5 DW



(d) House 6 DW



(e) DW General Model

Figure 6.4: Different distributions of different dishwasher types from the REFIT dataset. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

in daily activities, which are: tumble dryer, television and toaster. We used ten days for testing. Testing events are detected based on pre-set thresholds as explained in Chapter 3. We have extracted Gaussian approximation curve for each appliance, calculated mean and variance, and stored the values as shown in Figure 6.5 and

Table 6.5. Then, we label training data based on each method. Note that we model television and toaster and label them as *other*.

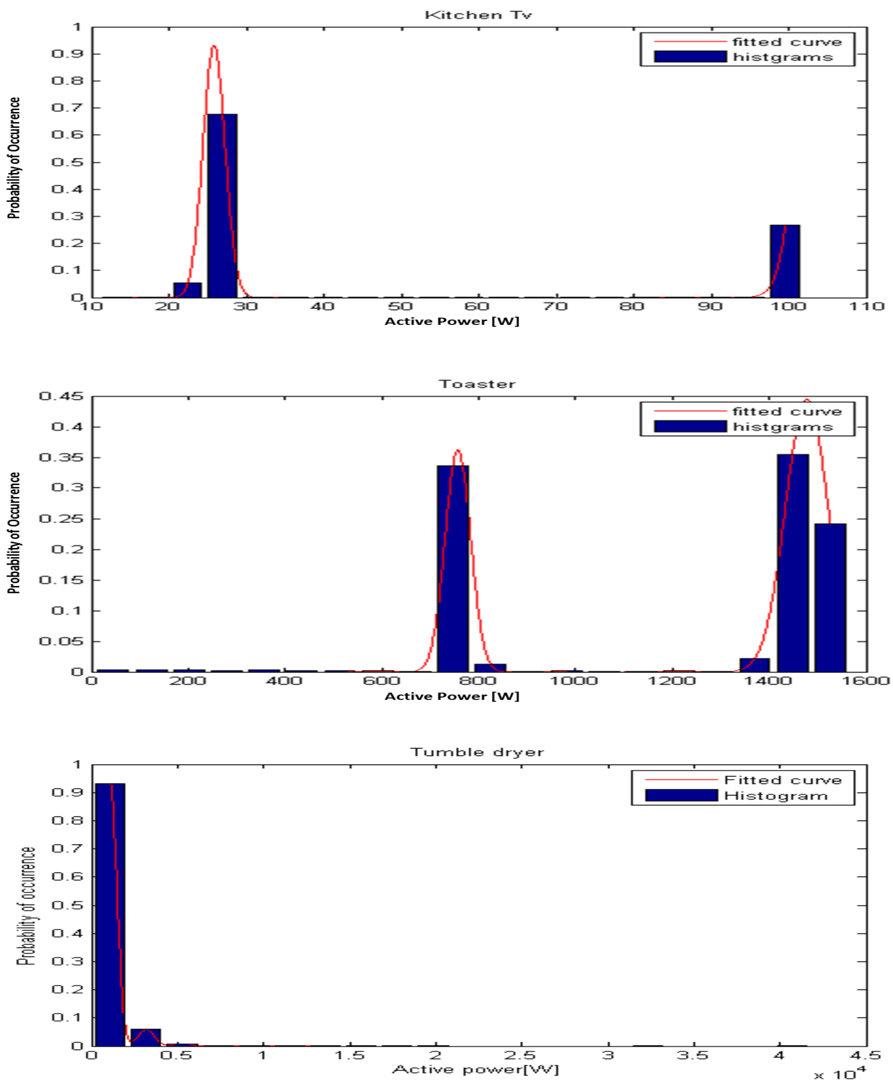


Figure 6.5: Gaussian models for television, toaster and tumble dryer in REFIT house 5.

6.4.1 General Modeling Method Training and Testing Steps

First, a general model for tumble dryer is generated using REFIT dataset House 17 in order to use it for training as shown in Table 6.5 and Figure 6.6; television and toaster are treated as one appliance *'other'*.

Linear and Gaussian combined algorithms train and test in three steps as explained

Table 6.5: RMSE, mean, variance for different appliances of television, toaster and tumble dryer in REFIT dataset House 5. General models are generated using REFIT dataset House 17. GM denotes General Model.

Appliance	Mean value [W]	Variance	RMSE
TV	26.07	0.69	7.90 E-07
	98.61	1.39	
Toaster	747.5	131.55	3.6 E-06
	1476.7	27.66	
Tumble dryer	1107	535.2	1.165 E-04
	3190	642	
	5274	772.3	
	7358	975.9	
Tumble dryer GM	1664	778.2	6.78 E-06
	8315	1667	
	4989	3055	
	6049	1210	

in Chapter 3. First, Trained k-means uses all training and testing points to detect all possible appliances as illustrated in Figures 6.7 and 6.8 (top figures). Second, we apply a radius on all cluster heads as explained in Figure 3.6 which is fixed to all known appliances within the same house. All testing points fall within that threshold will be removed then remaining of testing points will be fed to Linear and Gaussian SVM classifiers. Note here we do not remove any training points. Finally, Linear and Gaussian SVMs will train and test as explained before which is shown in Figures 6.7 and 6.8 (bottom figures).

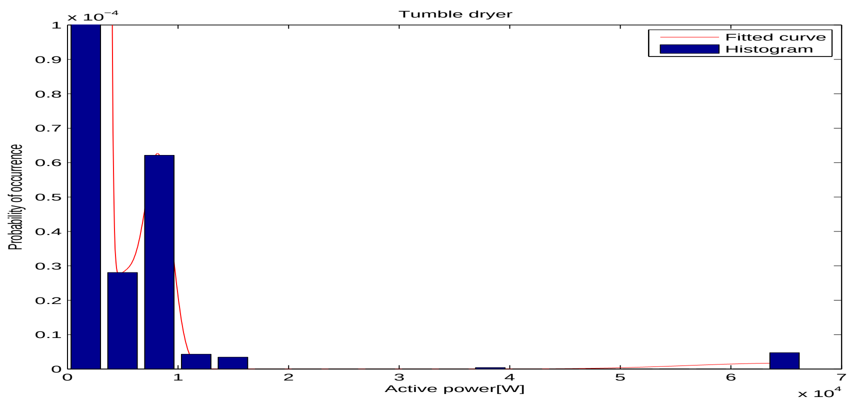
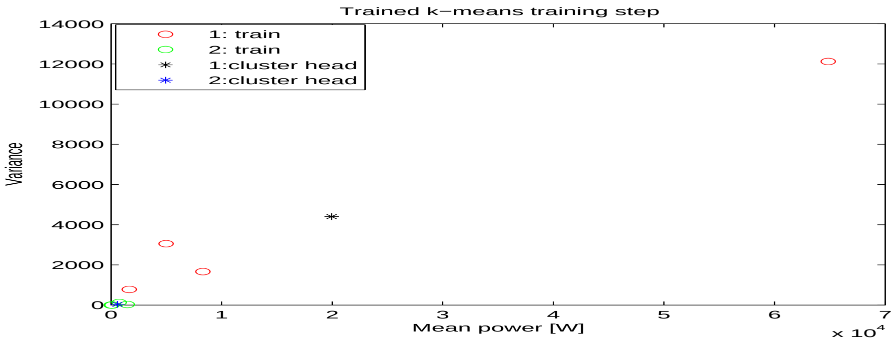
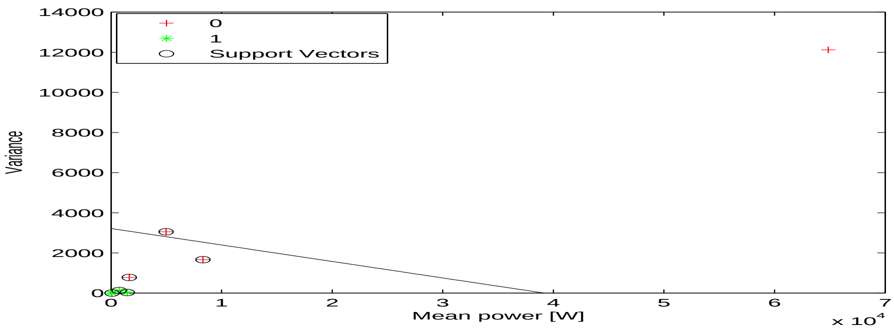


Figure 6.6: Tumble dryer general model using REFIT dataset House 17.



(a) k-means training

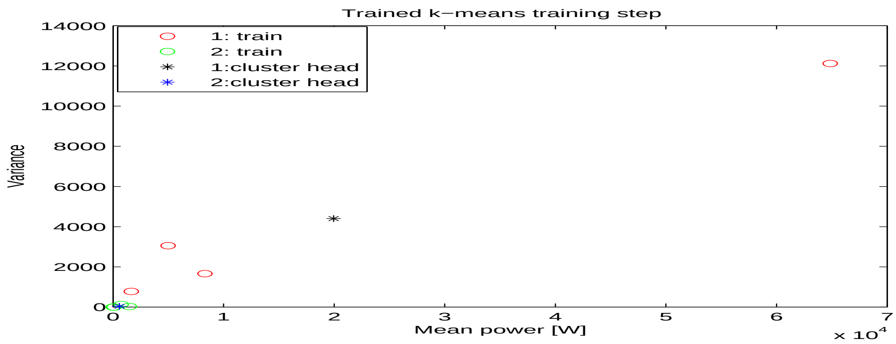


(b) Linear SVM training

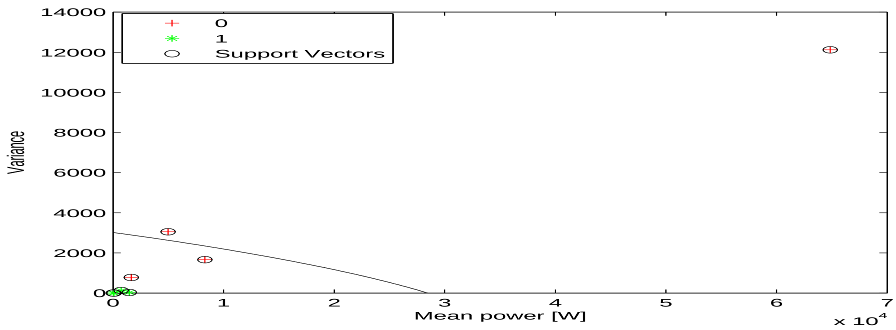
Figure 6.7: Simple case study to detect tumble dryer using General modelling method using Linear combined algorithm. Training points in green are the current class, red points are every other appliance, black circles are support vectors used to separate the two classes.

6.4.2 Gaussian Unsupervised Method Training and Testing Steps

Using Gaussian signatures extracted in the signature database, we derived samples from the Gaussian distribution models and used them to extract features to train our classifiers. 2-Dimensional, 3-Dimensional, 4-Dimensional and all 5 Dimensional feature combinations are formed and used to test our combined algorithms. Features are maximum value, minimum value, max/mean ration, duration of an statistical state and area under curve. Then we train and test our classifiers as shown in Figures 6.9 and 6.10. First, Trained k-means uses all training and testing points to detect all possible appliances as illustrated in top figures. Second, we apply a radius on all cluster heads as explained before which is fixed to all known appliances within the same house. All training and testing points fall within that threshold will be removed and remaining of training and testing points will be fed to Linear and Gaussian SVM



(a) k-means training



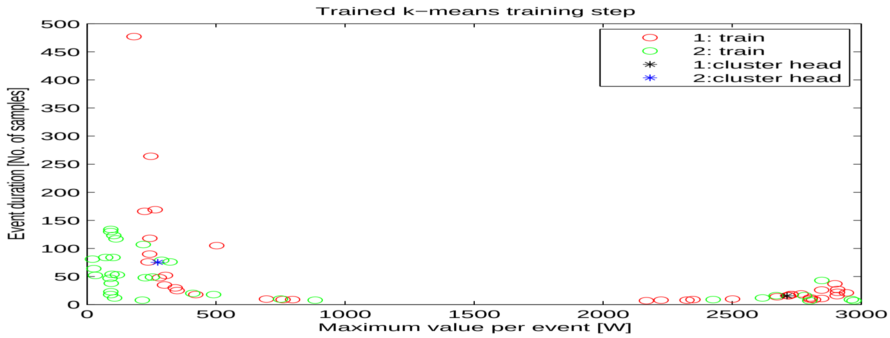
(b) Gaussian SVM training

Figure 6.8: Simple case study to detect tumble dryer using General modelling method using Gaussian combined algorithm. Training points in green are the current class, red points are every other appliance, black circles are support vectors used to separate the two classes.

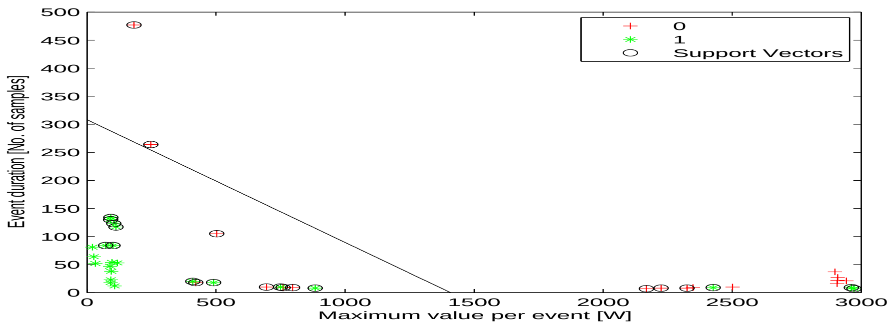
classifiers. Finally, Linear and Gaussian SVMs will train and test the remaining of training and testing data as in the figures below.

6.4.3 Disaggregation Case Study Results

The outcome of classifiers is a 1D vector of expected labels that corresponds to testing events. After that, we use confusion matrix to compare correct labels that we have generated in the training step with the expected ones to form True Positive, False Positive and False Negative events and we also calculate Precision, Recall and F-Measure. Results of both methods in detecting the tumble dryer are presented in Table 6.6. It can be seen that both algorithms performed poorly in detecting tumble dryer as it was mixed with washing machine signature which is expected based on Chapter 5 clustering methods. By other words, tumble dryer signature was not unique enough in this house to correctly disaggregate it.



(a) k-means training



(b) Linear SVM training

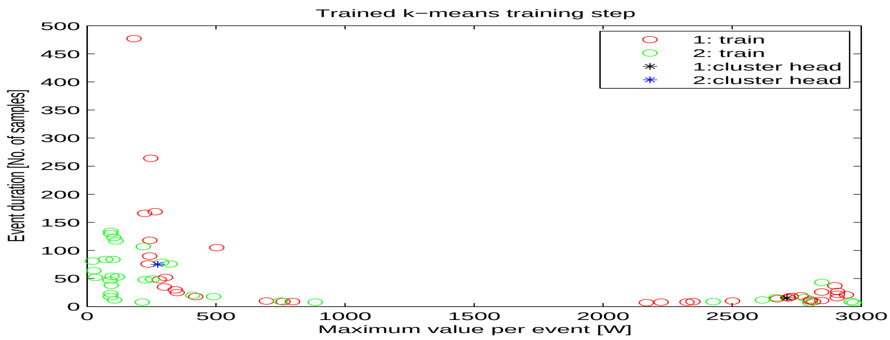
Figure 6.9: Simple case study to detect tumble dryer using Unsupervised Gaussian method using Linear combined algorithm. Training points in green are the current class, red points are every other appliance, black circles are support vectors used to separate the two classes.

6.5 Results and Discussion

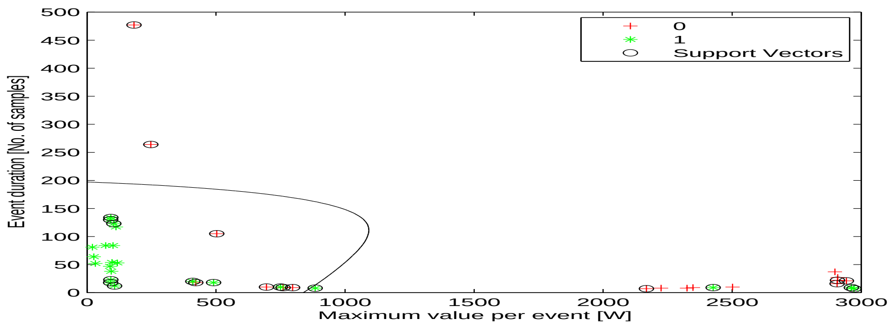
Note that both Unsupervised approaches can be used with different event-based supervised algorithms. That is, the designed general appliance model can replace training. We test both supervised and unsupervised approaches next. Then, we benchmark the results against the state-of-the-art HMM using GREEND and REFIT datasets.

We used washing machine and dishwasher in GREEND dataset in Houses 1, 2 and 3. Washing machine and dishwasher were selected, in our experiments, since these are the only two appliances present in at least 4 GREEND dataset houses, and also known to be main electricity consumers.

GREEND House 1 has 8 appliances operating: fridge, dishwasher, microwave, water kettle, washing machine, radio with amplifier, dryer and bedside lamp. House 2 has 8 appliances operating: TV, Network Access Storage (NAS), washing machine,



(a) k-means training



(b) Gaussian SVM training

Figure 6.10: Simple case study to detect tumble dryer using Unsupervised Gaussian method using Gaussian combined algorithm. Training points in green are the current class, red points are every other appliance, black circles are support vectors used to separate the two classes.

dryer, dishwasher, notebook, coffee machine and bread machine. House 3 has 9 appliances operating: entrance outlet, dishwasher, water kettle, fridge with freezer, washing machine, hair dryer, computer, coffee machine and TV.

We also used REFIT dataset Houses 2 and 3. In House 2, there are 9 'known' appliances present. However, only seven were chosen in our experiments, that are present in at least four houses, which are: Fridge-Freezer, Washing Machine, Dishwasher, Television, Microwave, Toaster, Kettle. Other appliances were treated as *"unknown"*, and hence they contribute to noise. Unknown appliances are usually appliances that changed signature or combined with other appliances in the same monitor.

In House 3, there are 9 'known' appliances present, which are: Toaster, Fridge-Freezer, Freezer, Tumble Dryer, Dishwasher, Washing Machine, Television, Microwave, Kettle.

Table 6.6: Results of case study to detect tumble dryer using Linear combined algorithm and Gaussian combined algorithm using Precision, Recall and F_m after disaggregation using REFIT dataset House 5.

Method			*tumble dryer*	*other*
			(%)	(%)
General modelling	*L combined*	Pr	0	81.15
		Re	0	100
		F_m	0	89.59
	G combined	Pr	0	81.15
		Re	0	100
		F_m	0	89.59
Gaus Unsupervised	*L combined*	Pr	20	92.41
		Re	8.1	92.41
		F_m	11.59	86.47
	G combined	Pr	32.35	83.18
		Re	22.44	89.09
		F_m	26.5	86.04

6.5.1 Time Complexity

In this sub-section, we compare supervised and unsupervised approaches using time complexity. Complexity is measured by execution time in seconds but we only monitor testing simulation time here, as training execution time for unsupervised methods is considered to be zero seconds.

6.5.1.1 Unsupervised General Method

Figure 6.11 shows testing execution times for washing machine and dishwasher in GREEND dataset Houses 1, 2 and 3 for Supervised and Unsupervised General modeling methods methods using Trained k-means, LSVM, GSVM, Linear combined algorithm and Gaussian combined algorithm. It can be seen that Trained k-means was the fastest in detecting our tested appliances as expected with less than 0.1 seconds in both methods. Linear SVM and Gaussian SVM are the slowest compared to other disaggregating algorithms in the figure. Our combined algorithms are slower than k-means, yet, much faster than linear and Gaussian SVM in both methods. These results are similar findings to Section 4.4.

It can also be seen that Unsupervised General modeling method has lower execution times using all tested classification algorithms by almost a half compared to Supervised method results. Interestingly, Gaussian-based (Gaussian SVM and Gaussian combined) algorithms are slower than Linear-based (Linear SVM and Linear combined) algorithms.

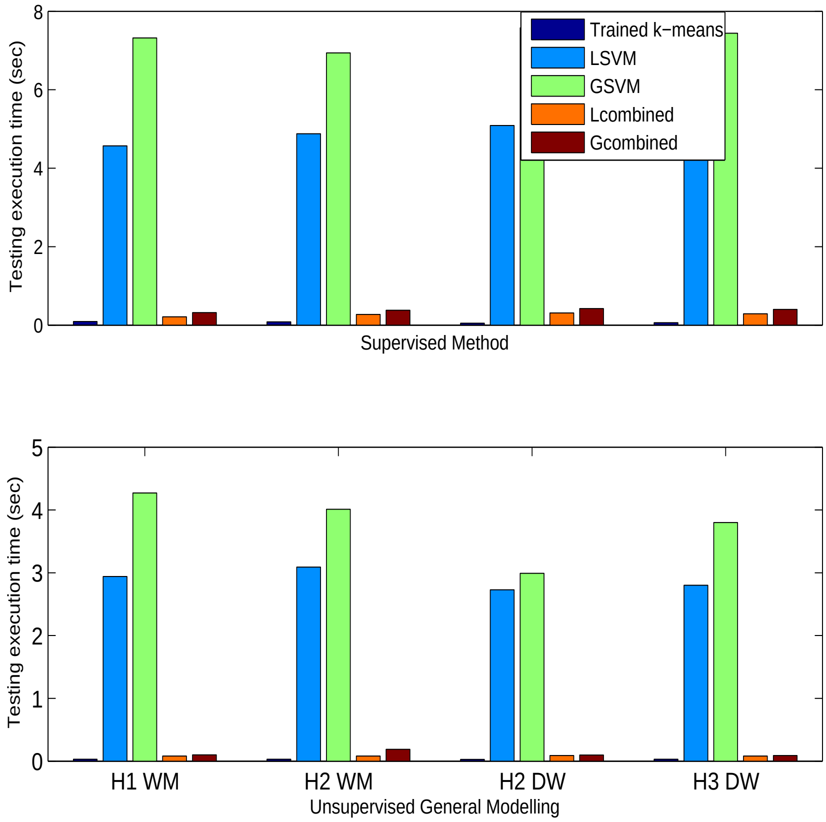


Figure 6.11: Testing execution times results for Supervised and Unsupervised General modeling methods using Trained k-means, LSVM, GSVM, Linear combined algorithm and Gaussian combined algorithm to detect for washing machine and dishwasher in GREEND dataset Houses 1, 2 and 3. x-axis shows testing execution time in [sec].

Figure 6.12 shows testing execution times for REFIT Houses 2 and 3 for Supervised and Unsupervised General modeling methods using Trained k-means, LSVM, GSVM, Linear combined algorithm and Gaussian combined algorithm. Houses 1 to 7 were used to build models for the target appliances. It can be seen that Trained k-means was the fastest in detecting our tested appliances as expected. Linear SVM and Gaussian SVM are the slowest compared to other disaggregating algorithms with higher execution times of up to roughly 4 and 6 seconds respectively for the Supervised method and 2 and 4.5 seconds using General modeling method. Our combined algorithms are slower than k-means as expected based on similar findings in Section 4.4 but much faster than linear and Gaussian SVM which is also expected regarding Section 4.4.

It can also be seen that Unsupervised General modeling method has lower execution times using all tested classification algorithms by almost a half compared to Supervised method results. Similar to GREEND dataset results, Gaussian-based (Gaussian SVM and Gaussian combined) algorithms are slower than Linear-based (Linear SVM and Linear combined) algorithms.

6.5.1.2 Gaussian Unsupervised Method

Figure 6.13 shows testing execution times for washing machine and dishwasher in GREEND dataset Houses 1, 2 and 3 for Supervised and Gaussian Unsupervised methods using Trained k-means, LSVM, GSVM, Linear combined algorithm and Gaussian combined algorithm. It can be seen that the Gaussian modeling method results by Trained k-means, Linear SVM, Gaussian SVM, Linear combined and Gaussian combined algorithm show a similar behavior to that by the Supervised method. Our combined algorithms are slower than k-means but much faster than linear and Gaussian SVM in both methods with less than 1 second in both training methods which is similar to the findings discussed in Section 4.4.

It can also be seen that Unsupervised Gaussian modeling method has lower execution times compared to that by Supervised method using all tested classification algorithms by one or two seconds roughly in most tested classification algorithms in this table.

Figure 6.14 shows testing execution times for REFIT dataset Houses 2 and 3 for Supervised and Gaussian Unsupervised methods using Trained k-means, LSVM, GSVM, Linear combined algorithm and Gaussian combined algorithm. Houses 1 to 7 were used to build models for the target appliances. It can be seen that the Gaussian modeling method results by Trained k-means, Linear SVM, Gaussian SVM, Linear combined and Gaussian combined algorithm show a similar behavior to that

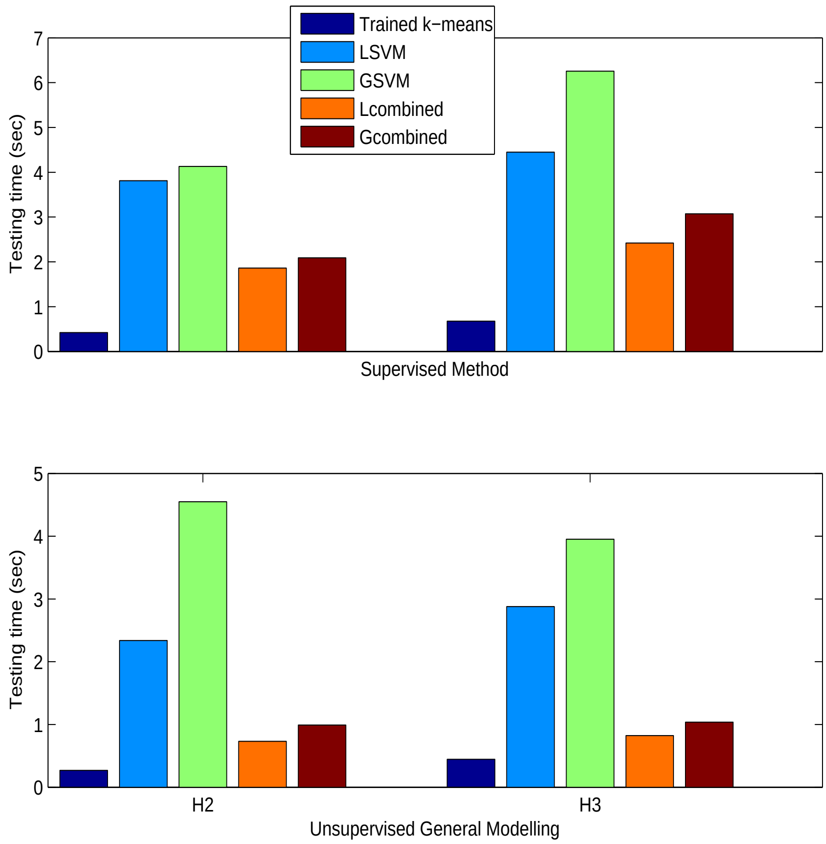


Figure 6.12: REFIT Houses 2 and 3 testing execution times results for Supervised and Unsupervised General modeling methods using Trained k-means, LSVM, GSVM, Linear combined algorithm and Gaussian combined algorithm. x-axis shows testing execution time in [sec].

by Supervised method, and similar pattern to that obtained by GREEND dataset. Trained k-means was the fastest again in detecting our tested appliances as expected, due to its simplicity. Linear SVM and Gaussian SVM are the slowest, due to their complexity, compared to the remaining of disaggregating algorithms.

It can also be seen that Unsupervised Gaussian modeling method has lower execution times compared to that by Supervised method using all tested classification algorithms by one or two seconds roughly in most tested classification algorithms.

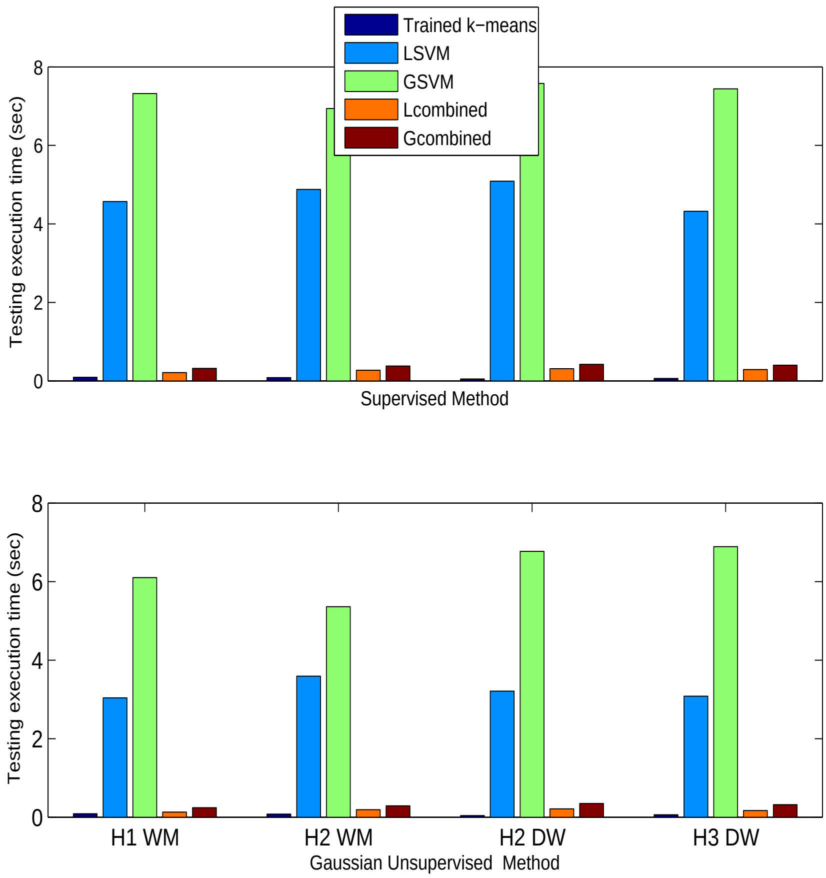


Figure 6.13: Testing execution times results for Supervised and Unsupervised Gaussian model methods using Trained k-means, LSVM, GSVM, Linear combined algorithm and Gaussian combined algorithm to detect washing machine and dishwasher in GREEND dataset Houses 1, 2 and 3. x-axis shows testing execution time in [sec].

6.5.2 Accuracy

In this sub-section, we compare supervised with unsupervised approaches using algorithm accuracy. Accuracy is measured here using Precision, Recall and F-Measure.

6.5.2.1 General Modeling Method

Tables 6.7 and 6.8 show a comparison between unsupervised General method and supervised method after disaggregation with Trained k-means, Linear SVM, Gaussian SVM, Linear combined and Gaussian combined using GREEND dataset Houses 1, 2 and 3. Results obtained by the supervised method are much higher that results

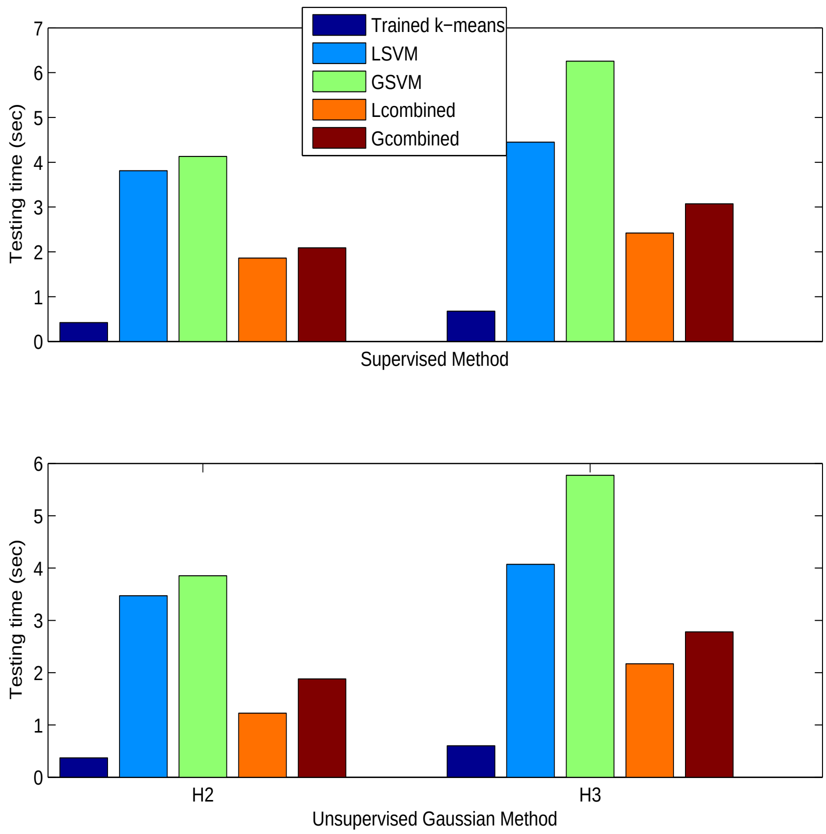


Figure 6.14: Testing execution times results for Supervised and Unsupervised Gaussian methods using Trained k-means, LSVM, GSVM, Linear combined algorithm and Gaussian combined algorithm. x-axis shows testing execution time in [sec].

obtained by the General Model method with up to 78.94% and 94.16% F_m results in detecting washing machine in both Houses 1 and 2 respectively and up to 98.48% and 94.73% F_m results in detecting dishwasher in Houses 2 and 3 respectively. It can also be seen from both tables that detecting washing machine events using unsupervised General model signature did not work but it gave good performance in detecting dishwasher.

