Residential Energy Data Analysis Using Green Button Data

Huan Yang
CSE Department, Lehigh University
huy213@lehigh.edu

Advisor: Prof. Liang Cheng
CSE Department, Lehigh University
cheng@cse.lehigh.edu

July 2014

TABLE OF CONTENTS

1. Introduction

1.1 Background

Officially launched in January 2012, the Green Button Initiative has enabled utility customers to easily and securely access their own energy usage information in a user-friendly format [1]. This industry-led standardization effort provides an essential building block of innovative applications that help customers understand and manage their energy usage. Based on the Energy Service Provider Interface (ESPI) data standard, the Green Button data standard includes both an XML-based format for energy consumption data and a data exchange protocol for automatic data transfer from a utility to a third party based on customer authorization [1]. The continuous development of the Green Button data standard has attracted the attention of both utility customers and application developers. It is believed that the availability of standardized energy consumption data can help utilities, government, and customers develop a mutual vision of future power grids for a more energy-efficient society.

Researchers have been studying non-intrusive load monitoring (NILM) for two decades. The purpose of NILM research is to develop algorithms and the underlying data collection systems to determine the power consumption of every individual appliance in a residential house or an industrial building and present such information to customers [2] [3]. As the solutions proposed by early researchers have relied on specialized transducers or the dense deployment of assorted sensors, the usability issue of most NILM solutions has hindered the wide adoption of those