The reason for this is that mean and variance of dishwasher is very unique against other appliances in the tested houses as was predicted in Section 5.6.1.3 due to its first state being different that other appliances in the same houses, which is not the case with the washing machine which was mixed with tumble dryer and microwave as predicted in Section 5.6.1.3. This flags up the need for training using the samples

Table 6.7: Results for washing machine disaggregation in GREEND Houses 1 and 2 for two different methods. Houses 3, 4 and 5 are used for training.

		General Modeling			Supervised Method		
H	Method	Pr %	Re %	F_m %	Pr %	Re %	F_m %
H1	Trained k-means	0	0	0	91.2	65.3	76.14
	Linear SVM	0	0	0	63.88	100	77.96
	Gaussian SVM	0	0	0	91.39	66.92	77.27
	Linear Combined	0	0	0	100	65.21	78.94
	Gaussian Combined	0	0	0	63.82	100	77.92
H2	Trained k-means	0	0	0	75	100	85.71
	Linear SVM	6.26	3.33	4.34	83.33	100	90.90
	Gaussian SVM	5.55	3.33	4.16	100	86.61	92.82
	Linear Combined	9.09	3.33	4.87	87.50	93.33	90.32
	Gaussian Combined	5.55	4.33	4.87	100	88.97	94.16

Table 6.8: Results for dishwasher disaggregation in GREEND Houses 2 and 3 for two different methods. Houses 1, 4 and 5 are used for training.

		General Modelling			Supervised Method		
H	Method	Pr %	Re %	F_m %	Pr %	Re %	F_m %
H2	Trained k-means	100	85.03	91.91	100	86.6	92.82
	Linear SVM	92.29	100	96.26	97.01	100	98.48
	Gaussian SVM	100	92.12	95.9	100	94.87	97.39
	Linear Combined	94.20	100	97.01	97.01	100	98.48
	Gaussian Combined	100	92.91	96.3	100	94.87	97.36
H3	Trained k-means	100	47.24	64.17	68.18	100	81.08
	Linear SVM	83.33	71.42	76.92	87.50	100	93.33
	Gaussian SVM	91.2	65.35	76.14	100	90	94.73
	Linear Combined	83.33	71.42	76.92	87.50	100	93.33
	Gaussian Combined	91.2	65.35	76.14	100	90	94.73

from the database, rather than using the features directly for the classification.

Figure 6.15 shows a comparison between all five tested algorithms in REFIT House 2 using Supervised method and General modeling method. It is obvious that all tested algorithms performed significantly better using Supervised method compared to Unsupervised general modeling method. Washing machine had slightly higher result using Gaussian combined method with 60.22% F-Measure but lower results using rest of algorithms due to its low state values being mixed with fridge-freezer and second state mixed with dishwasher as was predicted in Section 6.5.2.1 using mean-shift and Tree clustering methods. However, fridge-freezer had good performance using Linear combined, Linear SVM and Gaussian SVM with roughly 69% F-Measure. Trained k-means and Gaussian combined gave very high performance detecting the kettle with 100% Precision, but Gaussian combined had low recall which affected its F-Measure to be lower, due to its Gaussian signature being mistaken with dishwasher and microwave which also was predicted in Section 6.5.2.1 due to their signatures were clustered in the same category.

Figure 6.16 shows a comparison between all five tested algorithms in REFIT House 3 using Supervised method and General modeling method. It is obvious that all tested algorithms performed significantly better using Supervised method compared to Unsupervised general modeling method as expected. However, freezer general model was correctly detected by Linear SVM with a high 70% F-Measure and 74.80% using Linear combined, but much lower results by k-means, Gaussian SVM and Gaussian combined algorithm due to its signature values being mixed with fridge-freezer states. Tumble dryer general model, as well, was detectable but its signature was not unique enough as it only gave 40%, 34.4% and 38.8% using Linear SVM, Gaussian SVM and Gaussian combined algorithm respectively. Interestingly, washing machine results were very poor, unlike House2, due to its first state being mixed with fridge-freezer signature, second state was mixed with microwave states and its high states was mixed with tumble dryer as was predicted in Section due to their signatures being clustered in the same groups.

6.5.2.2 Gaussian Unsupervised Method

Tables 6.9 and 6.10 show Precision, Recall and F-Measure results in GREEND dataset Houses 1, 2 and 3 after disaggregation using Trained k-means, Linear SVM, Gaussian SVM, Linear combined and Gaussian combined algorithms. It can be seen that the Gaussian model sampling approach shows competitive performance to that of the supervised combined approach.

Washing machine results for Houses 1 and 2 are presented in Table 6.9 using Precision, Recall and F-Measure for both methods. It can be seen that all tested algorithms performed well. Trained k-means generally performed slightly lower than other algorithms as expected. However, other tested algorithms gave up to 76.92%

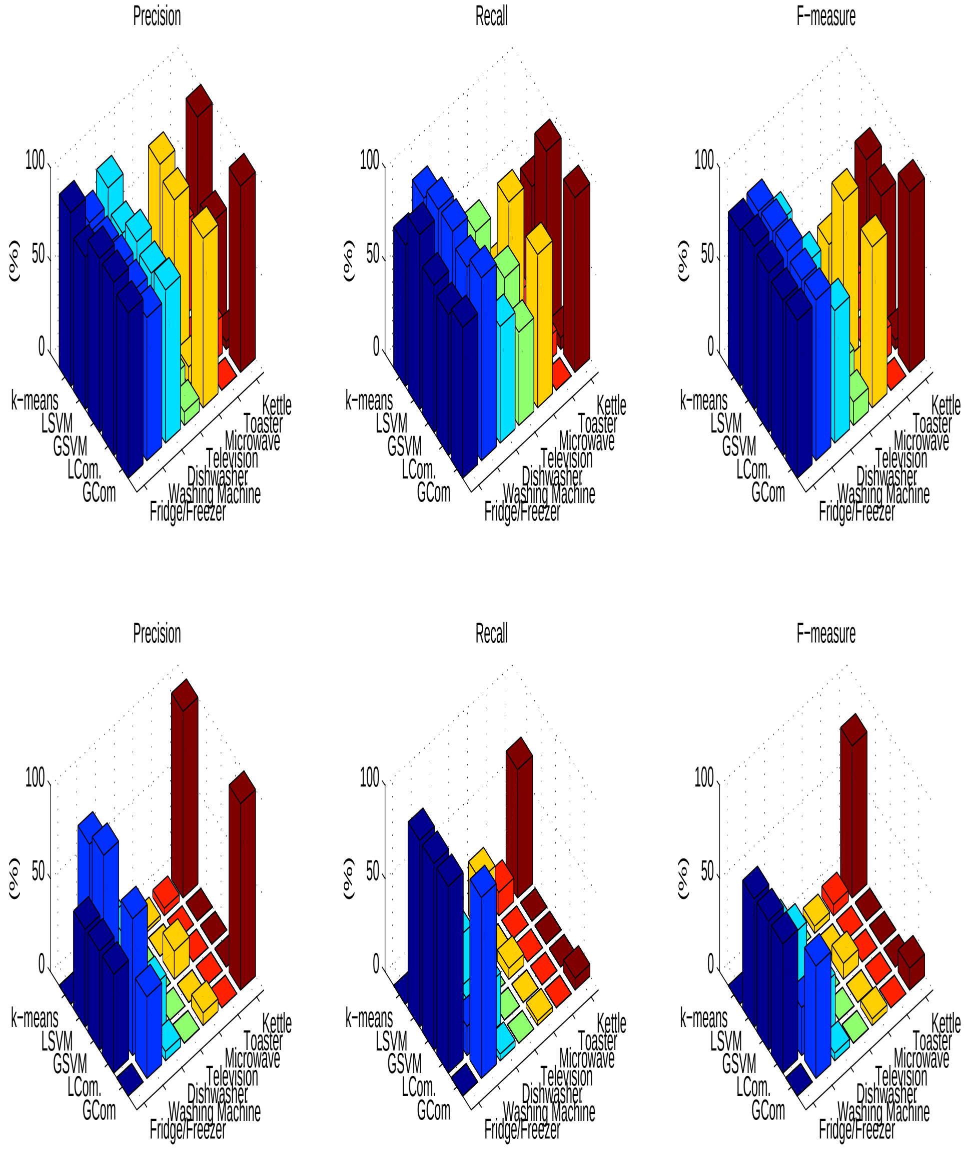


Figure 6.15: Disaggregation results for REFIT Houses 2 Supervised (Top figures) and Unsupervised General modelling (Bottom figures) methods using Trained k-means, LSVM, GSVM, Linear combined algorithm and Gaussian combined algorithm.

and 78.94% F_m by Gaussian modeling and the supervised method respectively in House 1, and up to 93.75% and 94.16% F_m in House 2 of Gaussian modeling and supervised method respectively.

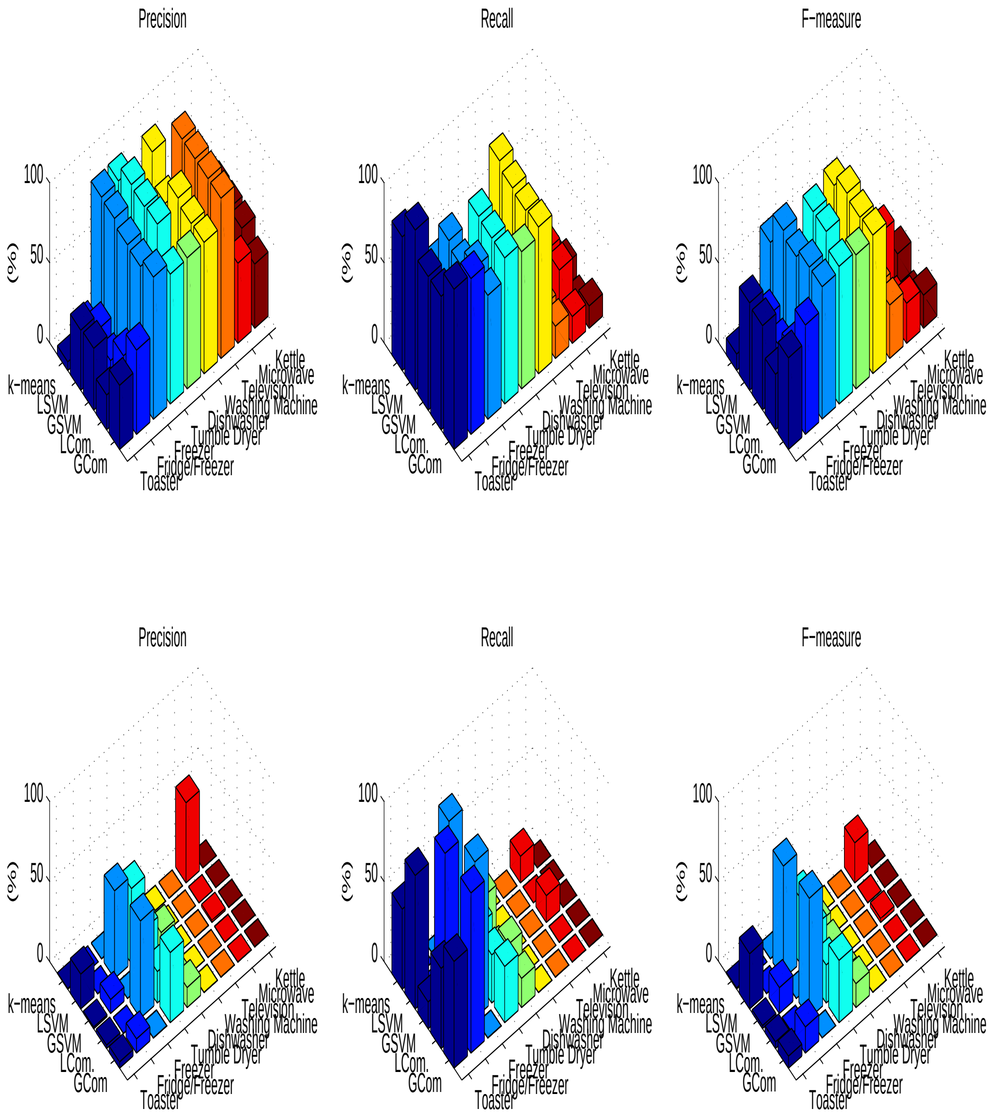


Figure 6.16: Disaggregation results for REFIT Houses 3 Supervised (Top figures) and Unsupervised General modeling (Bottom figures) methods using Trained k-means, LSVM, GSVM, Linear combined algorithm and Gaussian combined algorithm.

Similarly, dishwasher results for Houses 2 and 3 are presented in Table 6.10 using Precision, Recall and F-Measure for both methods. It can be seen that all tested algorithms performed well. Trained k-means generally performed 5-10% lower than other algorithms. However, all other algorithms gave up to 91.04% and 89.48% F_m by

Gaussian modeling and supervised method respectively in House 2, and up to 94.73% and 94.73% F_m in House 3 of Gaussian modeling and supervised method respectively. It can also be seen that Linear combined gave similar results to that of Linear SVM and Gaussian combined algorithm showed similar performance to that of Gaussian SVM.

Table 6.9: Results of washing machine disaggregation in GREEND Houses 1 and 2 for two different methods. Houses 3, 4 and 5 are used for training.

		Gaussian Modelling			Supervised Method		
H	Method	Pr %	Re %	F_m %	Pr %	Re %	F_m %
H1	Trained k-means	61.22	100	75.94	91.2	65.3	76.14
	Linear SVM	78.57	71.73	75	63.88	100	77.96
	Gaussian SVM	99.9	56.81	72.44	91.39	66.92	77.27
	Linear Combined	72.54	80.43	76.28	100	65.21	78.94
	Gaussian Combined	62.5	100	76.92	63.82	100	77.92
H2	Trained k-means	100	72.44	84.01	75	100	85.71
	Linear SVM	83.33	100	90.90	83.33	100	90.90
	Gaussian SVM	100	85.03	91.91	100	86.61	92.82
	Linear Combined	88.23	100	93.75	87.50	93.33	90.32
	Gaussian Combined	100	88.18	93.72	100	88.97	94.16

Table 6.10: Results of dishwasher disaggregation in GREEND Houses 2 and 3 for two different methods. Houses 1, 4 and 5 are used for training.

		Gaussian Modeling			Supervised Method		
H	Method	Pr %	Re %	F_m %	Pr %	Re %	F_m %
H2	Trained k-means	75	100	85.71	100	86.6	92.82
	Linear SVM	92.06	89.23	90.62	97.01	100	98.48
	Gaussian SVM	84.41	100	91.15	100	94.87	97.39
	Linear Combined	88.40	93.84	91.04	97.01	100	98.48
	Gaussian Combined	88.40	93.84	91.04	100	94.87	97.36
H3	Trained k-means	68.18	100	81.08	68.18	100	81.08
	Linear SVM	87.50	100	93.33	87.50	100	93.33
	Gaussian SVM	100	90	94.73	100	90	94.73
	Linear Combined	87.50	100	93.33	87.50	100	93.33
	Gaussian Combined	100	90	94.73	100	90	94.73

Figure 6.17 shows a comparison between all five tested algorithms in REFIT House 2 using Supervised method and Gaussian Unsupervised Method. It is obvious that all

tested algorithms performed very well for both Supervised method and Unsupervised method with above 80% F-Measure in detecting fridge-freezer and washing machine due to their high appearance in testing dataset. Oddly, the unsupervised method outperformed supervised method in few cases; microwave results were better using trained k-means as they increased from zero F-Measure with supervised method to 41.37% F-Measure. Linear SVM detected microwave with 3% F-Measure using supervised method compared to 41.73% F-Measure using unsupervised method. Kettle, as well, was detected better using unsupervised method with 81.48%, 81.48% and 31.57% using Linear SVM, Gaussian SVM and Linear combined algorithm respectively, compared to 76.9%, 69.76% 5% using supervised method by Linear SVM, Gaussian SVM and Linear combined algorithm respectively. Only Gaussian combined algorithm results were better using supervised method in detecting kettle with 96.74% against 81.48% using unsupervised method. However, kettle results could mean that sometimes noisy data like REFIT can benefit from using features derived from Gaussian approximations instead of using training dataset.

Figure 6.18 shows a comparison between all five tested algorithms in REFIT House 3 using Supervised method and Gaussian Unsupervised Method. Supervised method outperformed unsupervised method in most cases with all tested algorithms. However, some appliances were detected better using unsupervised method due to better training quality with Gaussian approximation model. Washing machine results were $F_M = 23.28\%$ using k-means but was about $F_M = 58.61\%$ using unsupervised method. Linear SVM, as well, had some appliances performed better using unsupervised method, freezer had 3% higher F-Measure, going from 80.86% using supervised method to 83.94% using unsupervised method. Tumble dryer has $F_M = 53.21\%$ using supervised method and up to $F_M = 67.76\%$ using unsupervised Gaussian method.

The Linear combined algorithm performed better for the unsupervised method in REFIT House 3 to detect toaster, fridge-freezer, dishwasher and television with 50%, 49.36%, 84.33% and 52.94% F-Measure respectively using unsupervised method compared to 34.4%, 40.94%, 0% and 33.33% F-Measure respectively for the supervised method. The fridge-freezer events were wrongly classified to the freezer events using supervised method, toaster and television were mixed with washing machine low-value events. Dishwasher events were mixed with fridge-freezer and washing machine events using supervised method.

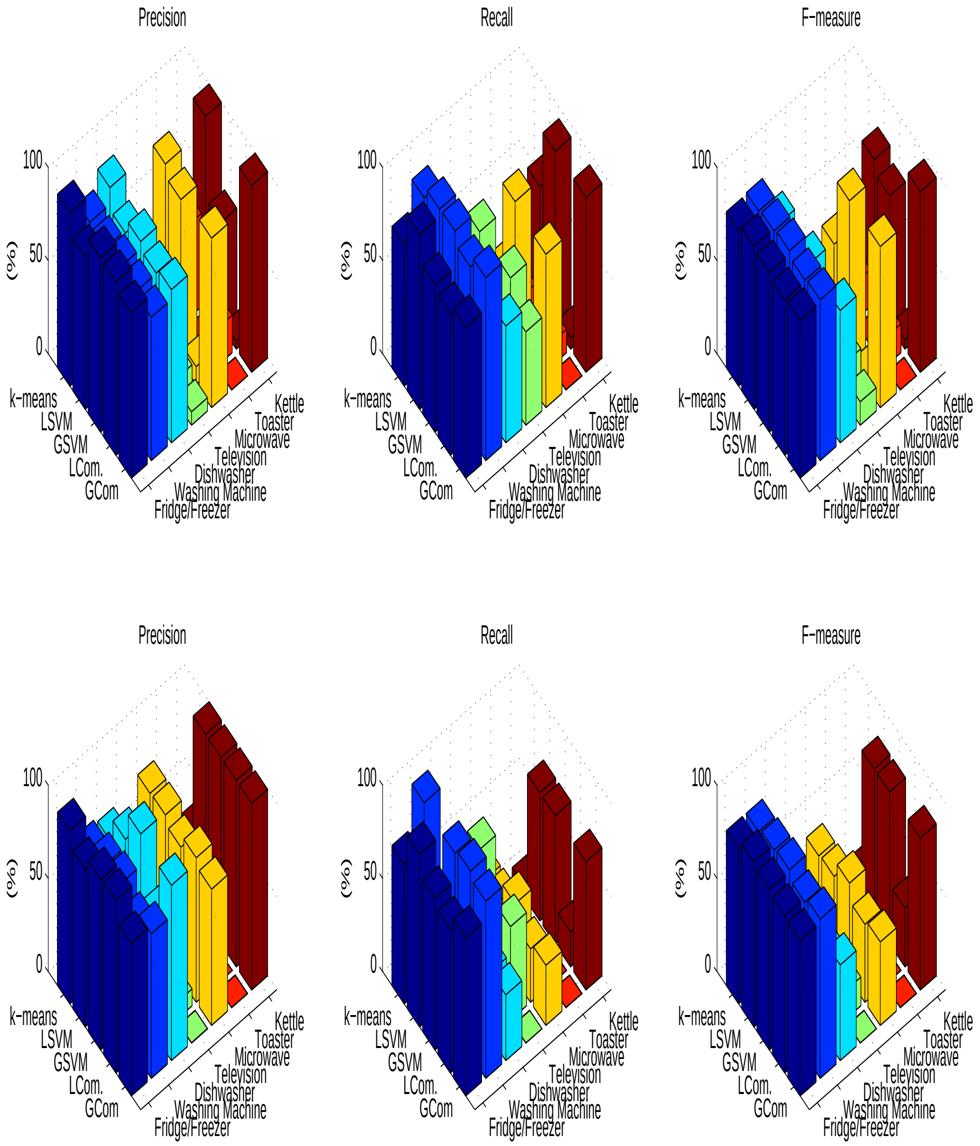


Figure 6.17: Disaggregation results for REFIT Houses 2 Supervised (Top figures) and Gaussian Unsupervised modeling (Bottom figures) methods using Trained k-means, LSVM, GSVM, Linear combined algorithm and Gaussian combined algorithm.

6.5.3 Feature Selection

Similar to the Supervised method introduced and discussed in the previous chapter, feature selection has a major role in disaggregating appliances using the Gaussian

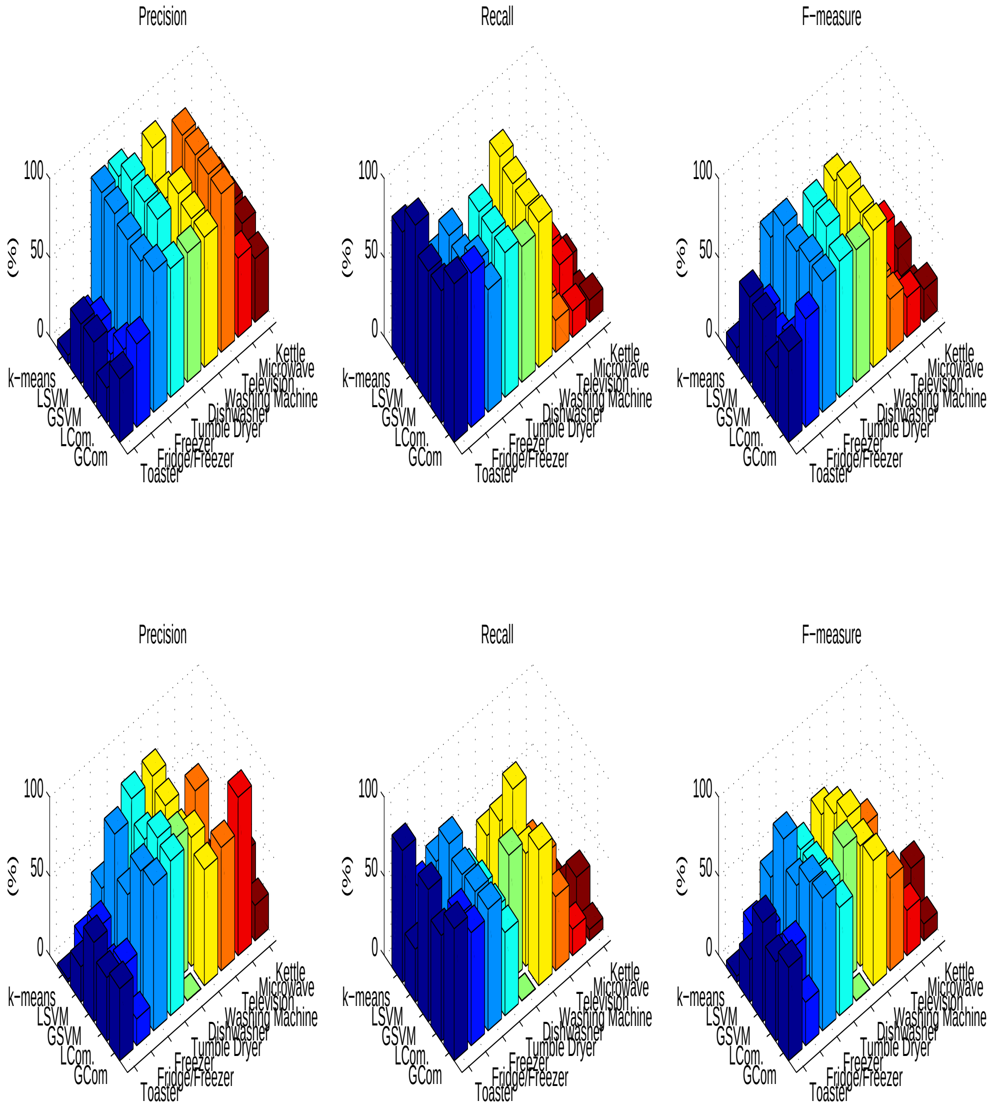


Figure 6.18: Disaggregation results for REFIT Houses 3 Supervised (Top figures) and Gaussian Unsupervised modelling (Bottom figures) methods using Trained k-means, LSVM, GSVM, Linear combined algorithm and Gaussian combined algorithm.

unsupervised method. Table 6.11 shows different feature combinations were used by Trained k-means, Linear SVM, Gaussian SVM, Linear combined and Gaussian combined algorithms to better detect washing machine appliances in GREEND dataset Houses 1 and 2 and dishwasher appliances in GREEND dataset Houses 2 and 3.

It can be seen that two-dimensional feature combinations gave better performance for different dimensional data. It is also clear that Maximum power value helps correctly disaggregate washing machine and dishwasher. Mean power value and Maximum over mean power ratio were also good in detecting washing machine and dishwasher; minimum power value was less likely used. Interestingly, the area under the curve did not give good performance by any tested algorithms.

Table 6.11: Best feature combinations for detecting washing machines and dishwashers among all possible feature combinations

Appliance	Method	*Supervised Method*	*Unsupervised Method*
H1 WM	Trained k-means	ratio & dur	max & dur
	Linear SVM	max & mean	max & mean
	Gaussian SVM	ratio & dur	max & ratio
	Linear Combined	max & mean	max & min
	Gaussian Combined	max & ratio	min & ratio
H2 WM	Trained k-means	max & min	min & ratio
	Linear SVM	max & mean	max & mean
	Gaussian SVM	max, min & dur	max & ratio
	Linear Combined	max & mean	max & ratio
	Gaussian Combined	min &mean	max & min
H1 DW	Trained k-means	max & min	max & mean
	Linear SVM	max & min	ratio & dur
	Gaussian SVM	ratio & dur	max &ratio
	Linear Combined	max & min	max & ratio
	Gaussian Combined	max & dur	max & ratio
H2 DW	Trained k-means	max & dur	max & dur
	Linear SVM	max & min	ratio & dur
	Gaussian SVM	max & dur	max & ratio
	Linear Combined	max & min	max & min
	Gaussian Combined	min & ratio	ratio & dur

Table 6.12 shows the best feature combinations among all possible feature combinations for REFIT House 2 using Supervised method and Unsupervised method. It is noticed that 2-dimensional and 3-dimensional feature combinations have performed better than 4-dimensional and 5-dimensional feature combinations, which can mean that high dimensional data mean harder job for classification algorithms. It can also be seen that *maximum power value and duration of event* are the main features that helped detecting most of appliances in House 2, such as freezer, television and microwave. It can be seen also that no toaster events were correctly classified using all tested algorithms which was predicted in Section 6.5.2.1 due to its states signatures

included across all groups which means that it can be easily misclassified into other appliances. *Minimum power value and maximum power value over mean power value* features were very useful to detect washing machine, dishwasher and kettle. However, *area* was less likely to help correctly disaggregate appliances.

Table 6.12: Best feature combinations among all possible feature combinations for REFIT House 2. NA= Not Available. Max= maximum power value. Min= minimum power value. dur= duration of an event. ratio= maximum power value over mean power value ratio.

Appliance	Method	*Supervised Method*	*Unsupervised Method*
Fridge-Freezer	Trained k-means	dur & ratio	max & min
	Linear SVM	max, dur & area	max & ratio
	Gaussian SVM	max, min & dur	min & dur
	Linear Combined	max, dur & area	max & ratio
	Gaussian Combined	area & dur	area & min
Washing Machine	Trained k-means	dur & ratio	max & min
	Linear SVM	max, min & dur	min & dur
	Gaussian SVM	max, dur & ratio	min & dur
	Linear Combined	dur & ratio	max & ratio
	Gaussian Combined	max, dur & ratio	dur & ratio
Dishwasher	Trained k-means	max, min & ratio	max & ratio
	Linear SVM	max, min & ratio	min & ratio
	Gaussian SVM	max, min & area	area & dur
	Linear Combined	max, area & ratio	max & dur
	Gaussian Combined	max, min & area	area & dur
Television	Trained k-means	NA	NA
	Linear SVM	area & dur	max, min & ratio
	Gaussian SVM	max, min & dur	NA
	Linear Combined	area & dur	max , dur & area
	Gaussian Combined	dur & ratio	NA
Microwave	Trained k-means	NA	min, dur & area
	Linear SVM	max, min & dur	area & dur
	Gaussian SVM	max & area	max & ratio
	Linear Combined	max, area & ratio	max & area
	Gaussian Combined	max & min	area & dur
Toaster	Trained k-means	NA	NA
	Linear SVM	NA	NA
	Gaussian SVM	NA	NA
	Linear Combined	NA	NA
	Gaussian Combined	NA	NA
Kettle	Trained k-means	max, min & ratio	area & ratio
	Linear SVM	min, dur & ratio	min & ratio
	Gaussian SVM	max, area & ratio	max & min

	Linear Combined	dur, area & ratio	area & ratio
	Gaussian Combined	max & ratio	area & dur

Table 6.13 shows the best feature combinations among all possible feature combinations for REFIT House 3 using Supervised method and Unsupervised method. Also here, it can be noticed that 2-dimensional and 3-dimensional feature combinations have performed better than 4-dimensional and 5-dimensional feature combinations. However, no common feature was used to detect appliances in House 3. However, *maximum power value, duration of events and maximum power value over mean power value ratio* were used to correctly disaggregate toaster events in both Supervised and Unsupervised events by all tested algorithms. *Duration of events and minimum power value* were unique values to help detect fridge-freezer events. *Duration of events and maximum power value over mean power value ratio* were used by trained k-means, Linear SVM and Gaussian SVM to detect freezer in Supervised method, while *minimum power value and maximum power value over mean power value ratio* helped in Unsupervised method. Interestingly, television and kettle area values were used by most of our tested algorithms.

Table 6.13: Best feature combinations among all possible feature combinations for REFIT House 3. NA= Not Available. Max= maximum power value. Min= minimum power value. dur= duration of an event. ratio= maximum power value over mean power value ratio.

Appliance	Method	*Supervised Method*	*Unsupervised Method*
Toaster	Trained k-means	max, min & ratio	max & dur
	Linear SVM	max, min & dur	dur & ratio
	Gaussian SVM	max, dur & ratio	max & dur
	Linear Combined	max & ratio	min & dur
	Gaussian Combined	max, dur & area	min & ratio
Fridge-Freezer	Trained k-means	dur & ratio	area & dur
	Linear SVM	max & dur	min & dur
	Gaussian SVM	area & dur	dur & ratio
	Linear Combined	dur & ratio	max & area
	Gaussian Combined	max & min	max & min
Freezer	Trained k-means	dur & ratio	max, min & ratio
	Linear SVM	dur & ratio	min & dur
	Gaussian SVM	dur & ratio	dur & ratio
	Linear Combined	max, min & dur	min & ratio
	Gaussian Combined	max, dur & area	max & min
Tumble Dryer	Trained k-means	min & dur	NA
	Linear SVM	max, dur & ratio	max & ratio

	Gaussian SVM	max & dur	max & min
	Linear Combined	max, min & ratio	max & dur
	Gaussian Combined	max & ratio	max & min
Dishwasher	Trained k-means	NA	NA
	Linear SVM	NA	NA
	Gaussian SVM	NA	NA
	Linear Combined	NA	area & dur
	Gaussian Combined	NA	NA
Washing Machine	Trained k-means	min & dur	max, dur & area
	Linear SVM	dur & ratio	dur & ratio
	Gaussian SVM	min, dur & area	max & dur
	Linear Combined	min, area & ratio	max & min
	Gaussian Combined	min, dur & area	max & ratio
Television	Trained k-means	area & dur	NA
	Linear SVM	area & dur	NA
	Gaussian SVM	area & dur	area & min
	Linear Combined	area & dur	area & dur
	Gaussian Combined	dur, area & ratio	max, min & ratio
Microwave	Trained k-means	NA	NA
	Linear SVM	NA	NA
	Gaussian SVM	max, min & dur	NA
	Linear Combined	area & ratio	NA
	Gaussian Combined	max & min	min & ratio
Kettle	Trained k-means	NA	NA
	Linear SVM	max, dur & ratio	NA
	Gaussian SVM	max, dur & area	min & ratio
	Linear Combined	min, area & ratio	max & area
	Gaussian Combined	dur, area & ratio	max & dur

6.5.4 Results Summary

As expected, it was clear that Linear and Gaussian combined algorithms provide a competitive performance compared to other tested algorithms in terms of time complexity and accuracy to correctly detect washing machine and dishwasher appliances in GREEND dataset Houses 1, 2 and 3, as well as, seven appliances in House 2 and nine appliances in House 3 in the REFIT dataset. Again, both combined algorithms provided a trade-off between speed of processing and quality of disaggregation using different sets of features that simplifies data and maximizes performance.

Similar appliances across both datasets like washing machine and dishwasher gave similar results. It can be summerized that washing machine was mixed with tumble dryer when present in same houses. Dishwasher was harder to be correctly classified across all tested houses since its low state was mixed with lower appliances signatures and high state was mixed with washing machine signature.

Our proposed Unsupervised methods showed a good replacement of regular training in case of clear appliance signature and also lower execution time in testing step. Next, we benchmark our proposed method with HMM algorithm.

6.6 Benchmark

Similar to Chapter 4, we compare the proposed approaches (Linear and Gaussian combined algorithms) with the state-of-the art HMM-based method of [43], which was designed for low-sampling (1 min) rates. For each dataset, all three tested algorithms always use the same amount of data for training. The HMM-based method [43] requires prior initialization of the model using expert knowledge (state variances, mean value for each state and state transition probabilities), which was carried out in our experiments either using the information provided by the authors of [43], or were generated during training. The combined algorithms select the best feature combination to be used for GREEND dataset washing and dishwasher and REFIT dataset Houses 2 and 3, and then perform classification.

Supervised Method uses *"Regular Training of one week data"* while Unsupervised method uses washing machine and dishwasher models, which were generated using *one month worth of data* from different houses.

6.6.1 Time Complexity

Table 6.14 shows a comparison between Linear combined algorithm, Gaussian combined algorithm and HMM in detecting washing machine and dishwasher appliances in GREEND dataset Houses 1, 2 and 3 using Supervised method and our two proposed Unsupervised methods. It can be seen that HMM has significantly higher testing execution times to classify washing machine in Houses 1 and 2 and dishwasher in Houses 2 and 3 with 1037.55, 1023.23, 954.93 and 864.82 seconds respectively compared to less than 1 second by Linear and Gaussian combined algorithms. Execution times in detecting dishwasher seem slightly lower than washing machine execution times results by HMM.

Table 6.14: Testing execution time results of washing machines and dishwasher disaggregation in GREEND Houses. H denotes House number. L C denotes Linear combined algorithm. G C denoted Gaussian combined algorithm. GM denotes General Modeling.

	Supervised Method		Unsupervised			GM method	
Appliance	L C (sec)	G C (sec)	HMM (sec)	L C (sec)	G C (sec)	L C (sec)	G C (sec)
H1 WM	0.21	0.32	1037.55	0.13	0.24	0.081	0.107
H2 WM	0.27	0.38	1023.23	0.19	0.29	0.082	0.190
H2 DW	0.31	0.42	954.93	0.21	0.35	0.090	0.097
H3 DW	0.29	0.40	864.82	0.17	0.32	0.082	0.089

Table 6.15 shows a comparison between Linear combined algorithm, Gaussian combined algorithm and HMM in detecting all known appliances in REFIT dataset Houses 2 and 3 using Supervised method and the two proposed Unsupervised methods. It can be seen that HMM has significantly higher testing execution times, due to its complexity, with 3884.26 and 5047.75 seconds in Houses 2 and 3 respectively, compared to less than 3 seconds by Linear and Gaussian combined algorithms. Execution times in detecting House 3 is slightly higher than detecting appliances in House 2 by HMM due to less number of appliances.

Table 6.15: Testing execution time results of disaggregation in REFIT Houses 2 and 3. H denotes House number. L C denotes Linear combined algorithm. G C denoted Gaussian combined algorithm. GM denotes General Modeling.

	Supervised Method		Unsupervised			GM method	
House	L C (sec)	G C (sec)	HMM (sec)	L C (sec)	G C (sec)	L C (sec)	G C (sec)
H2	1.861	2.091	3884.26	1.226	1.88	0.732	0.992
H3	2.42	3.071	5047.75	2.171	2.781	0.824	1.038

6.6.2 Accuracy

Tables 6.16 and 6.17 show Precision, Recall and F-Measure results in GREEND dataset Houses 1, 2 and 3 after disaggregation with Supervised, Unsupervised and

General modeling methods using Linear combined algorithm, Gaussian combined algorithm and HMM.

Washing machine results for Houses 1 and 2 are presented in Table 6.16 using Precision, Recall and F-Measure using different training methods. It can be seen that Linear and Gaussian combined algorithms performed well for Supervised and Unsupervised approaches with above 77% F-Measure for the supervised method. House 2 results were high by Linear and Gaussian combined algorithms with above 90% F-Measure for supervised and unsupervised methods. However, in the same table, Linear and Gaussian algorithms gave very poor results in detecting washing machine signature by General modeling method as expected. HMM, on the other hand, gave very poor performance in Precision and F-Measure of less than 5% for both tested GREEND houses and a very high Recall of 95.5% and 96.4% in Houses 1 and 2 respectively which reflects very small false negative but high false positive.

Similarly, dishwasher results for Houses 2 and 3 are presented in Table 6.17 using Precision, Recall and F-Measure for different training methods. It can be seen that Linear and Gaussian combined algorithms have a very high performance for Supervised and Unsupervised approaches of above 98% F-Measure in supervised method. House 2 results were slightly lower using supervised method with Linear and Gaussian combined algorithms with roughly 5% less, but gave slightly higher performance for roughly 3% higher F-Measure by unsupervised method. However, unlike the washing machine general modeling method, Linear and Gaussian algorithms gave quite high performance in detecting dishwasher with the General modeling method as discussed before. HMM, on the other hand, gave very poor performance in F-Measure of less than 10% for both tested GREEND houses. A Precision of 24.68% in House 2 means lower number of true positive events compared to false positive events. A high Recall of 82% in House 3 which reflects very small false negative but higher false positive.

Table 6.16: Results of washing machine disaggregation in GREEND Houses 1 and 2 for three different methods. Houses 3, 4 and 5 are used for training. L C denotes Linear combined algorithm. G C denoted Gaussian combined algorithm. GM denotes General Modeling.

		Supervised Method		Unsupervised			GM method	
H	%	L C	G C	HMM	L C	G C	L C	G C
H1	PR	100	63.82	0.5	72.54	62.5	0	0
	RE	65.21	100	95.5	80.43	100	0	0
	F_M	78.92	77.92	1.04	76.28	76.92	0	0
H2	PR	87.56	100	2.14	88.233	100	9.09	5.55
	RE	93.33	88.97	96.4	100	88.18	3.33	4.33
	F_M	90.32	94.16	4.19	93.57	93.72	4.87	4.87

Table 6.17: Results of dishwasher disaggregation in GREEND Houses 2 and 3 for three different methods. Houses 1, 4, 5 and used for training. L C denotes Linear combined algorithm. G C denoted Gaussian combined algorithm. GM denotes General Modeling.

		Supervised Method		Unsupervised			GM method	
H	%	L C	G C	HMM	L C	G C	L C	G C
H2	PR	97.01	100	24.68	88.40	88.40	92.2	100
	RE	100	94.87	3.19	93.84	93.84	100	92.91
	F_M	98.48	97.36	5.66	91.04	91.04	97.01	96.30
H3	PR	87.50	100	2.51	87.50	100	83.33	91.2
	RE	100	90	82	100	90	71.42	65.35
	F_M	93.33	94.73	4.88	93.33	94.73	76.92	76.14

Figure 6.18 shows a comparison between all three tested algorithms in REFIT House 2. Seven known appliances are present: fridge-freezer, washing machine, dishwasher, television, microwave, toaster, kettle. Fridge-freezer and washing machine have the highest number of events in the testing set, that is 671 and 612 events respectively, which made all algorithms have a significantly high Precision, Recall and F-Measure results. 80% or above for fridge-freezer for Precision, Recall and F-Measure for both Supervised and Unsupervised methods. Linear combined had slightly lower F-Measure result with roughly 75% compared to Gaussian combined algorithm. Washing machine performed well (F_M above 80 %) using Linear combined and Gaussian combined algorithms with both Supervised and Unsupervised methods. Dishwasher, was high in F-Measure using Gaussian combined in Supervised method but much lower using Linear combined in Supervised, while both algorithms gave

lower performance using Unsupervised method. General modeling method, on the other hand, was very poor for all tested algorithms disaggregating all seven appliances except for washing machine which means that its general model was unique enough compared to other appliances. HMM suffered from higher number of appliances.

Table 6.18: Results after disaggregation in REFIT Houses 2 and 3 for three different methods. G M Method denotes General Modelling Method. L C denotes Linear combined algorithm. G C denoted Gaussian combined algorithm. Fr-Frz=Fridge-Freezer. WM=Washing Machine. DW=Dishwasher. TV=Television. MW=Microwave. Tstr=Toaster. K=Kettle. GM denotes General Modeling.

		Supervised Method		Unsupervised			GM Method	
Appliance	%	L C	G C	HMM	L C	G C	L C	G C
Fr-Frz	PR	92.85	88.72	0	92.33	81.84	52.95	0
	RE	75.55	80.92	0	77.19	84.64	100	0
	F_M	83.31	84.64	0	84.09	83.22	96.24	0
WM	PR	78.33	76.77	71.53	70.24	77.48	73.28	43.68
	RE	91.41	98.05	15.88	98.70	94.81	15.55	96.90
	F_M	84.36	86.12	25.99	82.07	85.27	25.66	60.22
DW	PR	79.72	82.08	0	64.58	93.93	26.37	4.83
	RE	42.04	62.5	0	35.22	35.22	27.27	3.4
	F_M	54.81	70.96	0	45.58	51.23	26.81	4
TV	PR	12.5	7.14	0	9.09	0	0	0
	RE	66.66	50	0	50	0	0	0
	F_M	21.05	12.5	0	15.38	0	0	0
MW	PR	9.52	90.78	100	77.41	72.97	0	6.6
	RE	2.38	82.14	5.95	28.57	32.14	0	2.3
	F_M	3.8	86.25	11.23	41.73	44.62	0	3.5
Tstr	PR	20	0	0	0	0	0	0
	RE	12.5	0	0	0	0	0	0
	F_M	15.38	0	0	0	0	0	0
K	PR	4.16	100	20	100	100	0	100
	RE	6.25	93.75	6.25	18.75	68.75	0	6.25
	F_M	5	96.77	9.52	31.57	81.48	0	11.76

Figure 6.19 shows a comparison between all three tested algorithms in REFIT House 3. Nine known appliances are present: Toaster, Fridge-Freezer, Freezer, Tumble Dryer, Dishwasher, Washing Machine, Television, Microwave, Kettle.

Freezer and washing machine have the highest number of events in testing set, namely, 326 and 364 events respectively, which was reflected on all tested algorithms, except HMM, by having a significantly high Precision, Recall and F-Measure results. 80% or above for both freezer and washing machine for F-Measure in both Supervised and Unsupervised methods. Dishwasher was not detected at all, using all possible feature combinations, by Linear combined algorithm using Supervised method but much higher F-Measure of about 70% was obtained using Unsupervised method, although, Gaussian combined algorithm gave good performance with 81.66%, 85.58% and 83.58% of Precision, Recall and F-Measure results, respectively, using Supervised method though the appliance was not detected using Unsupervised method. General modeling method, on the other hand, was very poor in all tested algorithms disaggregating all nine appliances except for tumble dryer which means that its general model was distinguishable compared to other appliances. HMM, as in House 2, gave very poor performance compared Linear and Gaussian algorithms.

Table 6.19: Results after disaggregation in REFIT Houses 2 and 3 for three different methods. G M Method denotes General Modelling Method. L C denotes Linear combined algorithm. G C denoted Gaussian combined algorithm. Fr-Frz=Fridge-Freezer. WM=Washing Machine. DW=Dishwasher. TV=Television. MW=Microwave. Tstr=Toaster. K=Kettle. TD=Tumble Dryer. Fzr=Freezer. GM denotes General Modeling.

		Supervised Method		Unsupervised			GM Method	
Appliance	%	L C	G C	HMM	L C	G C	L C	G C
Tstr	PR	21.73	40	0	40	45.45	2.1	3.9
	RE	83.33	100	0	66.66	83.33	50	66.66
	F_M	34.48	57.14	0	50	58.82	4.1	7.4
Fr-Frz	PR	36.61	52.61	12.22	38.23	16.94	0	8.86
	RE	46.42	97.33	28.57	69.64	71.42	0	100
	F_M	40.94	68.30	17.11	49.36	27.39	0	16.27
Fzr	PR	82.97	88.73	18.42	88.48	91.91	60	0
	RE	82.20	77.30	2.14	75.46	76.68	97.54	0
	F_M	82.58	82.62	3.8	81.45	83.61	74.82	0
TD	PR	100	81.11	75	100	97.67	19.51	43.75
	RE	91.25	91.25	7.5	53.84	52.5	30	35
	F_M	95.42	85.88	13.63	70	68.29	23.64	38.88
DW	PR	0	81.66	0	89.74	0	7.4	12.5
	RE	0	85.58	0	79.54	0	27.27	18.18
	F_M	0	83.58	0	84.33	0	11.65	14.81
WM	PR	81.5	82.01	84.37	81.11	73.23	0	0
	RE	89.56	91.48	14.83	71.15	85.71	0	0
	F_M	85.34	86.49	25.23	75.84	78.98	0	0

TV	PR	100	100	4	47.36	77.77	0	0
	RE	20	20	13.33	60	46.66	0	0
	F_M	33.33	33.33	6.15	52.94	58.33	0	0
MW	PR	7.6	50	0	0	100	0	0
	RE	33.33	16.66	0	0	16.66	0	0
	F_M	12.5	25	0	0	28.57	0	0
K	PR	50	40	0	42.10	22.22	0	0
	RE	3.4	13.79	0	27.58	6.89	0	0
	F_M	6.4	20.51	0	33.33	10.52	0	0

6.6.3 Benchmark Summary

Experimental results using GREEND dataset Houses 1, 2 and 3, and REFIT dataset Houses 2 and 3 demonstrate the competitiveness of the proposed solutions with respect to a state-of-the-art HMM-based approach. Indeed, the proposed approaches show significantly better performance to that of HMM, with up to 1000 times lower execution time for testing GREEND dataset, and up to 5000 times lower execution time for testing REFIT dataset. Simulations showed that HMM based algorithm could not perform well for high sampling rate of GREEND dataset of roughly 6 seconds and roughly 9 appliances in each house, and higher number of known appliances with noisy for REFIT dataset, which made it hard to model each appliance due to significantly high overlapping between appliances in aggregated data, unlike other disaggregating algorithms tested here.

6.7 Chapter Summary and Conclusion

Designing accurate NALM algorithms for different sampling rate data is not an easy task. In this chapter we tested our proposed Linear and Gaussian low-complexity algorithms based on combining Trained k-means and Linear and Gaussian SVMs on supervised and unsupervised approaches. We focused on washing machine and dishwasher in GREEND dataset as they are assumed to be high consumers of electricity and we used a range of GREEND dataset houses with a considerably high sampling rate. REFIT dataset was a challenge also for all tested algorithms, as it was nosier that GREEND, and higher number of appliances were present.

We have created and validated a database of signatures using Gaussian fitted models in Chapter 5; we have proposed and tested two unsupervised methods using signatures derived from that database. One method is called Training-less General modeling which uses signatures straight from the signature database which gave a poor results in detecting washing machine and a good performance in detecting dishwasher due to the unique signature of dishwasher in GREEND dataset Houses 2 and 3. However, results with this approach were poor in the REFIT dataset.

The second unsupervised method called Gaussian unsupervised method which uses a set of different feature combinations derived from Gaussian samples of the signatures to simplifies and maximize performance which shows a lower simulation time and competitive performance to that using training from houses-specific data. However, in some cases, in REFIT dataset results, the unsupervised method outperformed the supervised results, due to better quality of training using signatures instead of features per event, which could provide an attractive solution to noisy datasets.

The two combined algorithms (Linear and Gaussian) showed an accurate steady performance using supervised and unsupervised methods in all tested appliances and a much better performance compared to HMM algorithm in terms of complexity and accuracy, which struggled with higher number of appliances and a higher sampling rate of GREEND and REFIT datasets compared to REDD dataset results introduced in Chapter 4.

Chapter 7

Conclusions and Further Discussion

7.1 Introduction

In this chapter, the main findings with regards to the research objectives are summarized and general conclusions based on the findings of the studies presented in this thesis are described. Furthermore, the limitations of the proposed methods are considered and suggestions for further research are presented.

7.2 Main Findings

This thesis was driven by the concept of real-time response and low-complexity approaches. We proposed different NALM solutions; by combining Trained k-means and SVM, we managed to reduce the computational cost of SVM without losing its high performance. We tested our proposed Linear and Gaussian combined algorithms in comparison with k-means, Linear and Gaussian SVMs and also benchmarked our combined algorithms with an HMM-based NALM approach. We then introduced two *unsupervised* methods based on a signature database to reduce the need for training classifiers.

For the proposed supervised method, a proper training dataset is presented to help train classifiers to disaggregate testing data. Both training and testing data have to be from the same house in order to have good accuracy output. The proposed unsupervised methods, however, allow us to approach solutions with little or no information about present appliances. Many researchers prefer to use unsupervised methods but the lack of successful *(utterly blind)* methods make supervised methods still popular, adding to that, supervised methods usually provide much higher and more liability performance.

7.2.1 Supervised Method

In Chapter 4, we tested our proposed Linear and Gaussian low-complexity combined algorithms based on combining Trained k-means and Linear and Gaussian SVM. By combining SVM with k-means, we can benefit from the low complexity of k-means and the high performance of SVM. The appliances from a range of REDD dataset houses with roughly 1 minute sampling rate and 5 appliances each were used to evaluate performance and robustness.

We also tested robustness of the two proposed algorithms by inserting random errors in training datasets and reduced training from one week to roughly two days. The two combined algorithms using house-specific training data are accurate even when the training period as short as two days only for training and training errors are present with up to 20% error rate and also showed competitive performance to state-of-the-art approaches of Linear and Gaussian SVM and Hidden Markov Models. A set of different feature combinations were used to simplify and maximize performance, which were used to create 2-dimensional, 3-dimensional, 4-dimensional and 5-dimensional set of features from 1-dimensional power readings. It was found that 2-dimensional and 3-dimensional training data gave better results than higher dimensional training data to classify most of the appliances correctly due to classifiers having harder task with higher dimensional data.

7.2.2 Signature Database Creation and Clustering

In Chapter 5, we created a database of signatures that is based on Gaussian Distribution Estimation. The database includes appliance load models from GREEND and REFIT datasets. All houses from GREEND dataset and REFIT Houses 2 and 3 were clustered, studying similarities and dissimilarities between all different types of appliances based on their Gaussian signatures, using two simple, yet effective, techniques: First, we used mean-shift clustering algorithm to section all appliance and then sub-section any relatively big groups.

In the second method, we created a hierarchical tree using the well-known Genetic k-means algorithm, then cut the tree into a proper level to form different groups and sub-groups. Both methods, mean-shift and tree clustering, gave similar prediction results to the same appliances and good prediction to which appliances would be hard to classify. Tree clustering gave detailed connection between appliances from root to leaf nodes with one or two appliances in each leaf.

7.2.3 Unsupervised Methods

In Chapter 6, we tested our proposed Linear and Gaussian low-complexity algorithms based on combining Trained k-means and Linear and Gaussian SVMs on supervised and unsupervised approaches. We focused on washing machine and dishwasher from GREEND dataset as they are assumed to be high consumers of electricity and exist in at least four GREEND houses. We also used REFIT dataset as it offers higher number of appliances that exist in most houses such as tumble dryer, toaster and freezer.

We also proposed and tested two unsupervised methods using signatures derived from that database. One method called Training-less General modeling which uses signatures straight from the signature database which gave poor results in detecting washing machine and good performance in detecting dishwasher due to the unique signature of dishwasher in GREEND dataset. All tested appliances from REFIT Houses 2 and 3 gave poor performance with General modeling method.

The second unsupervised method called Gaussian unsupervised method which uses a set of different feature combinations derived from Gaussian samples of the signatures to simplify and maximize performance of our combined algorithms which showed a lower simulation time and competitive performance to that using training from houses-specific data. The two combined algorithms gave accurate steady performance using supervised and unsupervised methods for all monitored appliances and a much better performance compared to HMM algorithm that struggled with higher number of appliances and sampling rate of GREEND and REFIT datasets.

7.3 Further Discussion

From the conclusion points, it can be noticed that not all methods work perfectly with different types of appliances. That means, some appliances can be disaggregated better using unsupervised approaches than supervised method, or works better with Gaussian combined algorithm than Linear combined algorithm. The reason for that depends on how the appliance operates for short or long periods of time, as well as, whether the data is clean enough. However, here are few points worth mentioning:

- Refrigerators were usually easy to disaggregate by most used methods due to the high number of event windows present in training and testing datasets.
- Appliances such as washing machines and tumble dryer are usually hard to disaggregate when another high consuming appliance is present in the same house, due to their level of consumption and long cycles being similar.
- Different types of Televisions have significantly different operation cycles. For example, LCD televisions do not operate similar to Plasma televisions.
- Toasters and kettles were hard to separate when present in the same house due to their short cycles and high energy consumption values being similar.

These findings are summarized from this thesis outcomes using kmeans, Linear SVM, Gaussian SVM, Linear combined algorithm and Gaussian combined algorithm and benchmarked with HMM-based algorithm. There was a clear trade-off between accuracy and simplicity in all results, better results were obtained by Gaussian algorithms (Gaussian SVM and Gaussian combined algorithm) but slower performance due to higher complexity compared to linear algorithms (Linear SVM and Linear combined algorithm.

7.4 Advantages

This work mainly focuses into two major issues in developing NALM solutions; practicality and simplicity. The key advantage of this work was providing low-complex solutions to solve the NALM disaggregation problem. By combining Trained kmeans and support vector machines, we managed to enhance the performance of linear and Gaussian support vector machines, which then provided lower computational cost. The resulting combined algorithms were low-complex, reliable and out-performed kmeans and SVM individually.

We managed to use only one dimensional data which is active power and convert it into multi dimensional data by using different features that have provided a major impact on this research. By doing so, there was no need to use any extra source of information, such as circuits to measure current and voltage, to disaggregate appliances into separate readings.

We have also provided a dataset of appliance-load signatures for other researchers to use in energy disaggregation task. This dataset provides a number of load characteristics extracted from Gaussian mixture model of appliances from GREEND and REFIT datasets. We managed to advantage from using these signatures to provide Unsupervised solutions without the need of excessive training. The unsupervised approaches out-performed the supervised approach in few cases.

7.5 Limitations

Although this research have reached its goals, we are aware of its limitations and shortcomings, which are listed below:

- The proposed algorithms combine Trained k-means and SVM and use a preset threshold which separates training and testing datasets between Trained k-means and SVM. As explained in Section 3.5, this threshold is set manually which can be impractical. An adaptive threshold could solve this limitation.
- The proposed approaches, supervised and unsupervised, still need labeled data. By that, we mean a label-vector is needed to train classifiers in order to clearly separate appliances for classification.
- There is a need to train using *'all'* possible feature selections, as features play a key role in the disaggregation task, which can be unpractical.
- The current approaches can not separate many overlapped events i.e., when appliances operate at the same time, and such events are discarded, though,

partially overlapped-events can be classified by separating overlapped parts and total overlapped parts.

- Both approaches (supervised and unsupervised) require to re-train when a new appliance is introduced.

7.6 Potential Applications

Energy load disaggregation using NALM, based purely on analytical tools, has been gaining popularity, especially with ongoing smart meter roll-outs worldwide. Real-time feedback on energy consumption is a clear opportunity of energy disaggregation task. In technology sector, new products could be added into the existing smart meters by many providing companies. By devising many uses for NALM output, we could possibly deliver enhanced services to consumers such as detailed billing. Detailed billing with energy usage breakdown into separate appliances, could persuade consumers to adopt better energy consumption lifestyles and hint to which appliances could be replaced into a better energy saving models.

On the other hand, smart grids can benefit from enhanced smart meters. Since smart grids use demand-side energy management to self-regulate their energy footprint, a reduction of the overall energy consumption and peak power usage is key advantage of the long-term solution of NALM. Demand-side management carried out by smart grids requires smart buildings, where monitoring energy consumption of electrical loads is a constantly operating function to remotely control how much energy every load consumes. These tasks could be performed by modified smart meters to reduce energy usage and carbon emission which contributes to climate change.

7.7 Future work

The future work will contain:

- Include and test more feature combinations to further improve the performance of the combined algorithms such as order of maximum and minimum peaks.

- Implementation of the Linear and Gaussian combined algorithms on other types of appliances for both supervised and unsupervised methods.
- Further development of the database of signature by generating appliance-load signatures of different datasets.
- Further investigation of clustering methods using signatures generated from other datasets including further REFIT dataset houses.

Appendix A

Supervised Methods Experimental Results

Table A.1: Comparison between selected feature combinations using Trained k-means, Linear SVM, Gaussian SVM, Linear combined algorithm and Gaussian combined algorithm for REDD dataset Houses 1, 2 and 6.

House number	*Feature combination*	*Trained k-means* (%)	*Linear SVM* (%)	*Gaussian SVM* (%)
House 1	*Max. & Min.*	70.33	67.63	71.83
	Area. & Dur.	48.72	45.09	47.15
	Max.,Dur. & Area	64.83	69.77	66.77
	Max.,Min. & Max/Mean	71.14	74.52	71.67
	Min.,Dur.,Area & Max/Mean	63.75	66.77	62.02
	All features	66.57	67.4	63.44
House 2	*Max. & Min.*	82.53	85.55	81.78
	Area. & Dur.	85.42	52.76	53.89
	Max.,Dur. & Area	85.55	85.17	76.38
	Max.,Min. & Max/Mean	75.37	83.41	76.75
	Min.,Dur.,Area & Max/Mean	85.42	85.55	56.78
	All features	85.05	83.66	55.9
House 6	*Max. & Min.*	90.83	86.35	96.58
	Area. & Dur.	86.56	62.68	68.01
	Max.,Dur. & Area	93.6	63.96	74.8
	Max.,Min. & Max/Mean	91.04	72.49	71.21
	Min.,Dur.,Area & Max/Mean	93.81	66.31	75.05
	All features	94.45	66.52	74.84

A.1 Performance

A.1.1 House 1

Table A.2: Comparison between Trained k-means, Linear SVM, Gaussian SVM, Linear combined algorithm, Gaussian combined algorithm and HMM algorithm using F_M for REDD dataset House 1.

Method	%	*Refrigerator*	*Microwave*	*Toaster*	*Dishwasher*	*W dryer*	*Total*
Kmeans	Pr	70.4	85.45	52.63	100	74.48	72.75
	Recall	95.73	73.43	86.95	4.65	88.52	74.24
	Fm	81.13	78.99	65.57	8.88	80.89	73.49
L SVM	Pr	82.91	52.17	15.15	83.87	64.58	74.26
	Recall	97.31	84.21	13.88	29.21	75.6	80.37
	Fm	89.53	64.42	14.49	43.33	69.66	77.2
G SVM	Pr	81.31	41.42	19.14	43.47	92	71.62
	Recall	96.82	50.87	25	11.23	56.09	73.89
	Fm	88.39	45.66	21.68	17.85	69.69	72.74
L Combined	Pr	83.19	64.81	16.66	50	92.85	78.66
	Recall	96.82	61.40	2.77	28.08	63.41	76.42
	Fm	89.49	63.06	4.76	35.97	75.36	77.52
G Combined	Pr	90.18	68	100	68.96	65.30	80.15
	Recall	96.57	89.47	30.55	44.94	78.04	83.70
	Fm	90.18	77.27	46.80	54.42	71.11	81.88
HMM	Pr	90	79.31	0	44.63	0	77.16
	Recall	77.16	60.53	0	86.17	0	76.97
	Fm	83.12	68.66	0	58.80	0	77..06

A.1.2 House 2

Table A.3: Comparison between Trained k-means, Linear SVM, Gaussian SVM, Linear combined algorithm, Gaussian combined algorithm and HMM algorithm using F_M for REDD dataset House 2.

Method	%	*Refrigerator*	*Stove*	*Microwave*	*Toaster*	*Dishwasher*	*Total*
	Pr	92.28	0	0	45.31	0	84.79
Kmeans	Recall	95	0	0	48.33	0	84.79
	Fm	93.62	0	0	46.77	0	84.79
	Pr	93.41	4.16	0	54.9	33.33	85.18
L SVM	Recall	91.76	33.33	0	93.33	21.42	85.92
	Fm	92.58	7.40	0	69.13	26.08	85.55
	Pr	93.31	2.24	0.81	63.15	50	70.86
G SVM	Recall	93.97	66.66	2.56	60	21.42	85.55
	Fm	93.55	4.34	1.23	61.53	30	77.51
	Pr	92.87	66.66	15.38	69.62	22.22	73.30
L Combined	Recall	95.88	33.33	82.05	91.66	42.85	93.49
	Fm	94.35	44.44	25.91	79.13	29.26	82.17
	Pr	91.07	0	27.63	67.69	50	83.54
G Combined	Recall	97.50	0	53.84	73.33	21.42	91.83
	Fm	94.17	0	36.52	70.40	30	87.49
	Pr	87.45	38.10	35.71	50	33.33	84.85
HMM	Recall	87.93	66.67	58.14	92.45	7.56	80.05
	Fm	87.69	48.48	44.25	64.90	12.32	82.38

A.1.3 House 6

Table A.4: Comparison between Trained k-means, Linear SVM, Gaussian SVM, Linear combined algorithm, Gaussian combined algorithm and HMM algorithm using F_M for REDD dataset House 6.

Method	%	*Refrigerator*	*Stove*	*Microwave*	*Toaster*	*Air Conditioner*	*Total*
Kmeans	Pr	96.66	0	0	0	100	97.01
	Recall	100	0	0	0	85.96	97.01
	Fm	98.30	0	0	0	92.45	97.01
L SVM	Pr	93.83	9.67	3.33	0	24.24	72.85
	Recall	97.53	100	100	0	42.10	90.40
	Fm	95.65	17.64	6.45	0	30.76	80.68
G SVM	Pr	99	100	2.38	0	81.81	88.86
	Recall	98.27	33.33	100	0	94.73	97.01
	Fm	98.64	50	4.65	0	87.80	92.76
L Combined	Pr	98.28	20	100	0	88.88	95.09
	Recall	99.01	100	74.07	0	86.95	96.07
	Fm	98.65	33.33	85.10	0	87.91	95.58
G Combined	Pr	98.06	33.33	85.71	0	92.72	96.39
	Recall	99.75	33.33	88.88	0	91.07	97.36
	Fm	98.90	33.33	87.27	0	91.89	96.87
HMM	Pr	69.20	0	100	0	0.43	58.54
	Recall	96.13	0	100	0	100	96.32
	Fm	80.47	0	100	0	0.85	72.82

A.2 Robustness

Table A.5: Comparison between the Trained k-means, Linear SVM, Gaussian SVM, Linear combined algorithm and Gaussian combined algorithm using Testing execution time for REDD data Houses after reduction of the training dataset.

Method		House 1	House 2	House 6
		(sec)	(sec)	(sec)
Trained k-means	6000	0.153	0.235	0.121
	5000	0.153	0.201	0.136
	4000	0.183	0.262	0.11
	3000	0.138	0.266	0.12
	2000	0.145	0.255	0.186
Linear SVM	6000	0.581	0.698	0.566
	5000	0.659	0.722	0.464
	4000	0.624	0.731	0.529
	3000	0.688	0.682	0.391
	2000	0.57	0.718	0.361
Gaussian SVM	6000	1.14	1.461	0.876
	5000	1.282	1.028	0.938
	4000	1.097	1.189	0.596
	3000	1.118	0.867	0.548
	2000	0.677	1.069	0.493
Linear Combined	6000	0.328	0.476	0.364
	5000	0.356	0.449	0.366
	4000	0.328	0.366	0.307
	3000	0.333	0.401	0.283
	2000	0.316	0.317	0.208
Gaussian Combined	6000	0.444	0.543	0.465
	5000	0.408	0.569	0.443
	4000	0.401	0.446	0.306
	3000	0.302	0.495	0.337
	2000	0.283	0.42	0.366

A.2.1 House 1

Table A.6:** F_M **results of Trained k-means, Linear SVM, Gaussian SVM, Linear combined algorithm and Gaussian combined algorithm after reducing the training dataset for REDD dataset House 1. T denotes Training.

Training size		Fridge	Microwave	Toaster	Dishwasher	Washer dryer	Total
T k-means	6000	85.04	63.69	0	0	0	71.12
	5000	85.95	64.1	0	8.7	0	71.25
	4000	88.7	64.4	0	5.6	0	72.91
	3000	90	72.2	0	0	0	75.3
	2000	89.3	55	0	0	0	71.93
L SVM	6000	89.2	64.1	0	4.25	0	73.7
	5000	89.4	65.8	0	0	4.7	74.2
	4000	89.5	64	0	35	4.3	73.78
	3000	90.4	32.4	0	33.6	26	71.86
	2000	88.9	65.8	0	0	5	73.83
G SVM	6000	89.6	67.85	0	21.15	65.3	77.26
	5000	89.7	64.4	0	23.7	68.8	77.6
	4000	86.7	64.8	0	36.75	71.1	73.5
	3000	88.61	57.14	0	46.23	64.07	75.26
	2000	80.27	62.5	0	0	0	68.3
L combined	6000	89.4	62.9	0	31.08	20.3	74.4
	5000	89.3	42.6	0	28.5	13.6	71.7
	4000	89.5	71.5	0	40.4	0	73.8
	3000	89	32.2	0	35.2	28	72.81
	2000	88.9	62.9	0	0	0	73.26
G combined	6000	86.2	57.8	0	13.5	63.3	73.6
	5000	89	37.3	0	0	52.8	71.1
	4000	85.3	71.1	0	23.6	52.8	73.63
	3000	89.7	57.8	0	2.19	51.8	73.17
	2000	81	56.6	0	0	0	67.9

Table A.7:** F_M **results of Trained k-means, Linear SVM, Gaussian SVM, Linear combined algorithm and Gaussian combined algorithm for different Error rates for REDD dataset House 1. T denotes Trained.

Error rate		Fridge	Microwave	Toaster	Dishwasher	Washer dryer	Total
T k-means	5 %	89	40.2	0	13.5	39.2	72.66
	10 %	88.4	31.2	0	0	39.8	70.82
	15 %	87.6	15.7	0	0	39.2	71.28
	20 %	86.3	16.6	0	0	37.1	70.73
L SVM	5 %	89.5	64.4	0	41.66	65.78	78.1
	10 %	89.5	65.3	0	41.3	46.8	77.3
	15 %	89.5	65.3	0	40.6	26	76.72
	20 %	89.5	62.5	0	42	12.24	74.36
G SVM	5 %	87.6	44.6	36.5	24.3	69.69	70.33
	10 %	87.77	42.4	19.73	25.74	66.6	66.57
	15 %	88.27	48.73	36.17	26.2	32	69.45
	20 %	88.43	45.66	20	26.4	32	68.8
L combined	5 %	89.3	50.9	4.6	14.8	80	75.12
	10 %	89.2	66.2	0	13.3	81	75.6
	15 %	89	58.2	3	40.8	81.1	75.48
	20 %	88.9	58.5	3.3	37.9	81.1	76.83
G combined	5 %	89.7	56.3	0	0	34.9	70.1
	10 %	88.66	57.14	0	27.8	29.2	68.88
	15 %	90.2	72.8	0	63.7	25.7	73.1
	20%	88.5	54.4	0	0	32.9	69.12

A.2.2 House 2

Table A.8: F_M ***results of Trained k-means, Linear SVM, Gaussian SVM, Linear combined algorithm and Gaussian combined algorithm after reducing the training dataset for REDD dataset House 2. T denotes Trained.***

Training size		Fridge	stove	Microwave	Toaster	Dishwasher	Total
T k-means	6000	90.9	0	0	20	0	82.78
	5000	93.5	0	0	45	0	84.67
	4000	91.8	0	0	2.4	0	82
	3000	92.9	0	0	76.3	0	86.18
	2000	94.8	0	0	75.6	0	89.69
L SVM	6000	92.52	4.6	0	71.6	26	84.41
	5000	92.5	4.5	0	67.8	40	84.3
	4000	92.5	0	0	75.16	44.4	87.1
	3000	93.1	0	0	75.1	0	87.63
	2000	92.2	0	0	75.1	0	85.93
G SVM	6000	94.2	2.8	1.2	62.6	26.1	78.86
	5000	92.2	0	1.3	59.6	31.57	80.66
	4000	91.4	0	1.1	60.9	31.57	78.75
	3000	92.7	0	0	78.1	0	87.1
	2000	92.1	0	0	52.8	0	85.75
L combined	6000	91.84	0	0	8.9	41.6	84.85
	5000	91.88	0	0	17	0	84.4
	4000	91.7	0	0	6.4	40	84.7
	3000	90.4	0	0	60.15	0	72.68
	2000	90.22	0	0	60	0	84.7
G combined	6000	93.5	0	40	46.5	22.2	84.68
	5000	93.4	0	0	46.4	16	85.68
	4000	92.3	0	0	9.4	23	84.23
	3000	90.4	0	0	0	0	82.95
	2000	90.3	0	0	5.2	0	81.55

Table A.9: F_M ***results of Trained k-means, Linear SVM, Gaussian SVM, Linear combined algorithm and Gaussian combined algorithm for different Error rates for REDD dataset House 2. T denotes Trained.***

Error rate		Fridge	stove	Microwave	Toaster	Dishwasher	Total
T k-means	5 %	91.9	0	0	57.4	0	84.79
	10 %	93.4	0	0	46.6	0	84.79
	15 %	91.8	0	0	2.4	0	82.03
	20 %	91.9	0	0	2.4	0	82.03
L SVM	5 %	92.2	3	0	43.6	44.4	81.98
	10 %	92.2	1.3	0	0	45.5	77.35
	15 %	92	3.9	0	45.3	0	82.66
	20 %	91.7	0	0	0	0	85.53
G SVM	5 %	92.46	2.2	0	66.1	30	80.44
	10 %	92.6	0	0	47.9	30	72.47
	15 %	92.2	0	0	44.6	31.25	67.2
	20 %	91.27	0	0	0	14.6	66.8
L combined	5 %	91.8	0	0	11.4	17.39	84.6
	10 %	91.48	0	46.9	12.5	13.7	86.14
	15 %	93.3	0	0	11.9	42.8	85.58
	20 %	91.9	0	0	8.9	45.3	84.58
G combined	5 %	92.1	0	50.6	35.1	21.8	86.43
	10 %	93.6	0	48.7	33.3	20.4	87.29
	15 %	91.7	0	51.7	9.3	19.66	87.84
	20%	93.4	0	3.3	46.2	25.8	84.47

A.2.3 House 6

Table A.10: F_M results of Trained k-means, Linear SVM, Gaussian SVM, Linear combined algorithm and Gaussian combined algorithm after reducing the training dataset for REDD dataset House 6. T denotes for Trained.

Training size		Fridge	stove	Microwave	Toaster	Air conditioner	Total
T k-means	6000	98.25	0	0	0	86.36	95.9
	5000	98.65	0	0	0	87.17	96.5
	4000	98.25	0	0	0	18.18	85.2
	3000	98.25	0	0	0	86.36	95.94
	2000	92.8	0	0	0	0	86.5
L SVM	6000	96	17.14	6.4	0	29.67	80.7
	5000	96.2	16.6	4.8	0	43.27	81.5
	4000	98.25	10.34	0	0	73.78	86.3
	3000	97.75	10.52	0	0	27.16	83,8
	2000	96.7	10.1	0	0	13.6	83.15
G SVM	6000	98.6	50	4.6	0	87.8	92.76
	5000	98.5	50	4.6	0	87.1	92.55
	4000	97.75	44.4	0	0	71	93.81
	3000	97.3	57.14	0	0	77.68	94.2
	2000	92.38	66.6	0	0	24.7	85.5
L combined	6000	98.3	6.6	100	0	55.3	90.83
	5000	98.5	0	0	0	42.1	88
	4000	98.6	33.3	0	0	65.1	94.3
	3000	98.6	66.6	0	0	87.7	96.8
	2000	98.5	57.1	0	0	88.8	96.58
G combined	6000	84.5	18	0	0	47.9	75.13
	5000	98.6	14.28	100	0	92.45	96.5
	4000	98.14	15.7	0	0	82.14	92.4
	3000	92.79	66.6	0	0	5.7	85.92
	2000	91.7	60	0	0	0	84.6

Table A.11: F_M ***results of Trained k-means, Linear SVM, Gaussian SVM, Linear combined algorithm and Gaussian combined algorithm for different Error rates for REDD dataset House 6. T denotes Trained.***

Error rate		Fridge	stove	Microwave	Toaster	Air conditioner	Total
T k-means	5 %	98.3	0	0	0	91.6	96.5
	10 %	98.5	0	0	0	88.6	95.94
	15 %	98.6	0	0	0	89.3	96.58
	20 %	98.3	0	0	0	87.5	96.37
L SVM	5 %	93.6	5.7	0	0	36.6	85.6
	10 %	78.49	3.7	0	0	0	60.55
	15 %	76.2	3.6	0	0	0	57.7
	20 %	87.2	12.7	0	0	0	76
G SVM	5 %	98.5	0	0	0	86.9	88.9
	10 %	98.6	0	0	0	89.2	89.8
	15 %	98.5	0	4.1	0	89.2	91.24
	20 %	98.38	0	4	0	52.5	87.8
L combined	5 %	97.7	6.4	28.5	0	45.3	89.8
	10 %	96.3	5	100	0	88.8	91.87
	15 %	96.6	0	88.8	0	72.2	92.2
	20 %	96.2	0	4	0	22.8	83.29
G combined	5 %	93	44.4	66.6	0	65.3	96.53
	10 %	98.7	44.4	66.6	0	92.15	97.1
	15 %	98.9	0	66.6	0	89.2	96.66
	20%	98.3	0	66.6	0	88.3	87.4

Appendix B

Unsupervised Methods Experimental Results

B.1 Database Creation

B.1.1 GREEND Dataset

Table B.1: RMSE, mean, variance, 1st order correlation coefficient and 2nd order correlation coefficient for different Fridges for different GREEND dataset houses.

Appliance	Mean value	Variance	RMSE	1st Cor.	2nd Cor.
H0 Fridge with Freezer	73.74 236.74	9.89 6.4	0.0065	0.3669	0.3249
H1 Fridge	35.2	46.48	1.56 E-04	0.0086	-0.2045
H3 Fridge with freezer	109.57	6.69	7.18 E-05	0.5009	0.4663
H4 Fridge with freezer	140.27	35.44	1.04 E-02	0.2934	0.2016
H5 Fridge with freezer	64.82 202.72	3.14 66.24	4.3 E-05	0.0952	0.066
H7 Fridge with freezer	88.52 426.68 641.16	37.04 136.86 29.07	1.65 E-04	0.2272	0.1632
H7 Freezer	101 56.24	21.97 1.07	5.1 E-03	0.0448	-0.0273

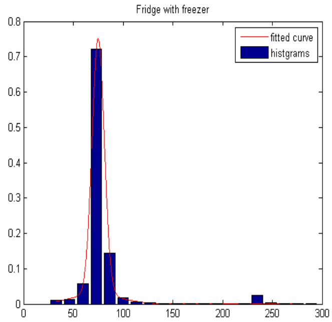
Table B.2: RMSE, mean, variance,* 1*st order correlation coefficient and* 2*nd order correlation coefficient for different TV's for different GREEND dataset houses.

Appliance	Mean value	Variance	RMSE	1st Cor.	2nd Cor.
H0 TV	55.48	11.6	0.023	0.3495	0.2982
H2 TV	359.45	78.52	0.0239	0.04	0.0152
H3 TV	90.11	9.87	7 E-03	0.3366	0.2606
H4 Kitchen TV	42.14	2.88	5.68 E-02	0.2698	0.2455
H4 living room TV	16.68	48.41	4.9 E-03	0.0962	0.0184
H5 LCD TV	35.25 56.24	2.46 1.07	3.77 E-04	0.0199	0.0768
H5 Plasma TV	144.47 201.45	13.94 34.67	8.3 E-03	0.1442	0.0746

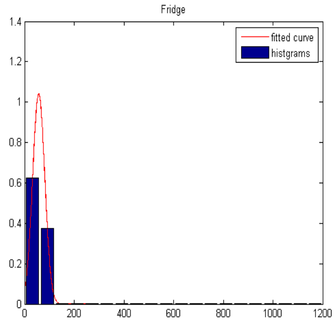
B.1.1.1 House 0

Table B.3: RMSE, mean, variance,* 1*st order correlation coefficient and* 2*nd order correlation coefficient for different appliances of GREEND dataset House 0.

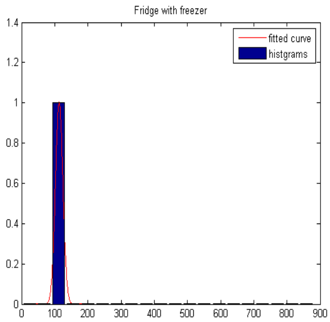
Appliance	Mean value [W]	Variance	RMSE	1st Cor.	2nd Cor.
Coffee machine	83.11	283.37	8.4 E-3	-0.0103	-0.1678
Washing machine	80.1 1955.6	93.8 73.07	4.2 E-03	-0.0751	-0.1059
Radio	8.67 10.8	0 0.1325	0.0017	-0.3196	-0.2253
Kettle	40.3 1769	137.04 20.63	0.001	0.1431	0.0014
Fridge with Freezer	73.74 236.74	9.89 6.4	0.0065	0.3669	0.3249
Dishwasher	77 1953	14.05 77.8	6.17 E-05	0.1925	0.1226
Kitchen lamp	38.81	1.04	1.2 E-04	-0.027	-0.0092
Tv	55.48	11.6	0.023	0.3495	0.2982
Vacuum cleaner	1208	164.98	0.0038	0.1005	-0.0529



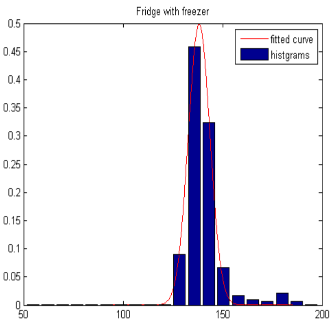
(a) House 0



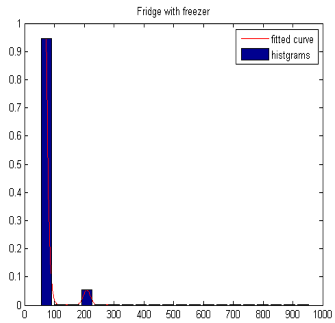
(b) House 1



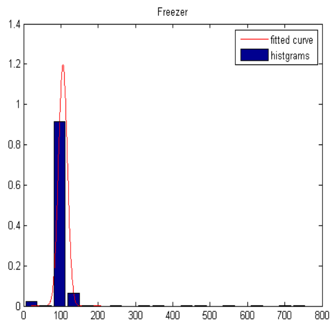
(c) House 3



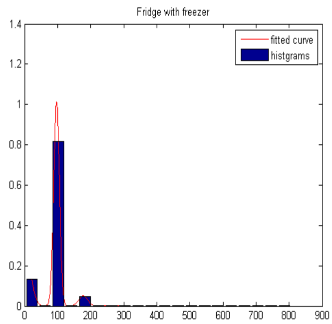
(d) House 4



(e) House 5

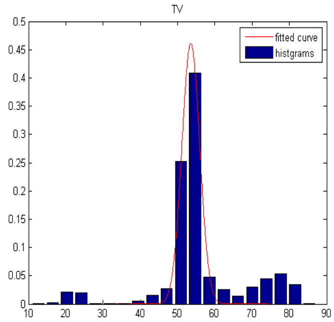


(f) House 7

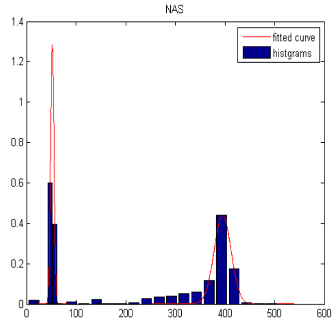


(g) House 7

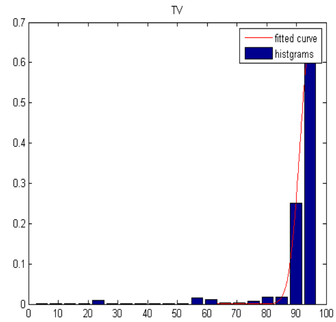
Figure B.1: Different distributions of different Fridges types from the GREEND dataset. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].



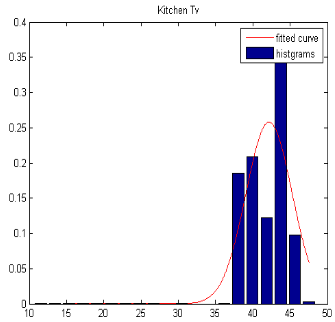
(a) House 0



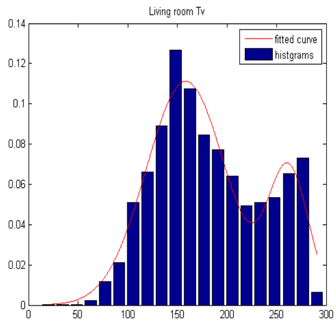
(b) House 2



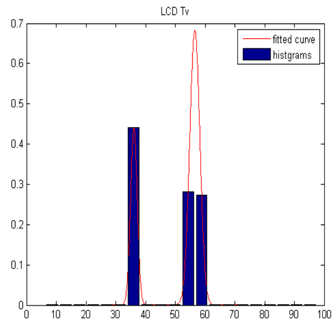
(c) House 3



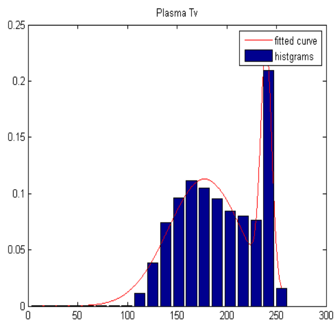
(d) House 4



(e) House 4



(f) House 5



(g) House 5

Figure B.2: Different distributions of different TV types from the GREEND dataset. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

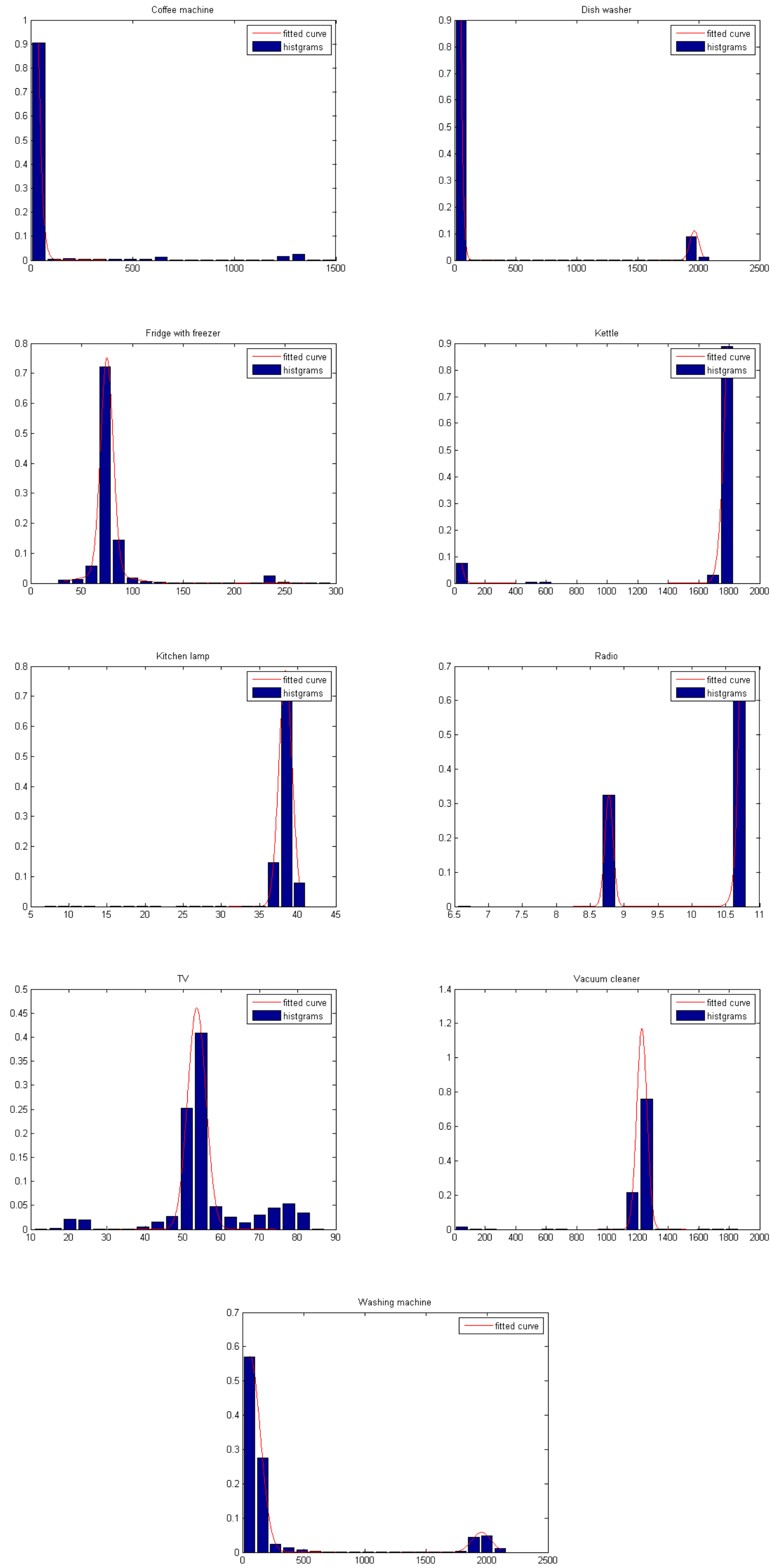


Figure B.3: Pdf for different appliances from the GREEND dataset House 0. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

B.1.1.2 House 1

Table B.4: RMSE, mean, variance, 1st order correlation coefficient and 2nd order correlation coefficient for different appliances of GREEND dataset House 1.

Appliance	Mean value [W]	Variance [W^2]	RMSE	1st Cor.	2nd Cor.
Fridge	35.2	46.48	1.56 E-4	0.0086	-0.2045
Dishwasher	13.7 1796.1	28.9 29.54	4.54 E-05	-0.042	-0.1111
Microwave	62.1 1316.6	74.19 62.42	4.9 E-03	0.0611	-0.0485
Kettle	834.51	70.82	1.10 E-03	0.1582	-0.0073
Washing machine	40.4 1991.7	73.97 90.92	3.80 E-03	0.022	-0.1126
Radio with amplifier	10.1 19.08	3.94 0	3.61 E-04	-0.0487	0.1707
Hair Drier	1569.9	152.86	1.5 E-03	0.1552	-0.0034
Bedside lamp	54.15	18.43	5.51 E-02	0.0768	0.0377

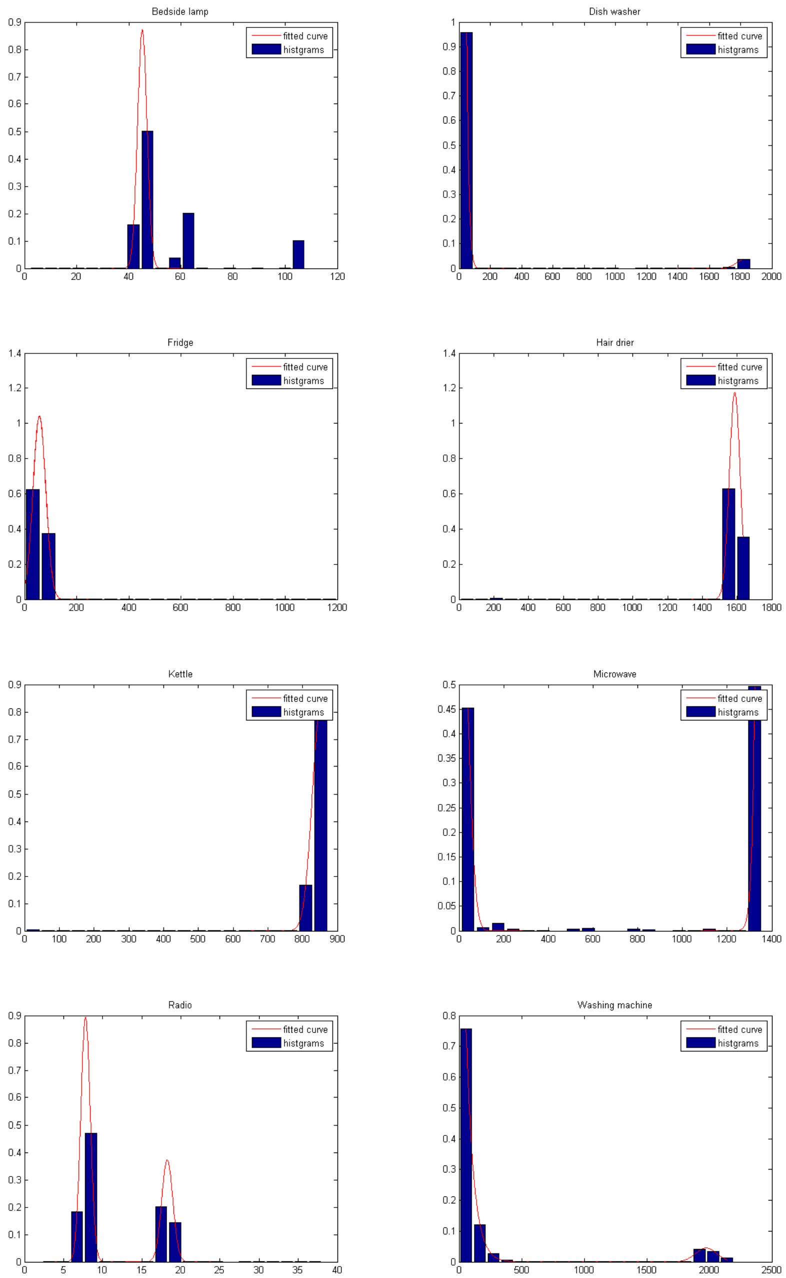


Figure B.4: Pdf for different appliances from the GREEND dataset House 1. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

B.1.1.3 House 2

Table B.5: RMSE, mean, variance, 1st order correlation coefficient and 2nd order correlation coefficient for different appliances of GREEND dataset House 2.

Appliance	Mean value [W]	Variance	RMSE	1st Cor.	2nd Cor.
Tv	359.45	78.52	0.0239	0.04	0.0152
NAS	52.56	9.54	7.90 E-04	0.0106	0.0251
Washing machine	47.2 2081	88.84 58.3694	7.20 E-03	0.082	0.0317
Tumble drier	87.1 2558.2	94.7 72.21	5.94 E-04	0.0013	-0.0479
Dishwasher	18.1 2071.3	33.19 38.79	9.97 E-05	-0.066	-0.1186
Notebook	24.99	19.61	4.8 E-02	-0.1024	-0.1368
Coffee machine	21.44	150.01	1.7 E-03	-0.1272	-0.1235
Bread maker	98.83	36.26	2.97 E-02	-0.0072	-0.265

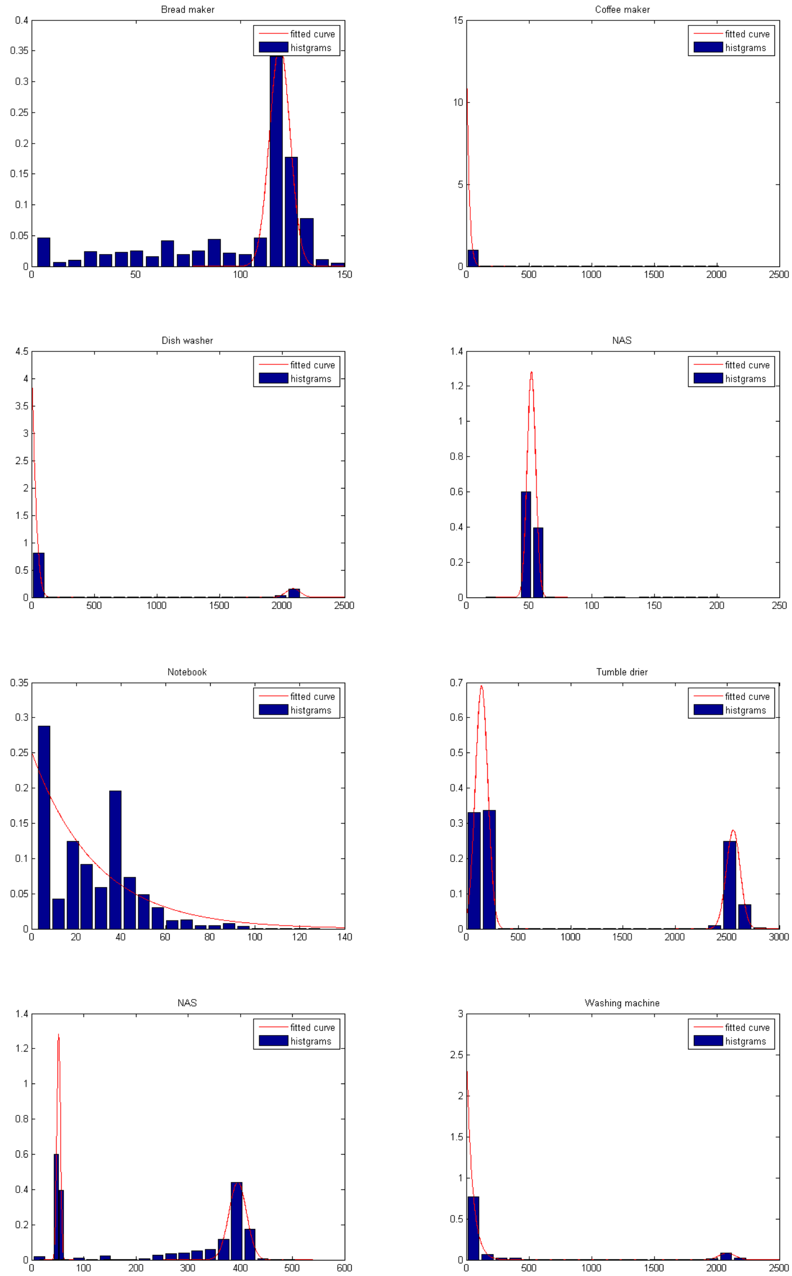


Figure B.5: Pdf for different appliances from the GREEND dataset House 2. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

B.1.1.4 House 3

Table B.6: RMSE, mean, variance, 1st order correlation coefficient and 2nd order correlation coefficient for different appliances of GREEND dataset House 3.

Appliance	Mean value [W]	Variance	RMSE	1st Cor.	2nd Cor.
Entrance outlet	1258	141.92	0.0027	0.0469	-0.036
Dishwasher	39.9 1760.7	36.8 24.1	1.15 E-04	0.2213	0.1674
Kettle	1955.6	177.94	1.7 E-0.3	0.0905	-0.008
Fridge with freezer	109.57	6.69	7.18 E-05	0.5009	0.4663
Washing machine	94.7 1957.8	115.59 69.51	1.91 E-02	0.0609	-0.0838
Hair drier	527 1117.1 1867.9	195.84 13.68 11.8	7.74 E-04	0.1208	-0.0284
Computer	49.76 80.69	7.18 4.54	2.12 E-02	0.8874	0.8708
Coffee machine	49.7 531.2 1155.1	1.07 312.55 25.9	6.1 E-03	0.0359	-0.2965
Tv	90.11	9.87	7 E-03	0.3366	0.2606

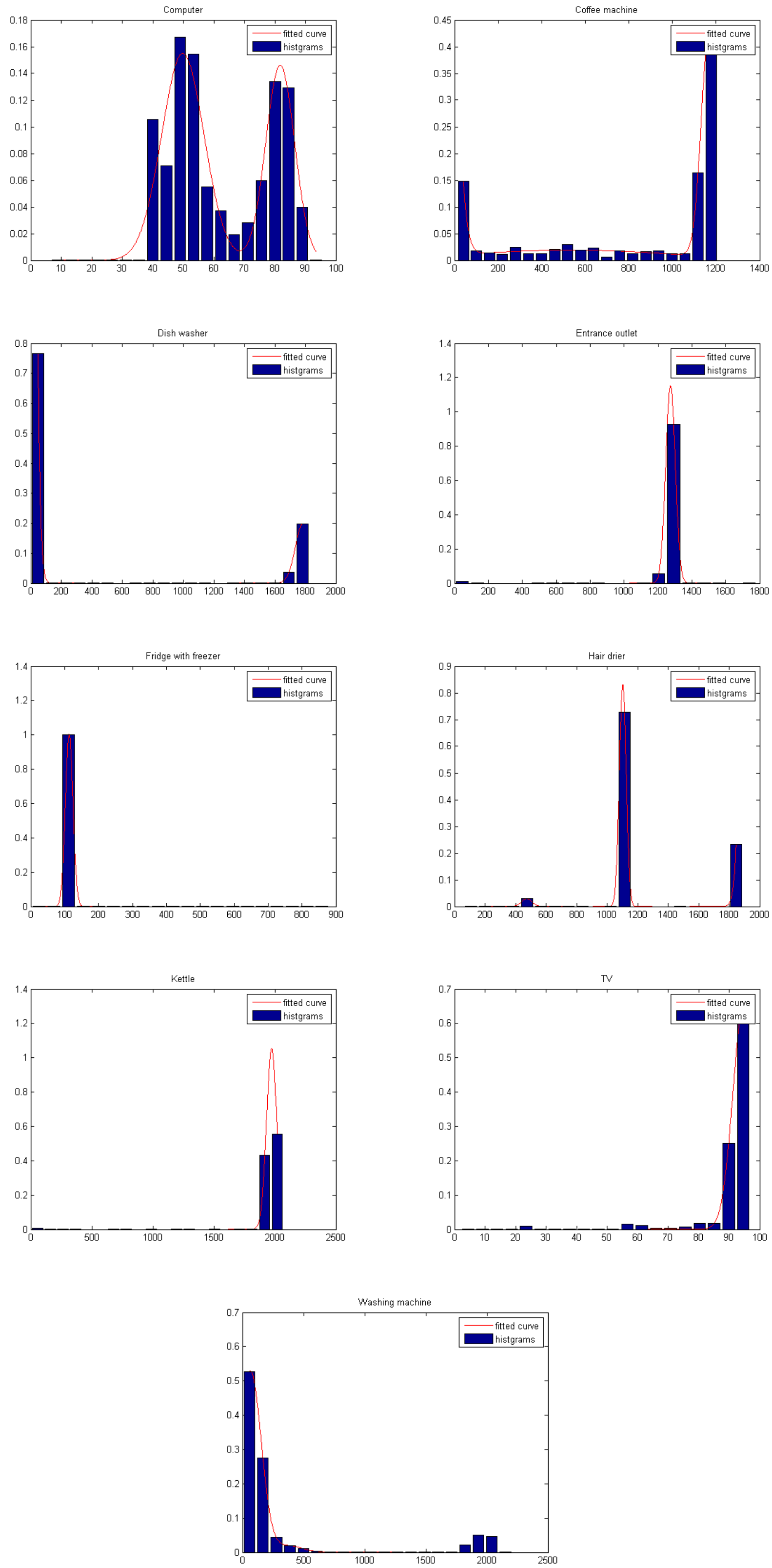


Figure B.6: Pdf for different appliances from the GREEND dataset House 3. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

B.1.1.5 House 4

Table B.7: RMSE, mean, variance, 1st order correlation coefficient and 2nd order correlation coefficient for different appliances of GREEND dataset House 4.

Appliance	Mean value [W]	Variance	RMSE	1st Cor.	2nd Cor.
Kitchen Tv	42.14	2.88	5.68 E-02	0.2698	0.2455
Living R Tv	16.68	48.41	4.9 E-03	0.0962	0.0184
Fridge w/ freezer	140.27	000	1.04 E-02	0.2934	0.2016
Electric oven	31.4 1618.2	119.42 28.19	4.21 E-04	0.021	-0.0712
Comp. w/ scanner & printer	25.7 814 1028.4	16.57 67.96 20.5	7.42 E-05	0.2083	0.1518
Washing machine	54.9 597.4 1946.1	93.71 13.77 224	3.3 E-03	0.0298	-0.2105
Hood	15.93 144.12	1.05 23.45	1.41 E-02	-0.0382	0.0253

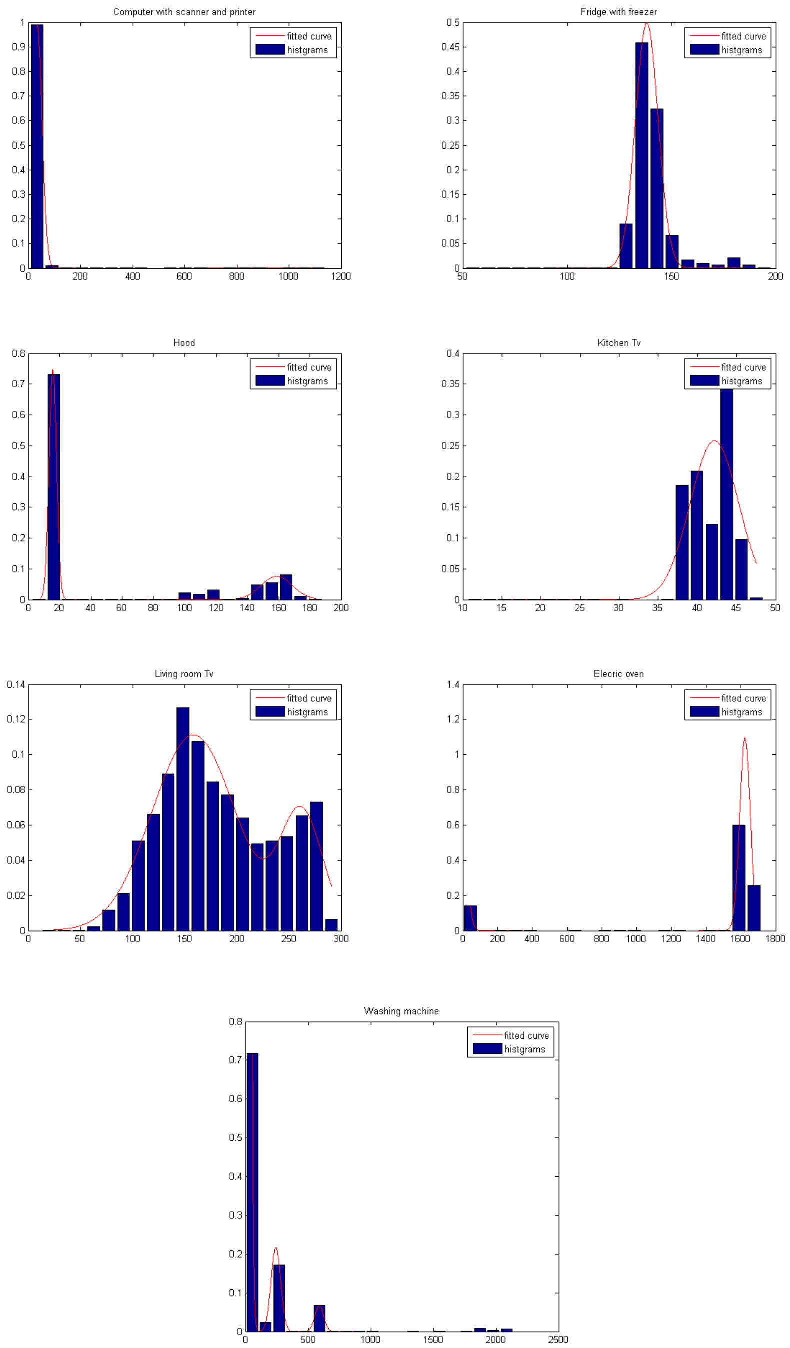


Figure B.7: Pdf for different appliances from the GREEND dataset House 4. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

B.1.1.6 House 5

Table B.8: RMSE, mean, variance,* 1*st order correlation coefficient and* 2*nd order correlation coefficient for different appliances of GREEND dataset House 5.

Appliance	Mean value [W]	Variance	RMSE	1st Cor.	2nd Cor.
Plasma Tv	144.4 201.45	13.94 34.67	8.3 E-03	0.1442	0.0746
Lamp	4.52 205.79	5.99 2.64	2.4 E-04	-0.1311	0.0631
Toaster	747.5 1476.7	131.55 27.66	2.3 E-03	0.1285	-0.0502
Stove	677.7 1353.1	58.2 26.98	1.2 E-03	0.0682	-0.0144
Iron	678 728.3 1235 1849.4	6.62 234.5 29.9471 34.23	3.2 E-02	0.0236	-0.0417
Comp. w/ scanner & printer	2.19 16.87	0 8.56	1.33 E-02	0.1909	0.1247
LCD Tv	35.25 56.24	2.46 1.07	3.77 E-04	0.0199	0.0768
Washing machine	31.3 1866.1	54.95 68.14	1.4 E-03	0.0222	-0.1068
Fridge w/ freezer	64.82 202.72	3.14 66.24	4.3 E-05	0.0952	0.066

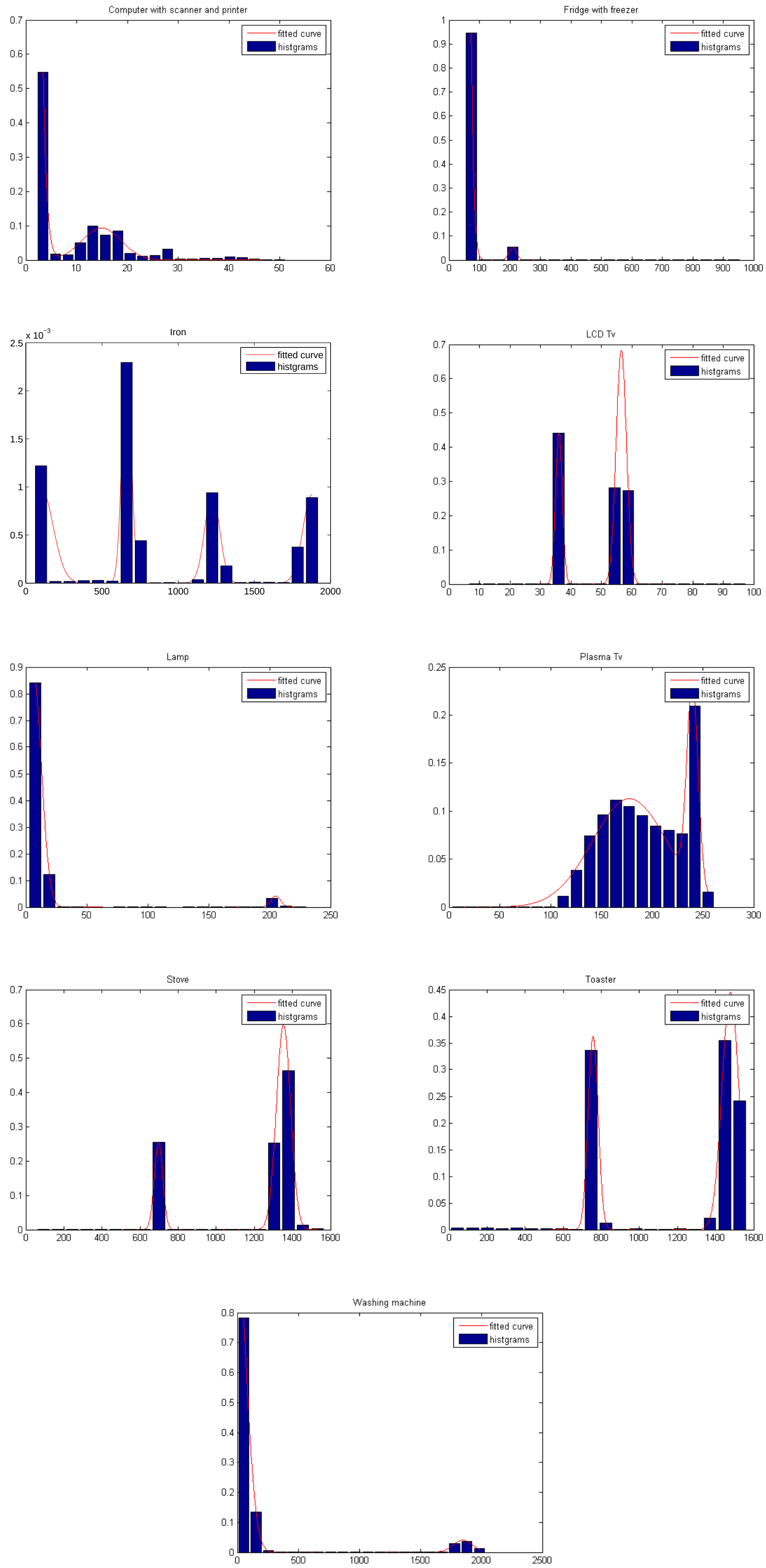


Figure B.8: Pdf for different appliances from the GREEND dataset House 5. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

B.1.1.7 House 7

Table B.9: RMSE, mean, variance, 1st order correlation coefficient and 2nd order correlation coefficient for different appliances of GREEND dataset House 7.

Appliance	Mean value [W]	Variance	RMSE	1st Cor.	2nd Cor.
Tv with decoder	10.94 112.48	0 14.03	3.51 E-02	-0.011	-0.0177
Electric oven	1470	150.81	2.8 E-03	-0.0479	0.0032
Hood	6.58 173.14	12.41 1.94	1.33 E-04	0.03	-0.0023
Fridge with freezer	88.52 426.68 641.16	37.04 136.86 29.07	1.65 E-04	0.2272	0.1632
Kitchen Tv	26.07 98.61	0.69 1.39	2.03 E-04	0.0155	0.0113
ADSL modem	2.19 16.87	0 8.56	1.33 E-02	0.1909	0.1247
Freezer	101 56.24	21.97 1.07	5.1 E-03	0.0448	-0.0273
Kitchen Tv	2.6 34.06	0 12.11	1 E-02	0.2447	0.2939

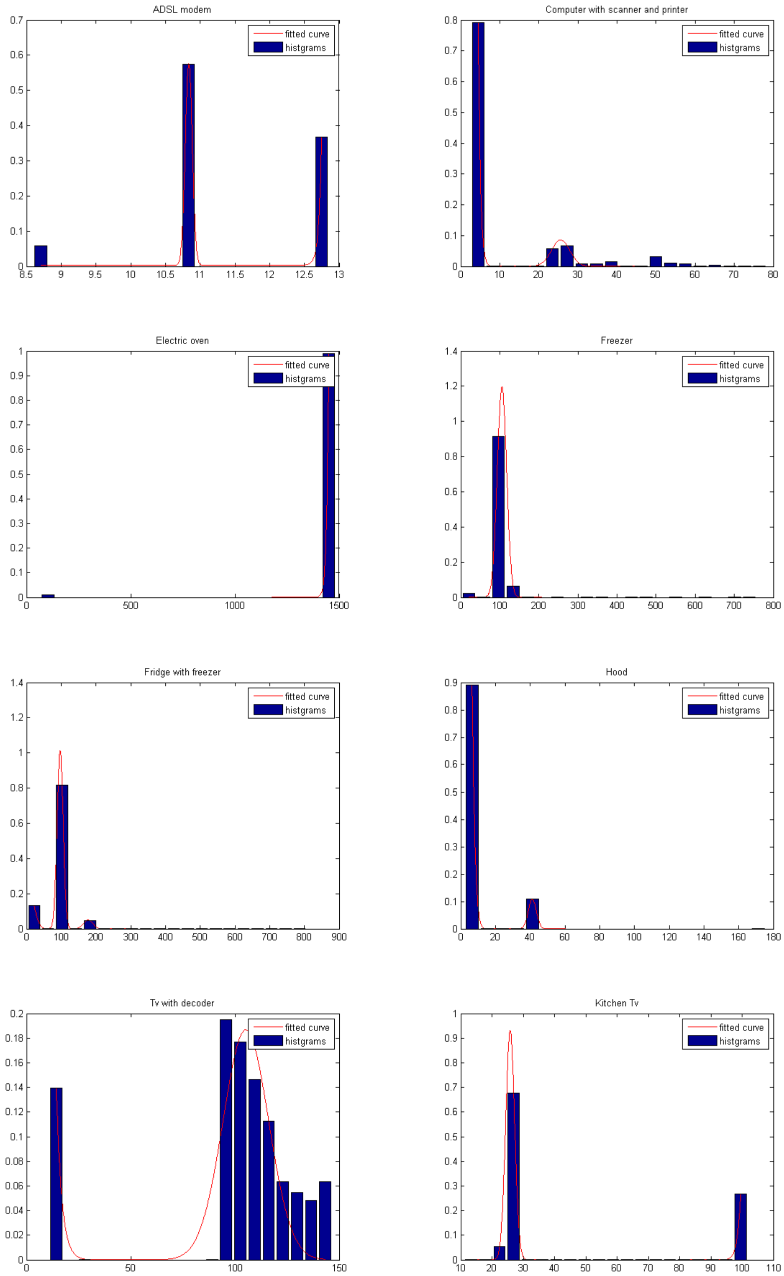


Figure B.9: Pdf for different appliances from the GREEND dataset House 7. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

B.1.2 REFIT Dataset

B.1.2.1 House 1

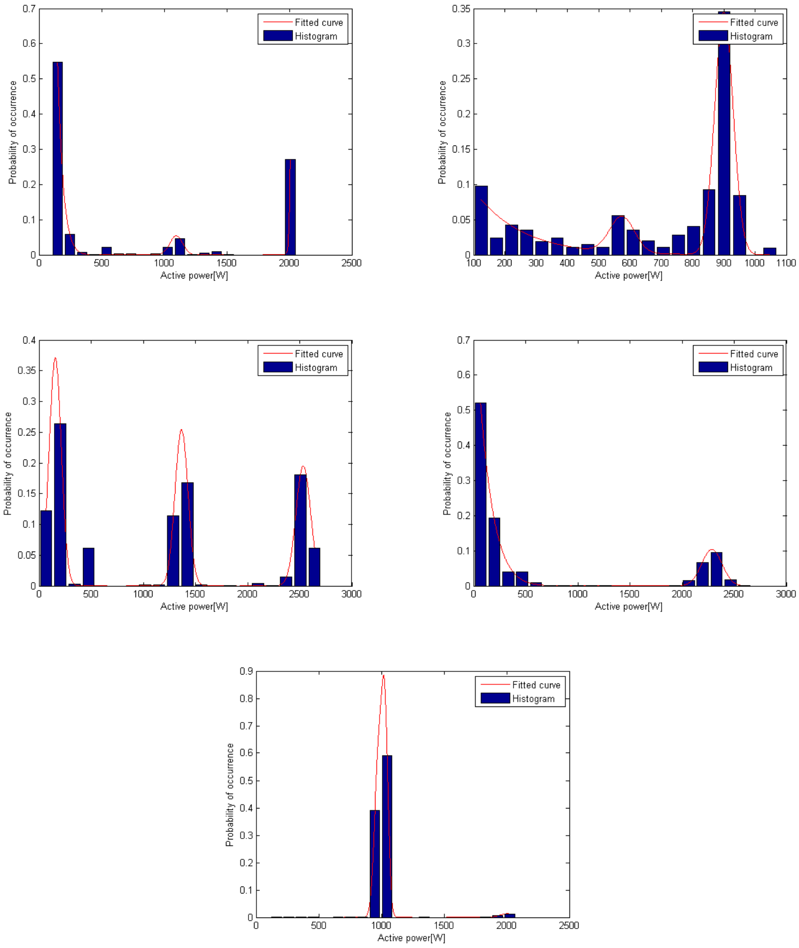


Figure B.10: Pdf for different appliances from the REFIT dataset House 1. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

B.1.2.2 House 2

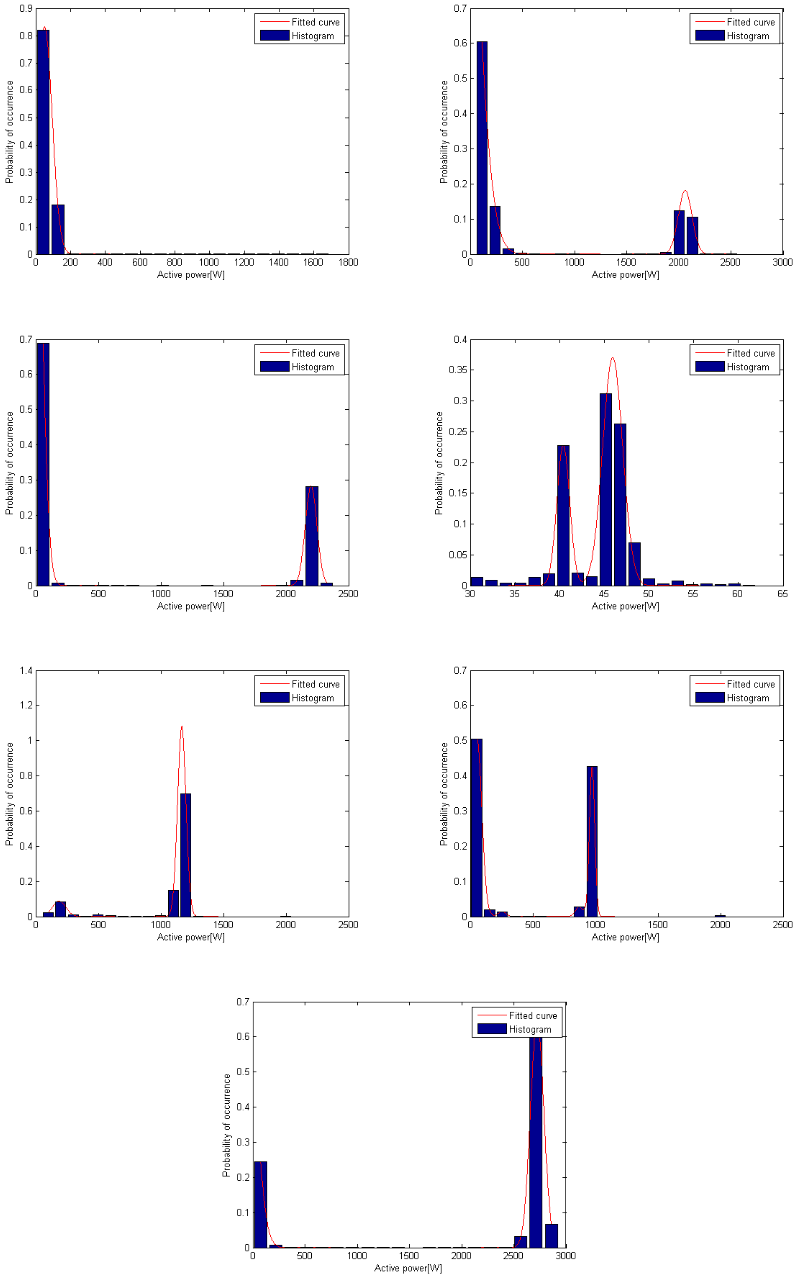


Figure B.11: Pdf for different appliances from the REFIT dataset House 2. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

B.1.2.3 House 3

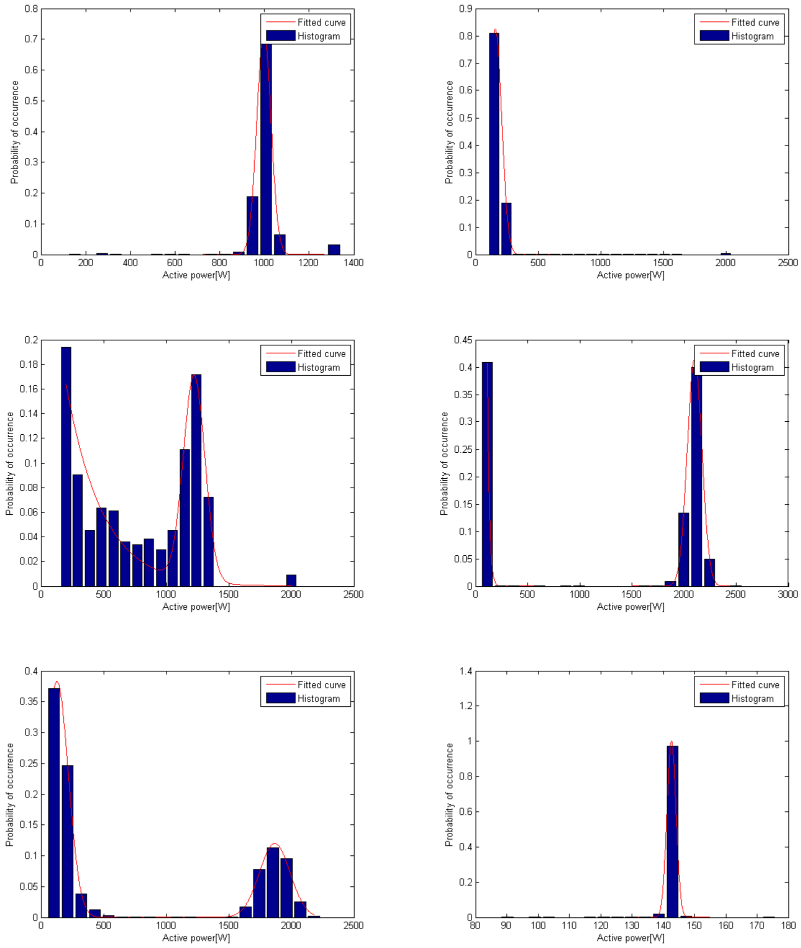


Figure B.12: Pdf for different appliances from the REFIT dataset House 3. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

B.1.2.4 House 4

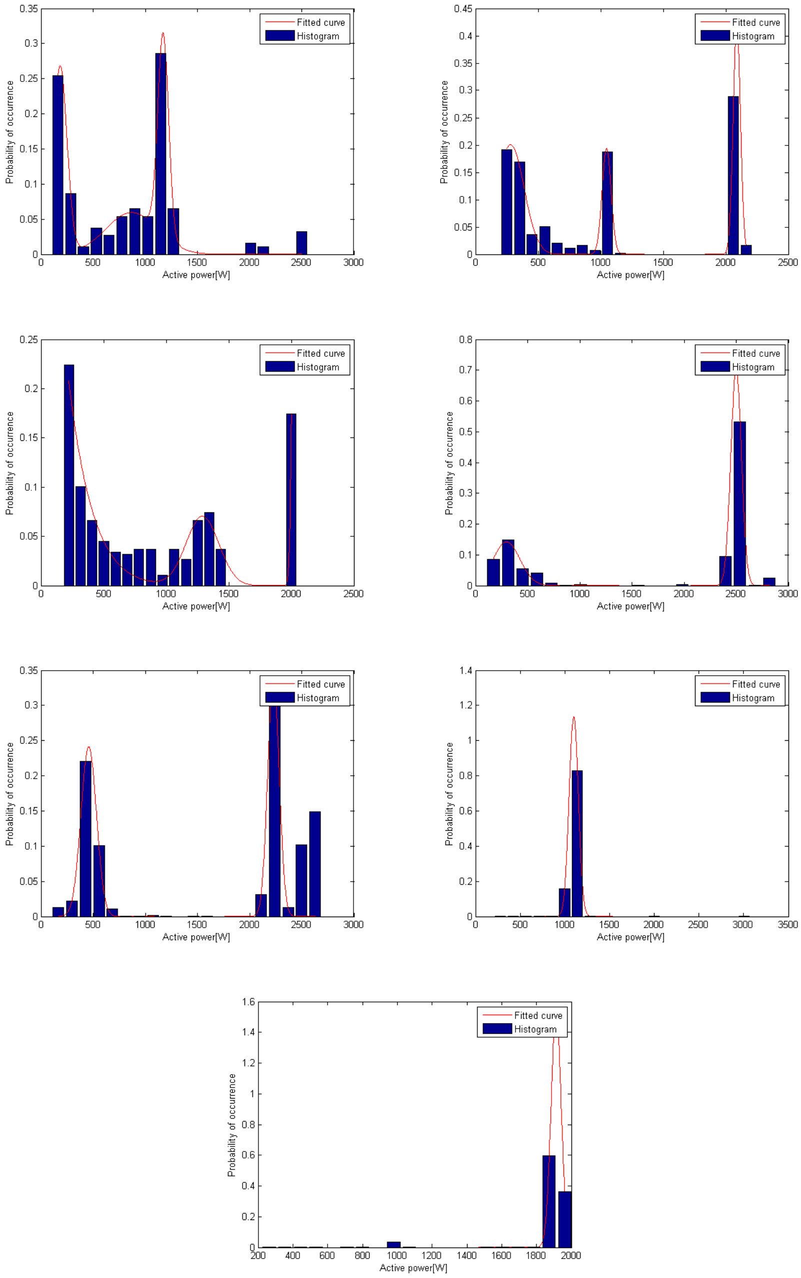


Figure B.13: Pdf for different appliances from the REFIT dataset House 4. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

B.1.2.5 House 5

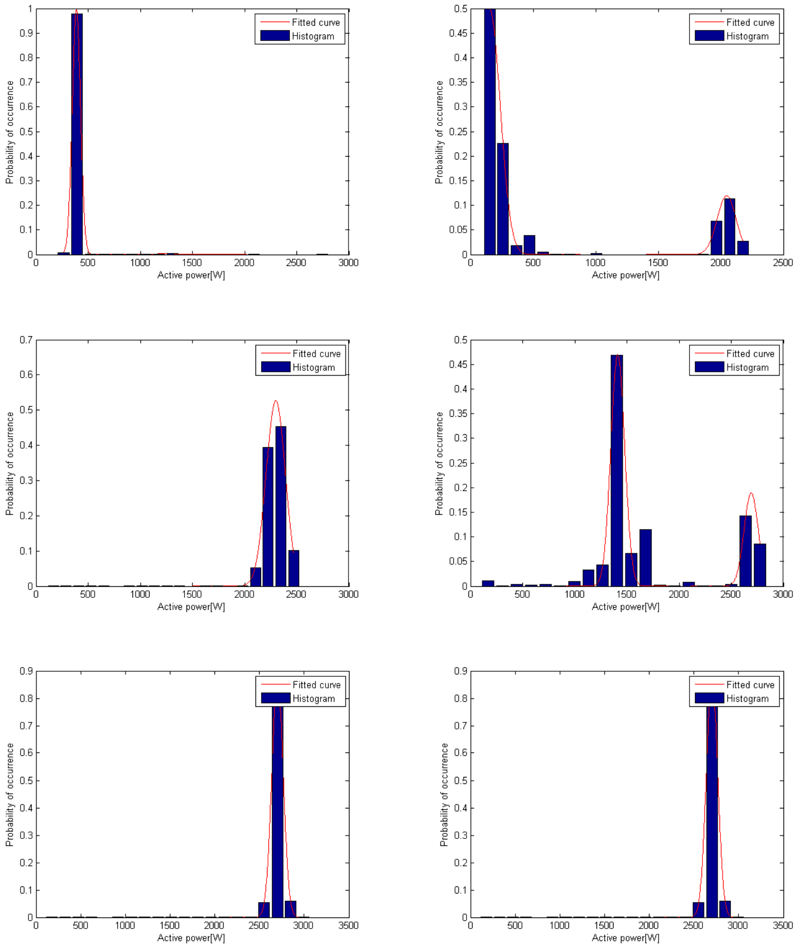


Figure B.14: Pdf for different appliances from the REFIT dataset House 5. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

B.1.2.6 House 6

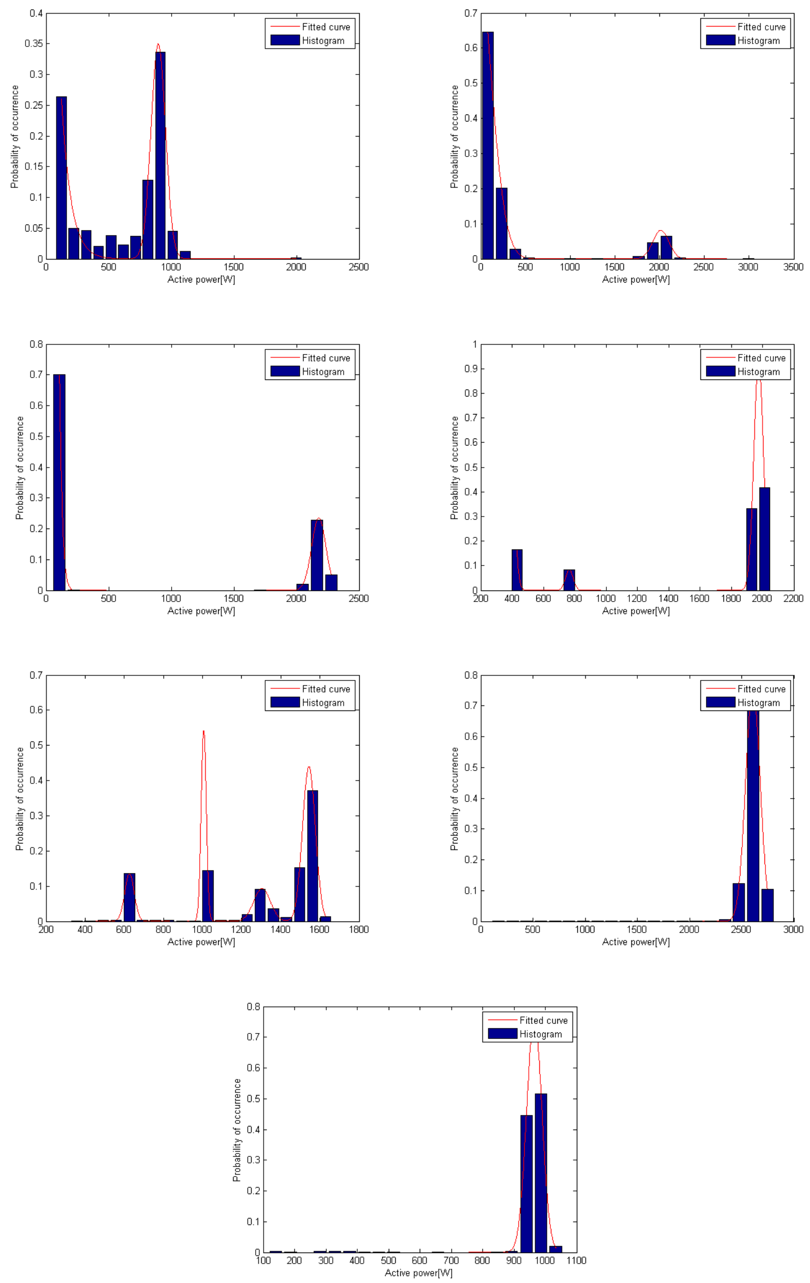


Figure B.15: Pdf for different appliances from the REFIT dataset House 6. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

B.1.2.7 House 7

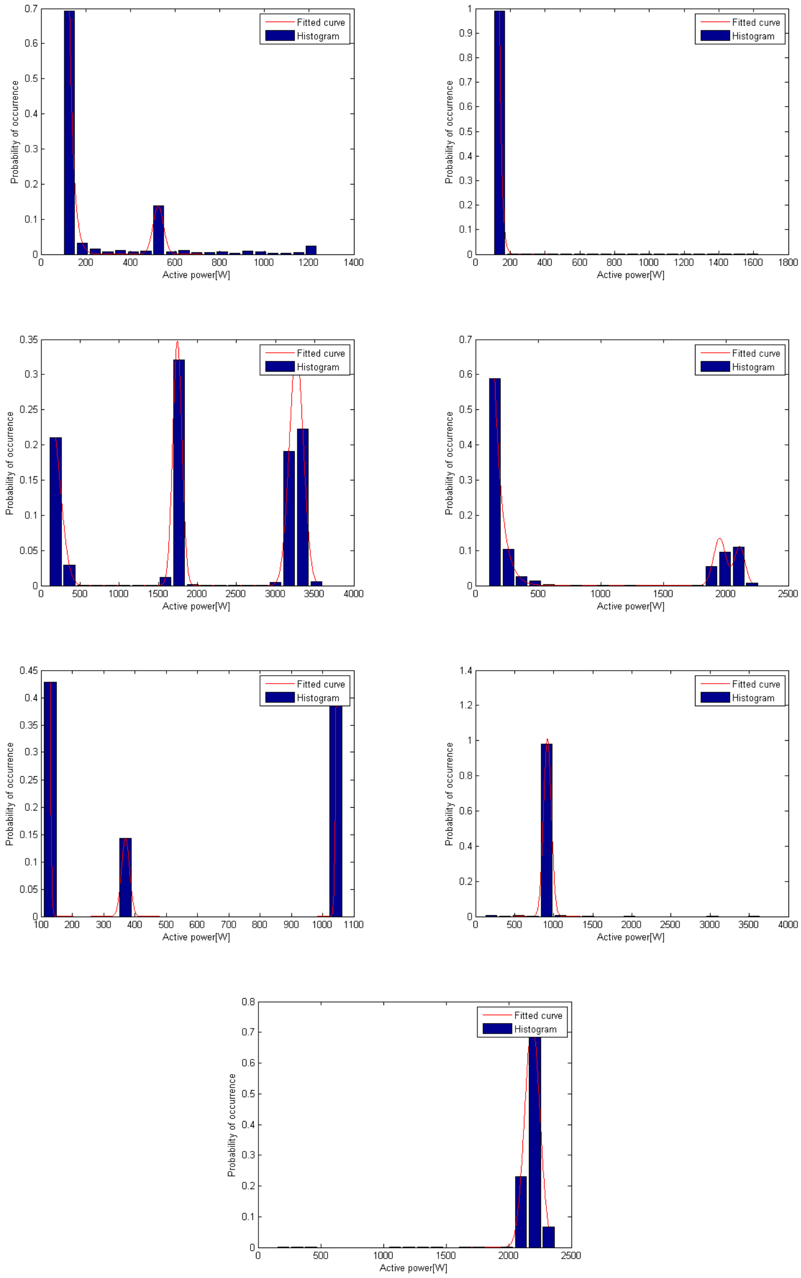


Figure B.16: Pdf for different appliances from the REFIT dataset House 7. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

B.1.3 General Modelling Results

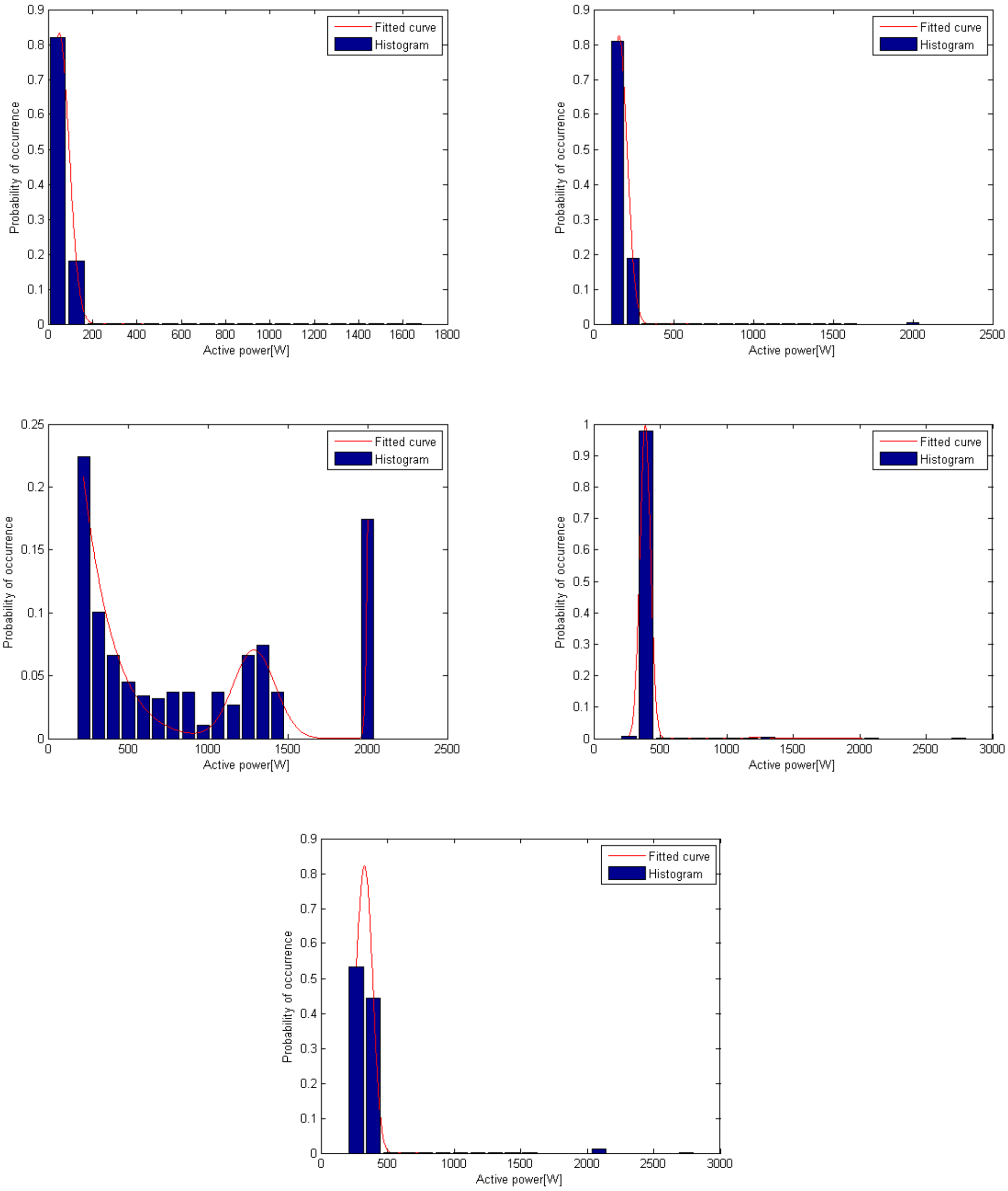


Figure B.17: Different distributions of different Fridge-Freezer types from the REFIT dataset. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

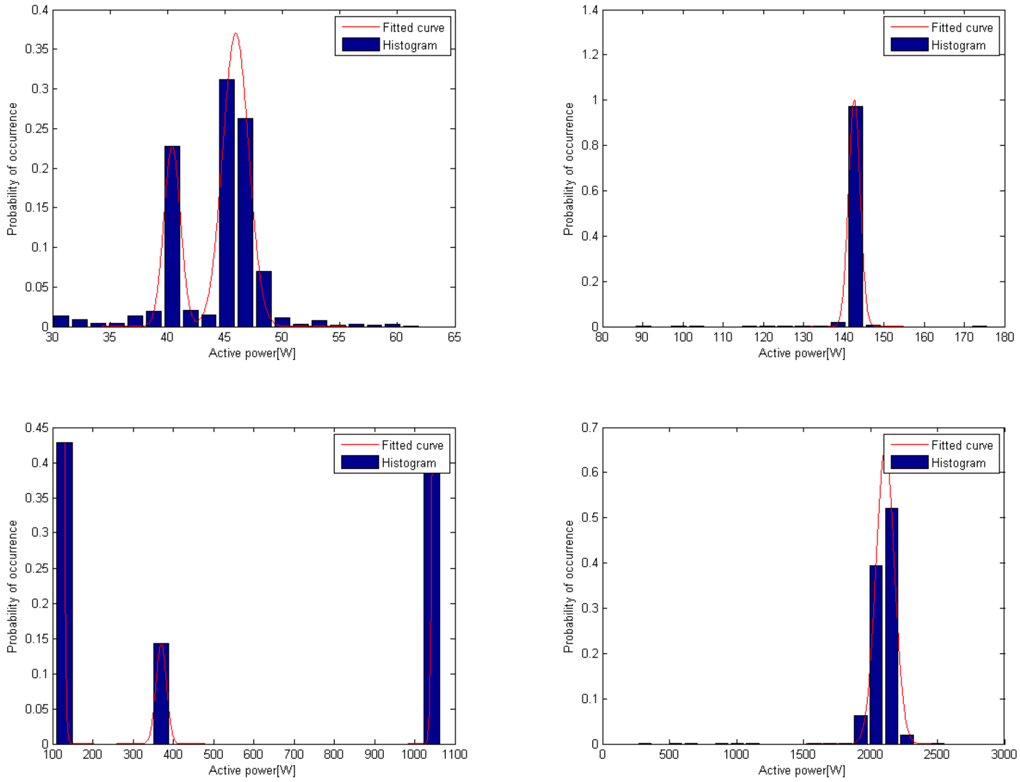


Figure B.18: Different distributions of different television types from the REFIT dataset. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

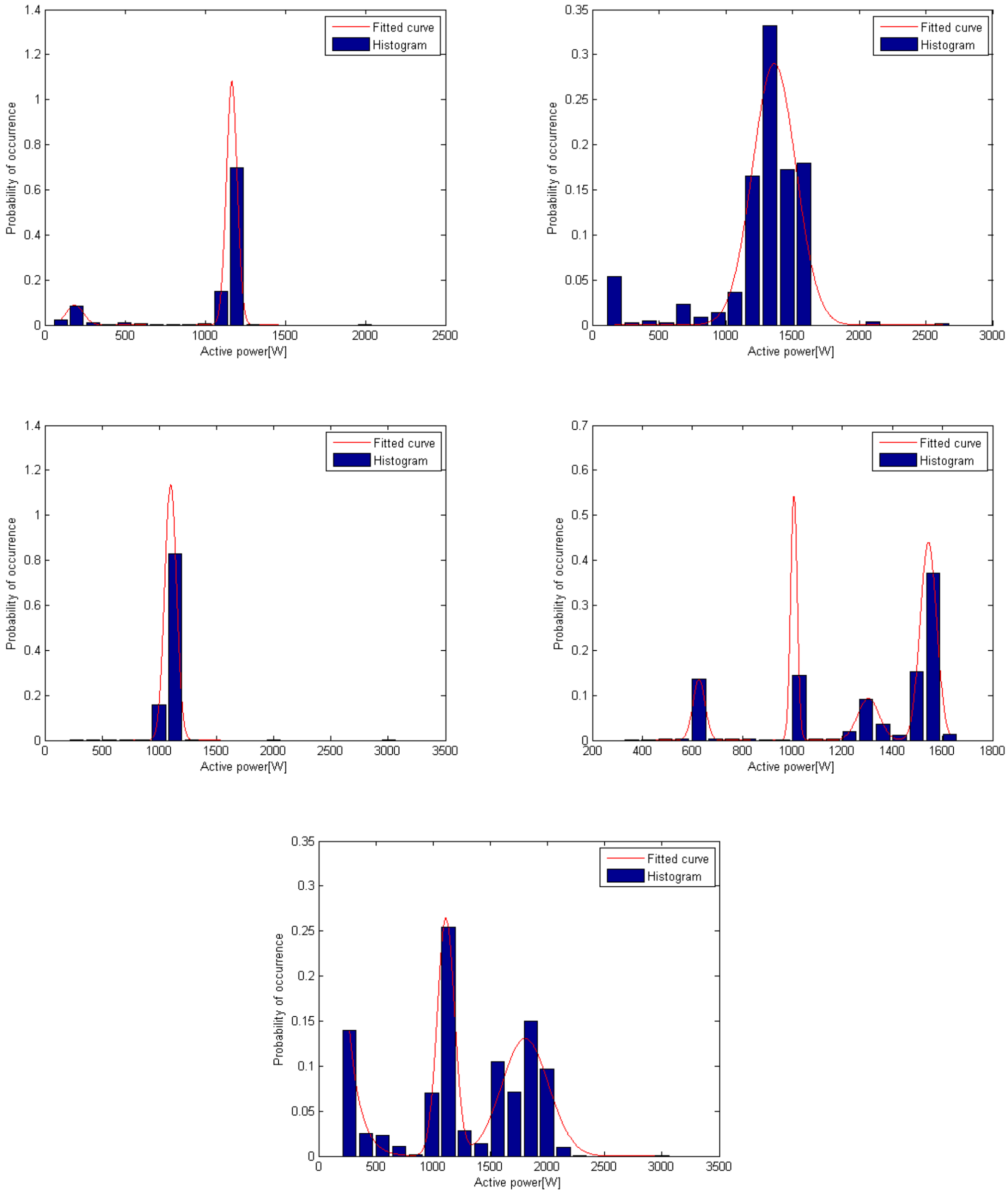


Figure B.19: Different distributions of different microwave types from the REFIT dataset. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

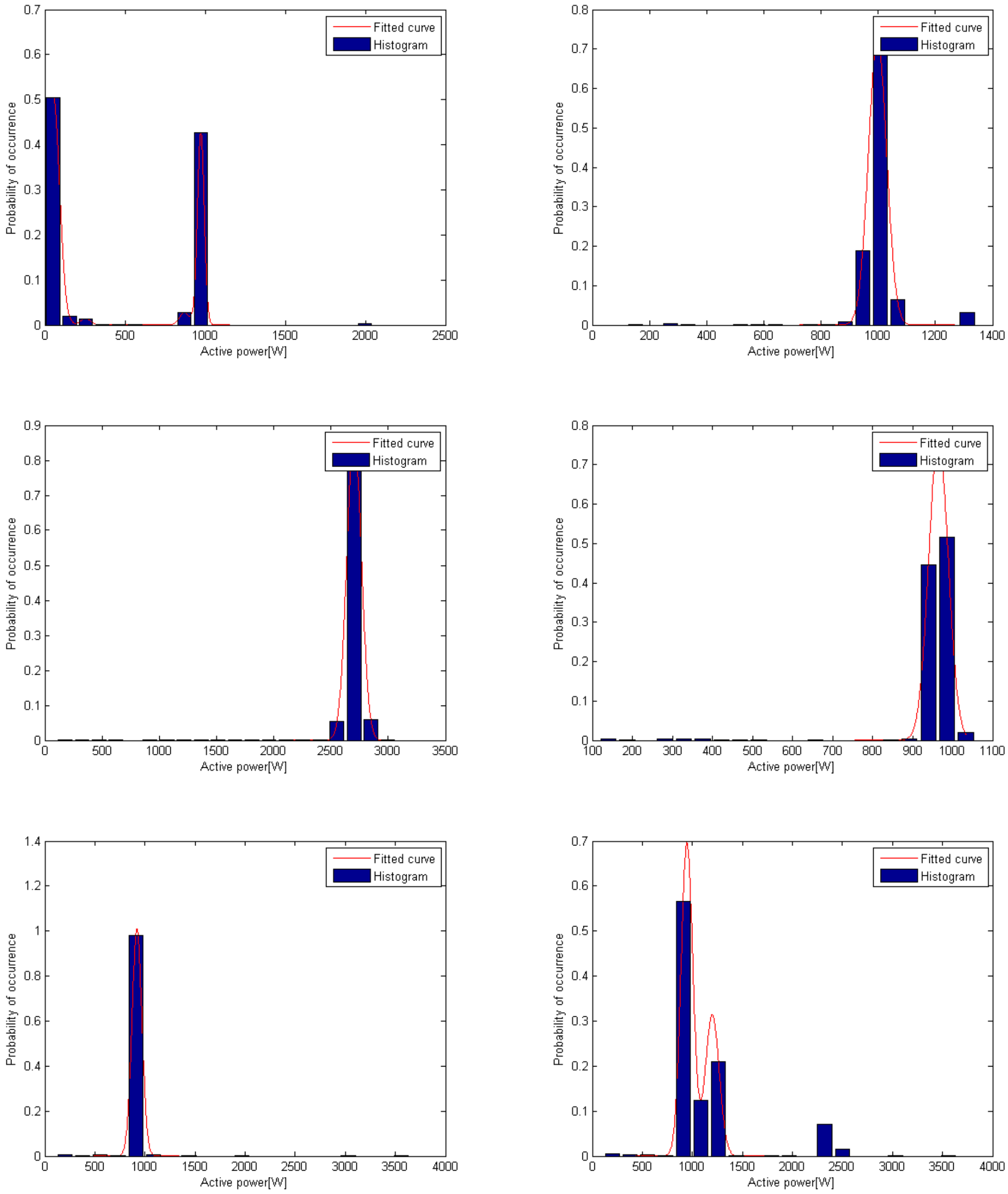


Figure B.20: Different distributions of different toaster types from the REFIT dataset. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

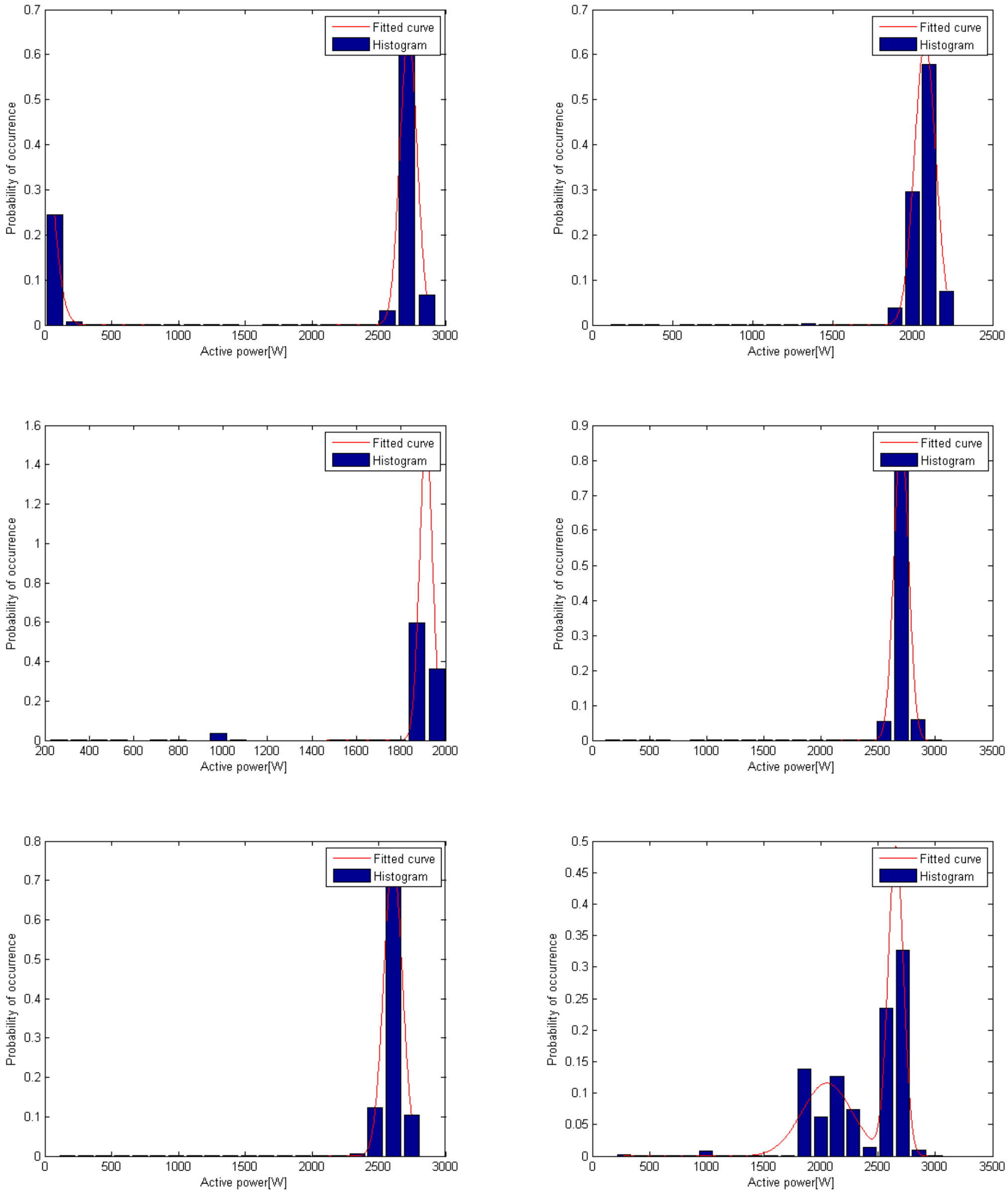


Figure B.21: Different distributions of different kettle types from the REFIT dataset. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

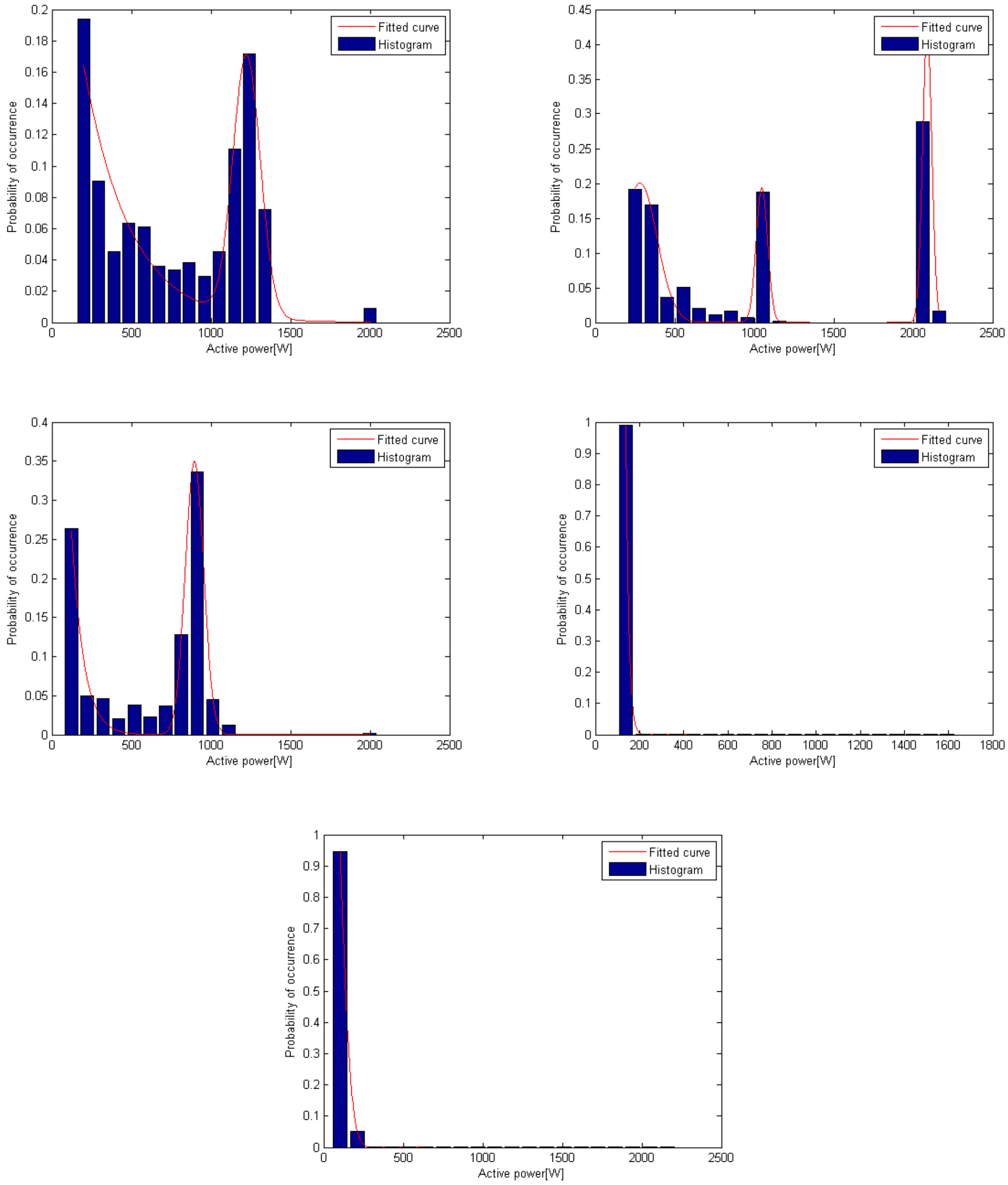


Figure B.22: Different distributions of different tumble dryer types from the REFIT dataset. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

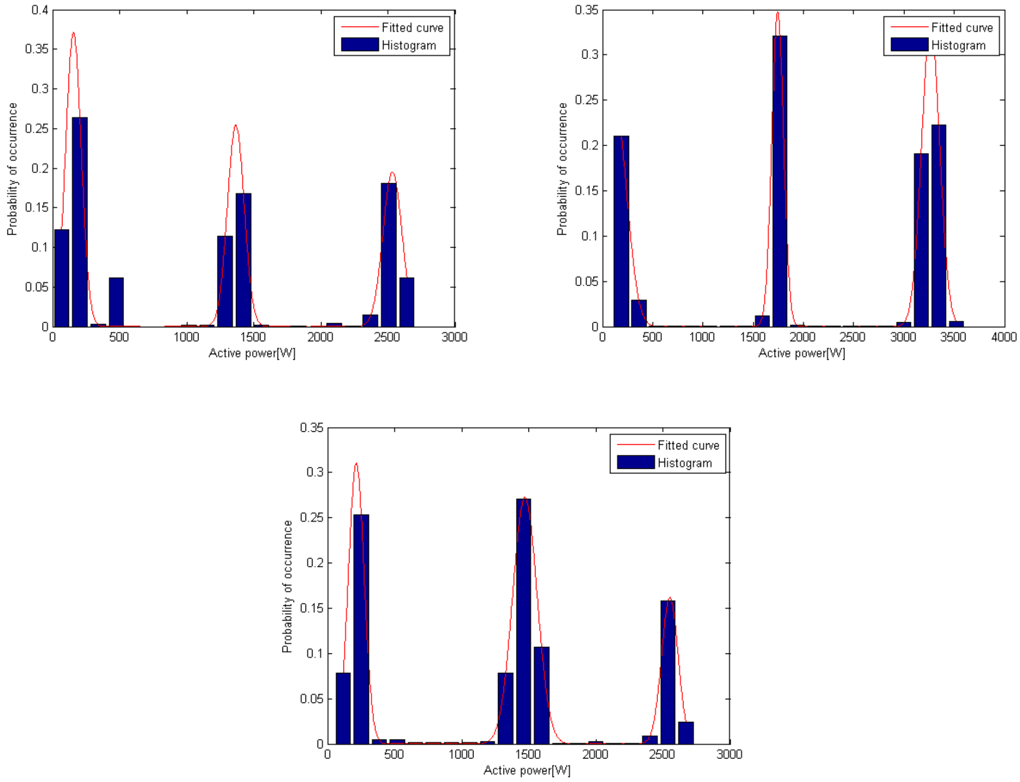


Figure B.23: Different distributions of different freezer types from the REFIT dataset. Histograms are showing true data obtained via sub-metering. x-axis shows active power in [W].

Table B.10: RMSE, mean, variance,** 1**st order correlation coefficient and** 2**nd order correlation coefficient for different fridge-freezer of different REFIT dataset houses. GM denotes the general model.

Appliance	Mean value	Variance	RMSE	1st Cor.	2nd Cor.
H2	34.7839	45.46	2.16 E-04	0.8751	0.0059
H3	137.92	127.36	8.84 E-04	0.022	-0.1126
H4	837.7 1291 2039	18.48 185.5 32.69	0.0199	-0.002	0.7333
H5	387.6 1248	56.06 137.7	7.50 E-04	0.0098	0.7105
GM	308.7 1125	85.64 167.76	0.0032		

Table B.11: RMSE, mean, variance, 1st order correlation coefficient and 2nd order correlation coefficient for different televisions of different REFIT dataset houses. GM denotes the general model.

Appliance	Mean value	Variance	RMSE	1st Cor.	2nd Cor.
H2	42.20 46.54	4.92 0.6325	1.49 E-02	-0.0701	-0.1159
H3	142.7	2.046	3.02 E-04	-0.022	-0.1026
H7	126.17 370.3 1064	23.73 18.41 12.66	6.71 E-05	-0.082	0.0017
GM	2110	97.25	0.00121		

Table B.12: RMSE, mean, variance, 1st order correlation coefficient and 2nd order correlation coefficient for different microwaves of different REFIT dataset houses. GM denotes the general model.

Appliance	Mean value	Variance	RMSE	1st Cor.	2nd Cor.
H3	1276.5 1955.6	340.07 73.07	3.16 E-02	0.0632	0.4578
H4	1124.7	102.84	0.0013	0.0532	0.1126
H6	626.5 1006 1304 1544	34.67 18.62 626.5 46.41	0.0038	0.0087	0.2117
GM	152 1113 1805	62 106.1 289.7	0.00222		

Table B.13: RMSE, mean, variance, 1st order correlation coefficient and 2nd order correlation coefficient for different toasters of different REFIT dataset houses. GM denotes the general model.

Appliance	Mean value	Variance	RMSE	1st Cor.	2nd Cor.
H2	12.5 161.1 950.6 941.9 2047.6	13.78 117.61 11.71 20.93 1.14	0.0017	0.0841	0.1439
H3	1003.1	83.22	0.0081	-0.0642	0.2226
H5	2685.2	193.68	8.90 E-04	0.0512	0.0817
H6	955.06	88.53	0.0025	0.0609	-0.0838
H7	914.54	135.91	0.0019	0.0578	0.0085
GM	945.7 1199	83.45 98.06	0.0195		

Table B.14: RMSE, mean, variance, 1st order correlation coefficient and 2nd order correlation coefficient for different kettles of different REFIT dataset houses. GM denotes the general model.

Appliance	Mean value	Variance	RMSE	1st Cor.	2nd Cor.
H2	63.2 2722.9	189.03 52.29	4.2 E-3	0.0751	-0.1059
H3	2061.1	133.09	6.90 E-03	0.0022	-0.1126
H4	963.9 1909.2	263.56 29.09	7.20 E-03	0.0802	0.0227
H5	2685.2 1957.8	193.68 69.51	8.90 E-04	0.0609	-0.0838
H6	2593.3	177.55	0.0017	0.0296	-0.2065
GM	2052 2653	310.7 94.74	0.0241		

Table B.15: RMSE, mean, variance,* 1*st order correlation coefficient and* 2*nd order correlation coefficient for different freezers of different REFIT dataset houses. GM denotes the general model.

Appliance	Mean value	Variance	RMSE	1st Cor.	2nd Cor.
H3	533.3 1233	362.67 67.16	0.0189	0.0221	0.0059
H4	280 1047 2088	153.7 50.93 41.49	3.8 E-3	0.0225	0.1126
H6	241.18 895.35	228.58 70.18	0.0181	-0.082	-0.0317
H7	109.24	99.07	6.08 E-042	0.0909	0.0138
GM	39.01	94.01	3.97 E-04		

Table B.16: RMSE, mean, variance, 1st order correlation coefficient and 2nd order correlation coefficient for different tumble dryers of different REFIT dataset houses. GM denotes the general model.

Appliance	Mean value	Variance	RMSE	1st Cor.	2nd Cor.
H1	50 1448	98.5 33.17	4.2 E-3	0.0751	0.1839
H3	727.07 1155.9 2119.02	66.37 224 128.7	3.8 E-3	0.0129	-0.1126
H5	48.2 1081	88.84 123.36	7.20 E-03	0.0082	0.0377
H7	195	115.59	1.91 E-2	0.0669	-0.0838
GM	214.4 1472 2552	83.02 128.7 87.31	3.2 E-3		

B.2 Appliance-Load Clustering Results

B.2.1 REFIT Dataset Clustering Results

B.2.1.1 Mean-shift Clustering

Table B.21 and Figure B.24 (upper figure) show groups of REFIT House 2 appliances after clustering using mean-shift method into 3 groups. It can be seen that groups can vary in the number of appliances or appliance-state they hold. Group 1, contains medium to high mean and variance values, such as washing machine second state, dishwasher second state and toaster highest state. Group 3 has relatively high states with microwave second state and toaster third and fourth states. Group 2, show slightly higher number of appliances compared to Groups 1 and 2, therefore, it was reclustered in further sub-groups.

In Table B.22 and Figure B.24 (middle figure) it can be seen that Group 2 was reclustered into 3 groups, group 2.1 and 2.3 show only one appliance each, microwave and kettle first states. Group 2.2 still holds higher number of appliances and was reclustered into further groups. In Table B.23 and Figure B.24 (lower figure) Group 2.2 was reclustered into 3 further groups, it can be seen that fridge/freezer first state, washing machine first state and toaster first and second states hold similar

Gaussian characteristics as they were grouped into the same group. Sub-group 2.2.1, has only television second state, but television first state was in a different group with dishwasher low state.

Table B.24 and Figure B.25 (top figure) show groups of REFIT House 3 appliances after clustering using mean-shift method into 3 groups. It can be seen that groups can vary in the number of appliances or appliance-state they hold. Groups 2 and 3 have relatively lower number of appliances, Group 2 contains high appliances-states such as tumble dryer, dishwasher and washing machine second state and kettle first state which means that they hold similar Gaussian characteristics. Group 3 represents medium mean and variance values toaster and microwave first states and freezer second state.

Group 1 shows slightly higher number of appliances, therefore, it was reclustered into further sub-groups which are shown in Table B.25 and Figure B.24 (bottom figure). It can be seen that Group 1.2 and 1.3 contain only one appliance each which are tumble dryer and fridge/freezer first states. Group 1.1 contains low mean and variance values states such as freezer, dishwasher, washing machine and television first states.

B.2.1.2 Tree Clustering

Figure B.26 shows groups of REFIT House 2 appliances after clustering using Tree method. It can be seen that 16 appliance-states are present in House 2 forming a tree of appliance-signatures. The tree was cut into suitable level forming 4 main groups of 3, 9, 3 and one appliance state. The group with 9 appliance-states was then re-cut into further groups, forming four groups. Table B.26 shows all groups and sub-groups from House 2 tree clustering, it can be seen that Group 1 contains kettle second state only, which means that it was different that all other appliance signatures. Group 2 shows a good connection between medium to high mean and variance values such as dishwasher, washing machine second states.

Group 3 shows a connection between low to medium consumption appliances which was sub-groups into 4 groups, Group 3.1 contains microwave and kettle first states and toaster second state. Group 3.2 was cut into further 2 sub-groups, Group 3.2.1 shows a similarity between fridge-freezer and toaster first states, while Group 3.2.2 shows a connection between television first and second state, dishwasher and

Table B.17: Groups of all appliances GREEND dataset based on Mean-shift clustering method. H denotes House number. DW=dishwasher. WM=washing machine. TD=tumble dryer. s denotes state number.

Group Number	*Appliance*
1	H2 TV H4 hood s2, H5 plasma TV s1 H0 kitchen lamp, H0 TV, H1 bedside lamp, H2 NAS, H3 computer s1, H3 coffee M s1, H4 kitchen TV, H5 LCD TV s1, H5 LCD TV s2, H5 fridge freezer s1, H7 kitchen TV s1, H7 laptop scanner printer s2 H0 radio s1, H0 radio s2, H1 radio s1, H1 hair dryer, H4 hood s1, H5 lamp s1, H5 computer scanner printer s1, H5 computer scanner printer s2, H7 TV w decoder s1, H7 hood s1, H7 ADSL modem s1, H7 ADSL modem s2, H7 ADSL modem s3, H7 laptop scanner printer s1 H0 fridge freezer s1, H0 DW s1, H2 Bread M, H3 fridge freezer, H3 computer s2, H3 TV, H7 fridge freezer s1, H7 kitchen TV s2, H7 freezer H1 fridge, H1 DW s1, H2 DW s1, H2 notebook, H3 DW s1, H4 living R TV, H4 computer scanner printer s1 H0 WM s1, H0 kettle s1, H1 microwave s1, H1 WM s1, H2 WM s1, H2 TD s1, H2 coffee M, H3 WM s1, H4 fridge freezer, H4 E oven s1, H4 WM s1, H5 WM s1 H7 fridge freezer s2 H0 fridge freezer s2, H5 plasma TV s2, H5 lamp s2, H5 fridge freezer s2, H7 hood s2 H0 coffee M
2	H0 WM s2, H0 kettle s2, H1 DW s2, H3 DW s2, H3 WM s2, H3 hair dryer s3, H5 iron s4, H5 WM s2
3	H0 DW s2, H1 WM s2, H2 WM s2, H2 DW s2, H3 kettle, H4 WM s3
4	H2 TD s2
5	H3 hair dryer s1, H3 coffee M s2
6	H1 kettle, H4 computer scanner printer s2, H4 WM s2, H5 toaster s1, H5 stove s1, H5 iron s1, H5 iron s2, H7 fridge freezer s3
7	H1 bedside lamp, H4 E oven s2, H5 toaster s2, H7 E oven
8	H0 vacuum cleaner, H1 microwave s2, H3 Entrance outlet, H5 stove s2, H5 iron s3
9	H3 hair dryer s2, H3 coffee M s3, H4 computer scanner printer s3

Table B.18: Sub-groups of Group 1 based on Mean-shift clustering method for GREEND dataset. H denotes House number. DW=dishwasher. WM=washing machine. TD=tumble dryer. s denotes state number.

Group Number	*Appliance*
1.1	H2 TV H4 hood s2, H5 plasma TV s1 H0 kitchen lamp, H0 TV, H1 bedside lamp, H2 NAS, H3 computer s1, H3 coffee M s1, H4 kitchen TV, H5 LCD TV s1, H5 LCD TV s2, H5 fridge freezer s1, H7 kitchen TV s1, H7 laptop scanner printer s2
1.2	H0 radio s1, H0 radio s2, H1 radio s1, H1 hair dryer, H4 hood s1, H5 lamp s1, H5 computer scanner printer s1, H5 computer scanner printer s2, H7 TV w decoder s1, H7 hood s1, H7 ADSL modem s1, H7 ADSL modem s2, H7 ADSL modem s3, H7 laptop scanner printer s1 H0 fridge freezer s1, H0 DW s1, H2 Bread M, H3 fridge freezer, H3 computer s2, H3 TV, H7 fridge freezer s1, H7 kitchen TV s2, H7 freezer H1 fridge, H1 DW s1, H2 DW s1, H2 notebook, H3 DW s1, H4 living R TV, H4 computer scanner printer s1
1.3	H0 WM s1, H0 kettle s1, H1 microwave s1, H1 WM s1, H2 WM s1, H2 TD s1, H2 coffee M, H3 WM s1, H4 fridge freezer, H4 E oven s1, H4 WM s1, H5 WM s1
1.4	H7 fridge freezer s2
1.5	H0 fridge freezer s2, H5 plasma TV s2, H5 lamp s2, H5 fridge freezer s2, H7 hood s2
1.6	H0 coffee M

Table B.19: Sub-groups of Group 1.2 based on Mean-shift clustering method for GREEND dataset. H denotes House number. DW=dishwasher. WM=washing machine. TD=tumble dryer. s denotes state number.

Group Number	*Appliance*
1.2.1	H4 hood s2, H5 plasma TV s1
1.2.2	H0 kitchen lamp, H0 TV, H1 bedside lamp, H2 NAS, H3 computer s1, H3 coffee M s1, H4 kitchen TV, H5 LCD TV s1, H5 LCD TV s2, H5 fridge freezer s1, H7 kitchen TV s1, H7 laptop scanner printer s2
1.2.3	H0 radio s1, H0 radio s2, H1 radio s1, H1 hair dryer, H4 hood s1, H5 lamp s1, H5 computer scanner printer s1, H5 computer scanner printer s2, H7 TV w decoder s1, H7 hood s1, H7 ADSL modem s1, H7 ADSL modem s2, H7 ADSL modem s3, H7 laptop scanner printer s1
1.2.4	H0 fridge freezer s1, H0 DW s1, H2 Bread M, H3 fridge freezer, H3 computer s2, H3 TV, H7 fridge freezer s1, H7 kitchen TV s2, H7 freezer
1.2.5	H1 fridge, H1 DW s1, H2 DW s1, H2 notebook, H3 DW s1, H4 living R TV, H4 computer scanner printer s1

Table B.20: Groups and Sub-groups of GREEND dataset based on tree clustering method. H denotes House number. DW=dishwasher. WM=washing machine. TD=tumble dryer. s denotes state.

Group Number		*Appliance*
1		H1 kettle, H2 TV, H3 hair dryer s1, H3 coffee M s2, H4 computer scanner printer s2, H5 toaster s1, H5 stove s1, H5 iron s1
2		H7 fridge freezer s2, H7 fridge freezer s3, H5 iron s2, H4 WM s2
	3.1	H1 microwave s1, H1 WM s1, H2 WM s1, H4 WM s1
	3.2	H0 kettle s1, H2 coffee M s1, H4 E oven s1
	3.3	H0 kitchen lamp, H2 notebook, H4 kitchen TV, H4 computer scanner printer s1, H5 LCD TV s1, H7 kitchen TV s1, H7 laptop scanner printer s2
	3.4	H0 radio s1, H0 radio s2, H1 radio s1, H1 hair dryer, H4 hood s1, H5 lamp s1, H5 computer scanner printer s1, H5 computer scanner printer s2, H7 TV with decoder s1, H7 hood s1, H7 ADSL modem s1, H7 ADSL modem2, H7 ADSL modem s3, H7 laptop scanner printer s1
3	*3.5*	H1 DW s1, H2 DW s1, H4 living R TV s1
	3.6	H1 fridge, H3 DW s1, H5 WM s1
	3.7	H0 fridge freezer s1, H0 DW s1, H3 computer s2, H3 TV, H7 kitchen TV s2
	3.8	H7 fridge freezer s1
	3.9	H0 TV, H1 radio s2, H2 NAS, H3 computer s1
	3.10	H3 coffee M s1, H5 LCD TV s2, H5 fridge freezer s1
4		H0 coffee M, H0 WM s1, H0 fridge freezer s2, H2 TD s1, H2 bread M, H3 fridge freezer, H3 WM s1, H4 fridge freezer,H4 hood s2,H5 plasma TV s1,H5 plasma TV s2, H5 lamp s2, H5 fridge freezer s2, H7 hood s2, H7 freezer
5		H0 vacuum cleaner, H1 microwave s2, H3 entrance outlet, H3 hair dryer s2, H3 coffee M s3, H4 computer scanner printer s3, H5 stove s2, H5 iron s3
6		H1 bedside lamp, H4 E oven s2, H5 toaster s2, H7 E oven
7		H0 WM s2, H0 kettle s2, H0 DW s2, H1 DW s2, H1 WM s2,H2 WM s2, H2 DW s2, H3 DW s2, H3 kettle,H3 WM s2, H3 hair dryer s3, H4 WM s3, H5 iron s4, H5 WM s2
8		H2 TD s2

Table B.21: Groups of REFIT dataset House 2 based on Mean-shift clustering method. DW=dishwasher. WM=washing machine. TV=television. s denotes state number.

Group Number	*Appliance*
1	WM s2, DW s2, toaster s5, kettle s2
2	microwave s1, kettle s1, TV s1, TV s2, fridge/freezer s1, WM s1, toaster s1, toaster s2, DW s1
3	microwave s2, toaster s3, toaster s4

Table B.22: Sub-groups for REFIT dataset House 2 of group 2 based on Mean-shift clustering method. DW=dishwasher. WM=washing machine. TV=television. s denotes state number.

Group Number	*Appliance*
2.1	microwave s1
2.2	TV s1, TV s2, fridge/freezer s1, WM s1, toaster s1, toaster s2, DW s1
2.3	kettle s1

Table B.23: Sub-groups for REFIT dataset House 2 of group 2.2 based on Mean-shift clustering method. DW=dishwasher. WM=washing machine. TV=television. s denotes state number.

Group Number	*Appliance*
2.2.1	TV s2
2.2.2	DW s1, TV s1
2.2.3	fridge/freezer s1, WM s1, toaster s1, toaster s2

Table B.24: Groups for REFIT dataset House 3 based on Mean-shift clustering method. DW=dishwasher. WM=washing machine. TV=television. TD=tumble dryer. s denotes state number.

Group Number	*Appliance*
1	freezer s1, DW s1, WM s1, TV s 1 TD s1, fridge/freezer s1
2	TD s2, DW s2, WM s2, kettle s1
3	toaster s1, freezer s2, microwave s1

Table B.25: Sub-groups for group 1 REFIT House 3 dataset based on Mean-shift clustering method. DW=dishwasher. WM=washing machine. TV=television. TD=tumble dryer. s denotes state number.

Group Number	*Appliance*
1.1	freezer s1, DW s1, WM s1, TV s 1
1.2	TD s1
1.3	fridge/freezer s1

Table B.26: Groups of REFIT House 2 dataset based on Tree clustering method. DW=dishwasher. WM=washing machine. s denotes state number.

Group Number			*Appliance*
1			kettle s2
2			DW s2, WM s2, toaster s5
	3.1		microwave s1, toaster s2, kettle s1
3	*3.2*	*3.2.1*	fridge/freezer s1, toaster s1
	3.2	*3.2.2*	TV s1, TV s2, DW s1
	3.3		WM s1
4			microwave s2, toaster s3, toaster s4

Table B.27: Groups of REFIT House 3 dataset based on Tree clustering method. DW=dishwasher. WM=washing machine. s denotes state number.

Group Number		*Appliance*
1		TD s2, WM s2, DW s2, kettle s1
2		freezer s2, microwave s1
3	*3.1*	TD s1, WM s1
	3.2	DW s1, fridge-freezer s1, TV s1
4		toaster s1, freezer s1

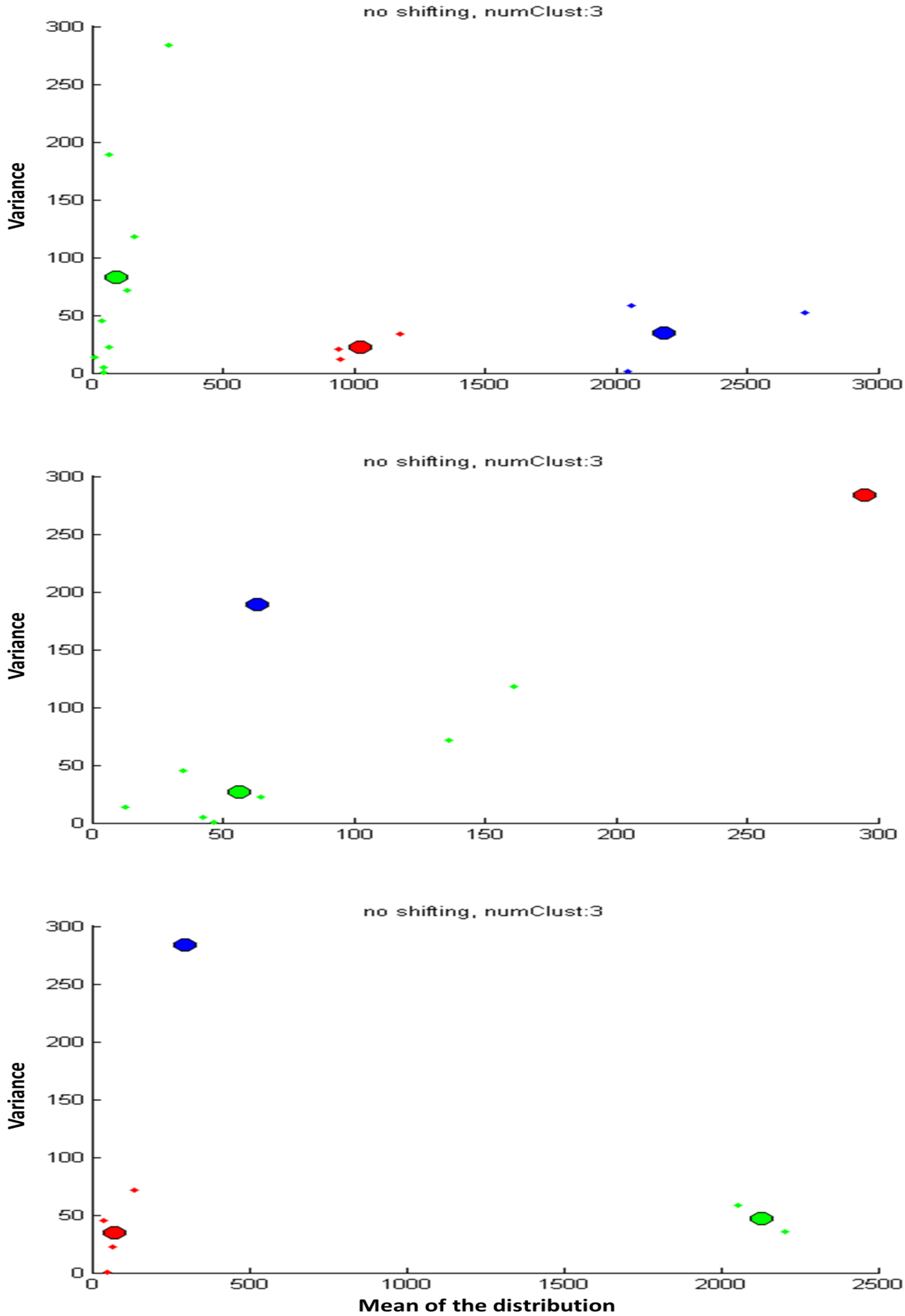


Figure B.24: Clustering groups for REFIT dataset House 2 appliances into groups and sub-groups using mean-shift algorithm. Upper figure has 3 groups. Middle figure has 3 sub-groups from group 2. Lower figure has 3 sub-groups from sub-group 2.2. x-axes is Power in [W]

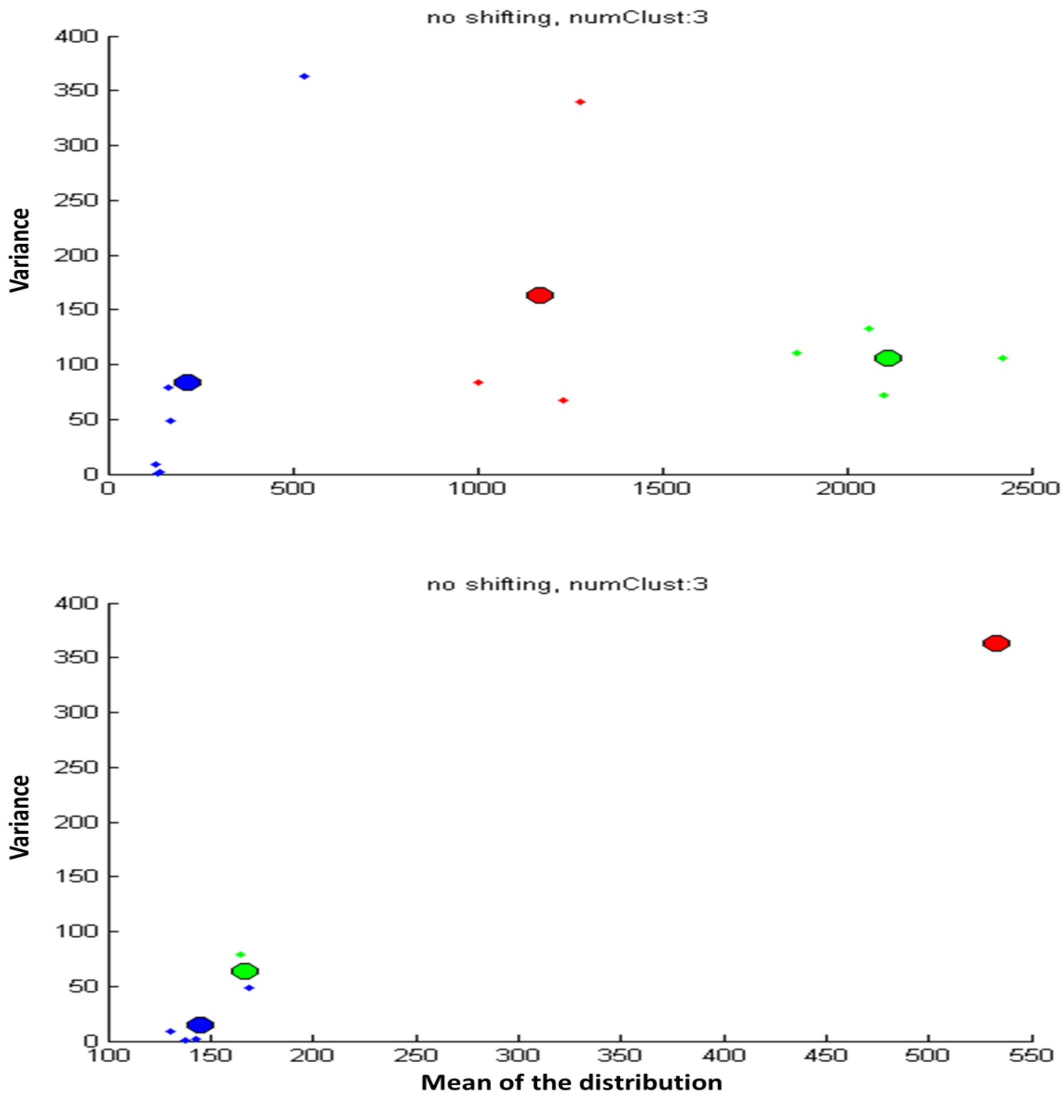


Figure B.25: Clustering groups for REFIT dataset House 3 into groups and sub-groups using mean-shift algorithm. Upper figure has 3 groups. Lower figure has 3 sub-groups from group 1.

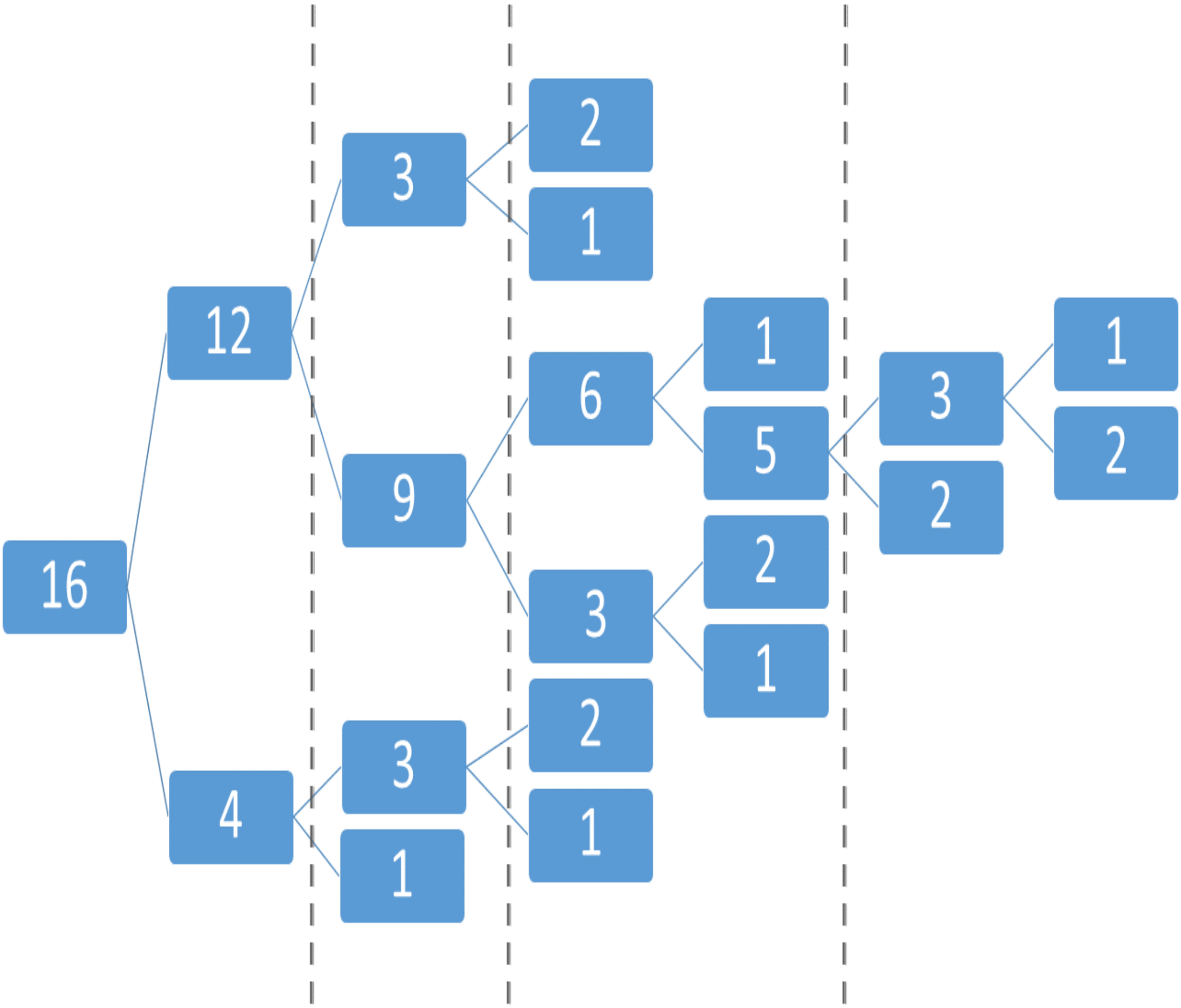


Figure B.26: Classification for REFIT dataset House 2 using Tree clustering. Nodes show number of appliances to be clustered. Dashed line is a suitable level to cut the tree.

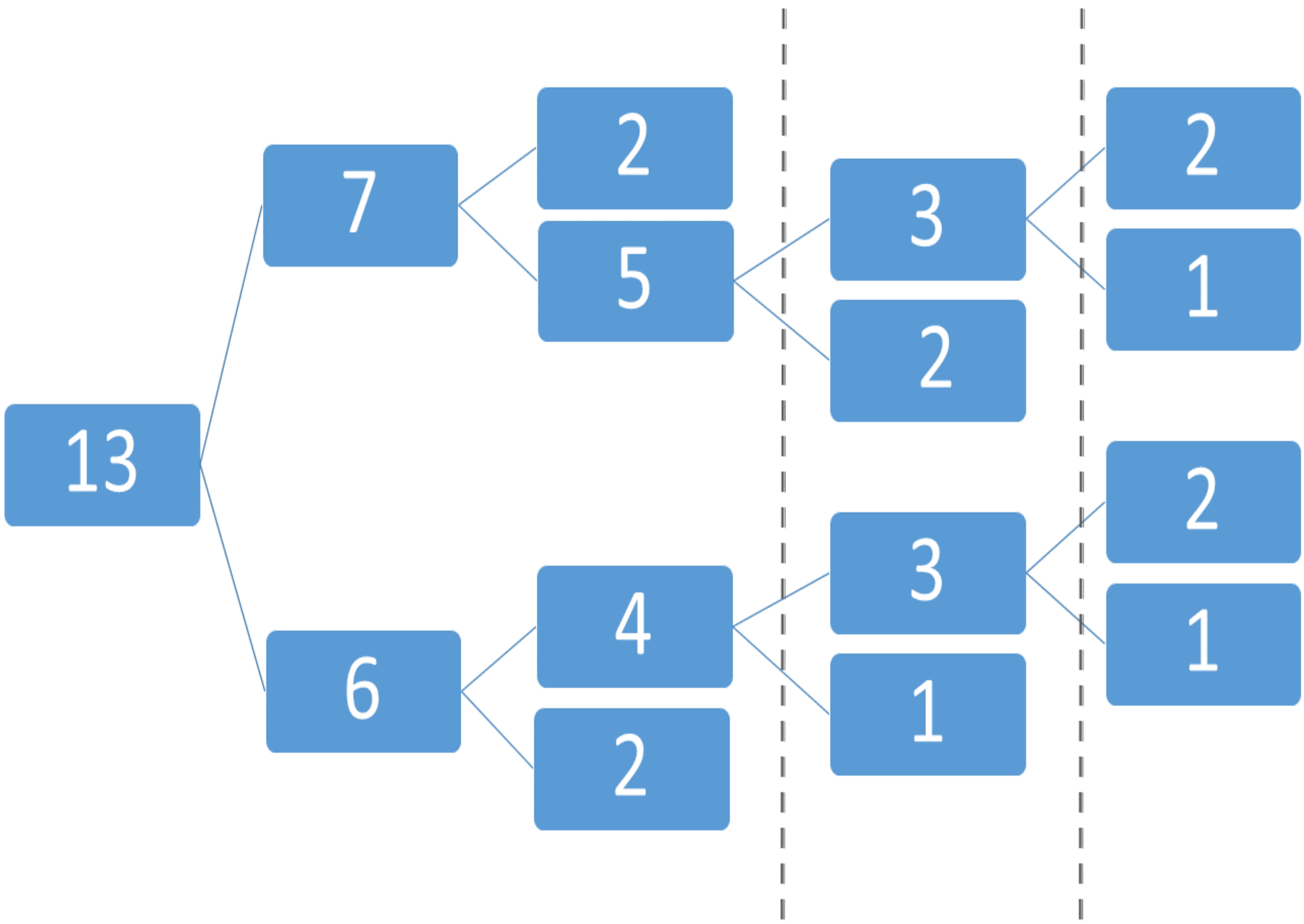


Figure B.27: Classification for REFIT dataset House 3 using Tree clustering. Nodes show number of appliances to be clustered. Dashed line is a suitable level to cut the tree.

washing machine first states. Group 3 contains one appliance state only which is washing machine first state. Group 4 shows a similarity between microwave second state, toaster third and fourth states.

Figure B.27 shows groups of REFIT House 3 appliances after clustering using Tree method. It can be seen that 13 appliance-states are present in House 3 forming a tree of appliance-signatures. The tree was cut into suitable hight forming 4 main groups of 2, 5, 4 and 2 appliance state. The group with 5 appliance-states was then re-cut into further groups, forming two groups. Table B.27 shows all groups and sub-groups from House 3 tree clustering; it can be seen that Group 1 contains medium consumption values using Gaussian estimation curve such as tumble dryer, washing machine, dishwasher and kettle second states. Group 2 shows a connection between freezer and microwave second states.

Group 3 was re-cut into further two sub-groups containing appliances with low states values. Group 3.1 shows similarity tumble dryer and washing machine first states. Group 3.2 contains dishwasher first state, freezer second state and television first state. Group 4 shows similarity between taster and freezer first states.

Bibliography

[1] EU Climate Change Organisation, `http://ec.europa.eu/clima/policies/strategies/2020_en`, [accessed: DEC 2016]

[2] G. R. Barai, S. Krishnan and B. Venkatesh, "Smart metering and functionalities of smart meters in smart grid - a review", in *the Proceedings of the IEEE Electrical Power and Energy Conference (EPEC)*, pp. 138—145, October 2015.

[3] The U.S. Energy Information Administration (EIA), `http://www.eia.gov/`, [accessed: Oct 2016]

[4] M. E. Berges, E. Goldman, H. Matthews, and L. Soibelman, "Enhancing electricity audits in residential buildings with non-intrusive load monitoring," in *Journal of industrial ecology*, vol. 14, pp. 844–858, 2010.

[5] S. Das, S. Srikrishna, A. Shukla, G. Harsha, and S. Deb, "A low cost non-intrusive appliance load monitoring system," in *Advance Computing Conference (IACC)*, IEEE 3rd International, pp. 1641-1644, 2013.

[6] Y. H. Lin and M. S. Tsai, "The Integration of a Genetic Programming-Based Feature Optimizer With Fisher Criterion and Pattern Recognition Techniques to Non-Intrusive Load Monitoring for Load Identification," in *International Journal of Green Energy*, vol. 12, pp. 279–290, 2015.

[7] P. Yeung, and M. Jung, "Improving Electric Reliability with Smart Meters", White Paper, in *Silver Spring Networks*, 2013.

[8] J. Zheng, D. W. Gao, and L. Lin, "Smart Meters in Smart Grid: An Overview", in *IEEE Green Technologies Conference*, Denver, Colorado, April 2013.

[9] US Department of Energy, "The smart grid: An introduction. How a smarter grid works as an enabling engine for our economy, our environment and our

future", Report prepared for the U.S. Department of Energy by Litos Strategic Communication under contract No. DEAC26-04NT41817.

[10] P. Ramila, and H. Rudnik, 'Assessment of the Introduction of Smart Metering in a Developing Country'. Available: `http://web.ing.puc.cl/~power/paperspdf/RamilaRudnick.pdf`

[11] Government response to the consultation on the second version of the Smart Metering Equipment Technical Specifications: Part 2, "*Department of Energy & Climate Change UK*, December 2013.

[12] Reuters News Website, `http://www.reuters.com/article/energy-efficiency-smartmeters-italy-idUSL5N0EA3HL20130530` , [accessed: Nov 2016]

[13] Victoria state government smart meter project, `http://www.smartmeters.vic.gov.au/`, [accessed: Nov 2016]

[14] Navigant Research website, `https://www.navigantresearch.com/blog/middle-east-set-to-embrace-the-smart-meter-revolution`, [accessed: Nov 2016]

[15] Metering & Smart Energy International website, `https://www.metering.com/middle-east-smart-meters-qatar-to-finish-phase-one-in-2016/`, [accessed: Nov 2016]

[16] K. C. Armel, A. Gupta, G. Shrimali, and A. Albert, "Is disaggregation the holy grail of energy efficiency ? The case of electricity", *Energy Policy*, vol. 52, pp. 213–234, 2013.

[17] M. N. Valero Pérez, "A non-intrusive appliance load monitoring system for identifying kitchen activities," Department of Automation and System Technology, Aalto University - School of Electrical Engineering, Espoo, Finland, 2011.

[18] N. Batra, H. Dutta, and A. Singh, "INDiC: Improved Non-intrusive Load Monitoring Using Load Division and Calibration," in *Machine Learning and Applications (ICMLA)*, 12th International Conference ,pp 79–84, 2013.

[19] G.Hart, "Nonintrusive Appliance Load Data Acquisition Method", *MIT Energy Laboratory Technical Report*, Sept. 1984.

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

[21] D. Benyoucef, P. Klein, and T. Bier, "Smart Meter with nonintrusive load monitoring for use in Smart Homes", in *IEEE Energy Conference and Exhibition (EnergyCon)*, pp. 96-101, 2010.

[22] D. L. Racines and J. E. Candelo, "Non Intrusive Load Identification with Power and Impedance obtained from Smart Meters", in *International Journal of Engineering and Technology*, vol. 6, pp. 0975–4024, 2014.

[23] K. Basu, V. Debusschere and S. Bacha, "Residential appliance identification and future usage prediction from smart meter", in *9th Annual Conference of the IEEE Industrial Electronics Society (IECON)*, vol. 3, pp. 4994–4999, 2013.

[24] D. Ming, P. C. M. Meira, X. Wilsun, and C. Y. Chung, "Non-Intrusive Signature Extraction for Major Residential Loads", in *Smart Grid, IEEE Transactions*, vol. 4, pp. 1421–1430, 2013.

[25] A. Zoha, A. Gluhak, M.A. Imran, and S. Rajasegarar, "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey," *Sensors,* vol. 12, pp. 16838–16866, Dec. 2012.

[26] K. L. Lian, K. S. Tung, and Y. C. Su, "A non-intrusive load monitoring system based on a cascaded method," in Electric Power and Energy Conversion Systems (EPECS), 2013 3rd International Conference on, 2013, pp. 1-6.

[27] C. Hsueh-Hsien, C. Kun-Long, T. Yuan-Pin, and L. Wei-Jen, "A new measurement method for power signatures of non-intrusive demand monitoring and load identification", in *IEEE Industry Applications Society Annual Meeting (IAS)*, pp. 1–7, 2011.

[28] A. I. Cole and A. Albicki, "Algorithm for non-intrusive identification of residential appliances", in *IEEE Circuits and Systems, (ISCAS)*, pp. 338–341.

[29] W. Xiaojing, L. Dongmei, Y. Jing, Z. Liqiang, and S. West, "An online load identification algorithm for non-intrusive load monitoring in homes", in *IEEE Intelligent Sensors, Sensor Networks and Information Processing*, pp. 1–6, 2013.

[30] C. Hsueh-Hsien, L. Kuo-Lung, S. Yi-Ching, and L. Wei-Jen, "Energy spectrum-based wavelet transform for non-intrusive demand monitoring and load identification", in *IEEE Industry Applications Society Annual Meeting*, pp. 1-9, 2013.

[31] A. Suzdalenko and I. Galkin, "Case study on using non-intrusive load monitoring system with renewable energy sources in intelligent grid applications", in *Compatibility and Power Electronics (CPE), 2013 8th International Conference*, pp. 115–119, 2013.

[32] K. X. Perez, W. J. Cole, J. D. Rhodes, A. Ondeck, M. Webber, M. Baldea, et al., "Non-intrusive disaggregation of residential airconditioning loads from sub-hourly smart meter data", Energy and Buildings, vol. 81, pp. 316-325, 2014.

[33] M. Mathis, A. Rumsch, R. Kistler, A. Andrushevich, and A. Klapproth, "Improving the Recognition Performance of NIALM Algorithms through Technical Labeling", in *IEEE 12th Embedded and Ubiquitous Computing International Conference (EUC)*, pp. 227–233, 2014.

[34] D. C. Bergman, J. Dong, J. P. Juen, N. Tanaka, C. A. Gunter, and A. K. Wright, "Distributed non-intrusive load monitoring," in *IEEE Innovative Smart Grid Technologies (ISGT)*, pp. 1–8, 2011.

[35] J. Lei, L. Suhuai, and L. Jiaming, "Automatic power load event detection and appliance classification based on power harmonic features in non-intrusive appliance load monitoring", in *IEEE 8th Industrial Electronics and Applications (ICIEA)*, pp. 1083–1088, 2013.

[36] W. Yung Fei, Y. Ahmet Sekercioglu, T. Drummond, and W. Voon Siong, "Recent approaches to non-intrusive load monitoring techniques in residential settings," in *Computational Intelligence Applications In Smart Grid (CIASG)*, IEEE Symposium on, pp. 73–79, 2013,.

[37] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook", *IEEE Trans. Consumer Electronics*, vol. 57, no. 1, pp. 76–84, Feb. 2011.

[38] S. Markonin, I.V. Bajic, and F. Popowich, "Efficient sparse matric processing for nonintrusive load monitoring", *2nd Int. Non-intrusive Appliance Load Monitoring Workshop,* Austin, TX, June 2014.

[39] J. Kolter, and M. Johnson, "REDD: A public data set for energy disaggregation research", *Workshop on Data Mining Applications in Sustainability (SIGKDD)*, San Diego, CA, 2011.

[40] A. Monacchi, D. Egarter, W. Elmenreich, S. D'Alessandro and A. M. Tonello, "GREEND: An Energy Consumption Dataset of Households in Italy and Austria", in *Proc. Smart Grid Comm*, IEEE, 2014.

[41] D. Murray, J. Liao, L. Stankovic, V. Stankovic, R. Hauxwell-Baldwin, C. Wilson, M. Coleman, T. Kane, and S. Firth, "A data management platform for personalised real-time energy feedback", in *Proc. EEDAL-2015 8th Int. Conf. Energy Efficiency in Domestic Appliances and Lighting,* Lucerne-Horw, Switzerland, Aug. 2015.

[42] C. Cortes and V. Vapnik, "Support-vector networks". Machine Learning 20 (3): 273. doi:10.1007/BF00994018, 1995.

[43] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Non-intrusive load monitoring using prior models of general appliance types," in *Proc. the 26th Conf. Artificial Intelligence (AAAI-12)*, Toronto, CA, pp. 356–362, July 2012.

[44] M.E. Berges, E. Goldman, H.S. Matthews, L. Soibelman, "Enhancing electricity audits in residential buildings with nonintrusive load monitoring". J. Ind. Ecol. vol(14), pp 844–858, 2010.

[45] K. D. Anderson, M. E. Bergés, A. Ocneanu, D. Benitez and J. M. F. Moura, "Event detection for Non Intrusive load monitoring,", in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, Montreal, QC, 2012, pp. 3312-3317.

[46] R. Bonfigli, S. Squartini, M. Fagiani and F. Piazza, "Unsupervised algorithms for non-intrusive load monitoring: An up-to-date overview," Environment and Electrical Engineering (EEEIC)", in *IEEE 15th International Conference*, Rome, pp. 1175-1180, 2015.

[47] S. Makonin, "Approaches to Non-Intrusive Load Monitoring (NILM) in the Home", Technical Report, Simon Fraser University, 2012.

[48] M. Figueiredo, A. de Almeida & B. Ribeiro, "An experimental study on electrical signature identification of non-intrusive load monitoring (nilm) systems". Adaptive and Natural Computing Algorithms, pp. 31-–40, 2011.

[49] M. Figueiredo, A. de Almeida & B. Ribeiro. "Home electrical signal disaggregation for non-intrusive load monitoring (nilm) systems". Neurocomputing(0), 2012.

[50] S. Gupta, M. Reynolds & 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*, pp.139–148, 2010.

[51] M. Berenguer, M. Giordani, F. Giraud-By & N. Noury, "Automatic detection of activities of daily living from detecting and classifying electrical events on the residential power line. Ine-Health networking, applications and services", Healthcom 2008. in *10th international conference*, pp.29—32, 2008.

[52] M. Tsai, & Y. Lin, "Modern development of an adaptive non-intrusive appliance load monitoring system in electricity energy conservation". in *Applied Energy*, pp. 55–73, 2012.

[53] L.K. Norford, & S.B. Leeb, "Non-intrusive electrical load monitoringin commercial buildings based on steady-state and transient load-detection algorithms", *Energy and Buildings*, Issue 24, Vol.1, pp.51–64, 1996.

[54] C. Laughman, K. Lee, R. Cox, S. Shaw, S. Leeb, L. Norford, and P. Armstrong, "Power signature analysis", *IEEE Power and Energy Magazine*, vol. 1, pp 56–63, Mar-Apr 2003.

[55] R. Fisera & K. Macek, "Virtual sub-metering via combined classifiers". In *Intelligent data acquisition and advanced computing systems (idaacs)*, vol. 1, p.126–131, 2011.

[56] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han, "Unsupervised disaggregation of low frequency power measurements", in *Proc. 11th SIAM Int. Conf. Data Mining,* Mesa, AZ, April 2010.

[57] M. Zeifman, "Disaggregation of home energy display data using probabilistic approach". *Consumer Electronics, IEEE Transactionson*, Issue 58, vol. 1, pp. 23–31, 2012.

[58] J. Kolter, S. Batra & A. Ng, "Energy disaggregation via discriminative sparse coding". *Proc. Neural Information Processing Systems*, 2010.

[59] J. Kolter, and T. Jaakkola, "Approximate inference in additive factorial HMMs with application to energy disaggregation", in *J. Machine Learning*, vol. 22, pp. 1472–1482, 2012.

[60] H. Chang, C. Lin & J. K. Lee, "Load identification in nonintrusive load monitoring using steady-state and turn-on transient energy algorithms". In *Computer supported cooperative work in design (cscwd)*, in *14th international conference*, pp. 27–32, 2010.

[61] S. Patel, T. Robertson, J. Kientz, M. Reynolds & 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*, pp.271—288, 2007.

[62] J. Liang, S. K. K. Ng, G. Kendall, J. W. M. Cheng, "Load signature study part I: Basic concept, structure, and methodology", *IEEE Trans. Power Delivery*, vol. 25, no. 2, pp. 551–560, Apr. 2010.

[63] J. Froehlich, E. Larson, S. Gupta, G. Cohn, M. Reynolds & S. Patel, "Disaggregated end-use energy sensing for the smart grid", in *IEEE Pervasive Computing*, Issue. 10, vol. 1,pp. 28–39. 2011.

[64] S. Intille, K. Larson, E. Tapia, J. Beaudin, P. Kaushik, J. Nawyn & R. Rockinson, "Using a live-in laboratory for ubiquitous computing research". *Pervasive Computing*, vol. 4, pp. 349—365, 2006.

[65] E. Tapia, S. Intille & K. Larson, "Activity recognition in the home using simple and ubiquitous sensors". *Pervasive Computing* , pp. 158-–175, 2004.

[66] M. Baranski and J. Voss, "Genetic algorithm for pattern detection in NIALM systems", in *Proc. SMC-2004 IEEE Int. Conf. Sys., Man and Cybernetics,* pp. 3462-3468, Hague, Netherlands, Oct. 2004.

[67] S. Leeb, S. Shaw & J. Kirtley, "Transient event detection in spectral envelope estimates for non-intrusive load monitoring". *IEEE Power Delivery Transactionson*, Issue. 10, vol. 3, pp. 1200–1210, 1995.

[68] D. Luo, L. Norford, S. Leeb & S. Shaw, "Monitoring hvac equipment electrical loads from a centralized location- methods and field test results". *ASHRAE Transactions*. Issue. 108, vol. 1, pp. 841-–857, 2002.

[69] M. Figueiredo, A. de Almeida, B. Ribeiro & A. Martins, "Extracting features from an electrical signal of a non-intrusive load monitoring system". *Intelligent Data Engineering and Automated Learning (IDEAL)*, pp. 210—217, 2010.

[70] W. Chan, A. So & L. Lai, "Harmonics load signature recognition by wavelets transforms". *Electric utility deregulation and restructuring and power technologies (drpt)*, pp. 666—671, 2000.

[71] N. Tse, L. Zhou & L. Lai, "Wavelet-based algorithm for nonstationary power system, waveform analysis". *Wavelet analysis and pattern recognition (icwapr)*, vol. 2, pp. 729–735, 2008.

[72] M. E. Berges, E. Goldman, H. Matthews & L. Soibelman, "Learning systems for electric consumption of buildings". *Asci International Workshop on Computing in Civil Engineering*, 2009.

[73] M. Baranski & J. Voss, "Non-intrusive appliance load monitoring based on an optical sensor". *IEEE Power tech conference proceedings*, vol. 4, Bologna, Italy, 2003.

[74] M. Baranski & J. Voss, "Detecting patterns of appliances from total load data using a dynamic programming approach". *IEEE Data mining fourth international conference (icdm)*, pp. 327–330, 2004a.

[75] K. Suzuki, S. Inagaki, T. Suzuki, H. Nakamura, K. Ito, "Non-intrusive Appliance Load Monitoring Based on Integer Programming". In *SICE Annual Conference*, vol. 174, pp. 2742—2747. Tokyo, Japan, August 2008.

[76] E. Schoneburg, F. Heinzmann, S. Feddersen, "Genetische Algoritbmen und Evolutionsstrategien", Addison-Wesley, New York, 1994.

[77] A. Marchiori, D. Hakkarinen, Q. Han, L. Earle, "Circuit-level load monitoring for household energy management", *IEEE Pervas. Comput.* vol. 10, pp. 40–48. 2011.

[78] M. Sanquer, F. Chatelain, M. El-guedri, N Martin, "Hierarchical bayesian learning for electrical transient classification", in *The 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Canada, 2013.

[79] A. G. Ruzzelli, C. Nicolas, A. Schoofs, and G.M.P. O'Hare, "Real-time recognition and profiling of appliances through a single electricity sensor", *IEEE SECON-2010 7th Annual Conf. Sensor Mesh and Ad Hoc Communications and Networks* , pp. 1–9, Boston, June 2010.

[80] J. Kelly, W. Knottenbelt, "Neural NILM: Deep Neural Networks Applied to Energy Disaggregation", in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, pp. 55–64 ,2015.

[81] M. Nguyen , S. Alshareef , A. Gilani , and W. G. Morsi , "A novel feature extraction and classification algorithm based on power components using single-point monitoring for NILM", presented at *the 28th annual IEEE Canadian Conference on Electrical and Computer Engineering*, Halifax, NS, pp. 37—40, Canada , 2015.

[82] S. Alshareef and W. G. Morsi, "Application of wavelet-based ensemble tree classifier for non-intrusive load monitoring", presented at *the IEEE Electrical Power & Energy Conference (EPEC 2015)*, pp. 397 — 401, London, ON, Canada , 2015.

[83] K. He, L. Stankovic, J. Liao & V. Stankovic, "Non-intrusive load disaggregation using graph signal processing". *IEEE Transactions on Smart Grid*, in press 2017.

[84] K. M. Rao, D. Ravichandran, and K. Mahesh, "Non-Intrusive Load Monitoring and Analytics", in *Device Prediction Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol I, March, Hong Kong, 2016.

[85] M. Singh, S. K. Semwal, R. S. Prasad, "Residential Load Signature Analysis for Their Segregation Using Wavelet—SVM", *Power Electronics and Renewable Energy Systems*, pp. 863–871.

[86] D. Srinivasan, W. Ng, A. Liew, "Neural-network-based signature recognition for harmonic source identification". *IEEE Trans. Power Del.*, issue. 21, pp. 398-–405,, 2006.

[87] T. Kato, H.S. Cho, D. Lee, "Appliance Recognition from Electric Current Signals for Information-Energy Integrated Network in Home Environments". In *7th International Conference on Smart Homes and Health Telematics*, Volume 5597, Tours, France, pp. 150-–157, July 2009.

[88] Y.X. Lai, C.F. Lai, Y.M. Huang, H.C. Chao, "Multi-appliance recognition system with hybrid SVM/GMM classifier in ubiquitous smart home". In *Inform. Sci*, 2012.

[89] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han, "Unsupervised disaggregation of low frequency power measurements", in *Proc. 11th SIAM Int. Conf. Data Mining,* Mesa, AZ, April 2011.

[90] S. Pattem, "Unsupervised disaggregation for non-intrusive load monitoring", in *IEEE Machine Learning and Applications (ICMLA), 11th International Conference*, vol. 2, pp. 515—520, 2012.

[91] M. J. Johnson and A. S. Willsky, "Bayesian non-parametric Hidden Semi-Markov Models", *J. Machine Learning Research (JMLR)*, vol. 14, pp. 673–701, Feb. 2013.

[92] B. Zhao, L. Stankovic, and V. Stankovic, "On a training-less solution for non-intrusive appliance load monitoring using graph signal processing", *IEEE Access*, vol. 4, pp. 1784–1799, 2016.

[93] M. Wytock and J. Z. Kolter, "Contextually supervised source separation with application to energy disaggregation", *ArXiv e-prints*, [arXiv:1312.5023], 2013.

[94] H. Goncalves, A. Ocneanu, and M. Berges, "Unsupervised disaggregation of appliances using aggregated consumption data", in *1st KDD Workshop on Data Mining Applications in Sustainability (SustKDD)*, Aug 2011.

[95] L. Jing, G. Elafoudi, L. Stankovic and V. Stankovic, "Power disaggregation for low-sampling rate data", in *the 2nd International Workshop on Non-Intrusive Load Monitoring*, 2014.

[96] G. Elafoudi,L. Stankovic and V. Stankovic, "Power disaggregation of domestic smart meter readings using dynamic time warping". *IEEE 6th International Symposium on Communications, Control and Signal Processing (ISCCSP)*, 2014.

[97] C. Metz, "Basic principles of roc analysis". In *Seminars in nuclear medicine*, vol. 8, pp. 283—298, 1978.

[98] M. Sokolova, N. Japkowicz & S. Szpakowicz, "Beyond accuracy, f-score and roc: A family of discriminant measures for performance evaluation". In *A. Sattar & B.-h. Kang (Eds.), Ai 2006: Advances in artificial intelligence*, vol. 4304, pp. 1015–1021. Springer Berlin / Heidelberg, 2006.

[99] NILM wiki website, `http://wiki.nilm.eu/index.php?title=NILM_datasets` [accessed: July 2016]

[100] D. Murray, L. Stankovic, V. Stankovic. "An electrical load measurements dataset of United Kingdom households from a two-year longitudinal study". *Scientific Data*, 2016.

[101] J. Wang, X. Wu, and C. Zhang, "Support vector machines based on k-means clustering for real-time business intelligence systems", *Int. J. Business Intelligence and Data Mining*, vol. 1, no. 1, pp 54-64,2005.

[102] Y. Yao, Y. Liu, Y. Yu, H. Xu, W. Lv, Z. Li, and X. Chen, "K-SVM: An Effective SVM Algorithm Based on K-means Clustering", *J. Computers,* vol. 8, Oct. 2013.

[103] Q. Gu and J. Jan, "Clustered Support Vector Machines", in *Proc. AISTATS-2013 16th Int. Conf. Artifical Intelligenece and Statistics,* Scottsdale, AZ, 2013.

[104] X.L. Xia, M.R. Lyu, T.M. Lok, and G.B. Huang, "Methods of decreasing the number of support vectors via k-mean clustering", in *ICIC 2005, LNCS 3644*, pp. 717–726, Spinger-Verlag Berlin Heidelberg, 2005.

[105] J. Liao, G. Elafoudi, L. Stankovic, and V. Stankovic, "Power disaggregation for low-sampling rate data", *2nd Int. Non-intrusive Appliance Load Monitoring Workshop,* Austin, TX, June 2014.

[106] H. Altrabalsi, J. Liao, L. Stankovic, and V. Stankovic, "A low-complexity energy disaggregation method: Performance and robustness", *SSCI-2014 IEEE Symposium on Computational Intelligence Applications in Smart Grid*, Orlando, FL, December 2014.

[107] M. Dong, P. C. Meira, W. Xu, and W. Freitas, "An Event Window Based Load Monitoring Technique for Smart Meters", in *IEEE Trans. Smart Grid 3* , vol. 2 , pp. 787–796 , 2012.

[108] J. MacQueen, "Some methods for classification and analysis of multivariate observations". In *Proceeding of the 5th Berkeley symposium 1*, pp. 281—297, 1967.

[109] X. Huang, A. Acero, H. w. hon, "Spoken Language Processing: A guide to Theory, Algorithm, and System Development", Prentice Hall, January 2001.

[110] N. Seo, Code: K-means clustering using euclid distance, `http://note.sonots.com/SciSoftware/kmeans.html#ma92e1c9`, [Acccessed: 10 October 2016]

[111] S.H. Al-Harbi and V.J. Rayward-Smith, "Adapting k-means for supervised clustering", *Appl. Intell.* vol. 24, pp. 219–226, 2006.

[112] S. T. Bow, "Pattern Recognition and Image Preprocessing", Marcel Dekker, 1992.

[113] D. Srinivasan, W. Ng,; A. Liew, "Neural-network-based signature recognition for harmonic source identification". IEEE Trans. *Power Del.* 2006, 21, 398–405.

[114] T. Kato, H.S. Cho, D. Lee, "Appliance Recognition from Electric Current Signals for Information-Energy Integrated Network in Home Environments". In *the 7th International Conference on Smart Homes and Health Telematics*, Tours, France, 1–3 July 2009; Volume 5597, pp. 150–157.

[115] G.y. Lin, S.C. Lee, Y.J. Hsu, W.R. Jih, "Applying Power Meters for Appliance Recognition on the Electric Panel". In *the 5th IEEE Conference on Industrial Electronics and Applications*, pp. 2254-–2259, Melbourne, Australia, June 2010.

[116] Vert, Jean-Philippe, K. Tsuda, and B. Schölkopf, "A primer on kernel methods", Kernel Methods in Computational Biology, 2004.

[117] J. Milgram, M. Cheriet, R. Sabourin, "One Against One" or "One Against All": Which One is Better for Handwriting Recognition with SVMs?", 2006.

[118] Hsu, Chih-Wei; and Lin, Chih-Jen, "A Comparison of Methods for Multiclass Support Vector Machines". *IEEE Transactions on Neural Networks*, 2002.

[119] K. B. Duan, S. S. Keerthi, "Which Is the Best Multiclass SVM Method? An Empirical Study". In *Multiple Classifier Systems .LNCS 3541*, pp. 278-–285. ISBN 978-3-540-26306-7, 2005.

[120] J. Gao, S. Giri, E. C. Kara, and M. Berges, "PLAID: a public dataset of high-resolution electrical appliance measurements for load identification research: demo abstract", in *Proc. the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, pp. 198–199, 2014.

[121] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis", *IEEE Trans. Pattern Anal. Machine Intell.*, 24:603–619, 2002.

[122] R. Duda, P. Hart and D. Stork, "Pattern Classification". Wiley. ISBN:0471056693, 2001.

[123] H. Y. Lam, G. S. K. Fung and W. K. Lee, "A Novel Method to Construct Taxonomy Electrical Appliances Based on Load Signaturesof", in *IEEE Transactions on Consumer Electronics*, vol. 53, no. 2, pp. 653–660, May 2007.

[124] G.y. Lin, S.C. Lee, Y.J. Hsu, W.R. Jih, "Applying Power Meters for Appliance Recognition on the Electric Panel". In *Proceedings of the 5th IEEE Conference on Industrial Electronics and Applications*, pp. 2254—2259, Melbourne, Australia, 15–17 June 2010.

[125] S. Barker, S. Kalra, D. Irwin, and P. Shenoy, "NILM redux: The case for emphasizing applications over accuracy", *NILM-2014 Workshop,* Austin, TX, June 2014.

[126] T. Zia, D. Bruckner, and A. Zaidi, "A Hidden Markov Model based procedure for identifying household electric loads", *In Proc. IECON-2011 37th Annual Conf. IEEE Industrial Electronics Society,* pp. 3218–3223, Melbourne, Australia, Nov. 2011.

[127] Electric Power Research Institute, *Low-cost NIALMS technology*, Technical Report TR-198918-V1, Sept. 1997.

[128] J. Powers, B. Margossian, B. Smith, "Using a rule-based algorithm to disaggregate end-use load profiles from premise-level data", *IEEE Computer Applications in Power*, vol. 4, no. 2, pp. 42–47, 1991.

[129] 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 & Building*, vol.30, pp. 245–259, 1999.

[130] M. L. Marceau and R. Zmeureanu., "Nonintrusive load disaggregation computer program to estimate the energy consumption of major end uses in residential buildings", in *Energy Conversion & Management*, vol. 41, pp. 1389–1403, 2000.

[131] 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*, Pittsburgh, US, 2012.

[132] H. Altrabalsi, V. Stankovic, J. Liao, L. Stankovic, "Low-complexity energy disaggregation using appliance load modelling", *AIMS Energy*, Issue 4, vol. 1, pp. 1-21, 2016.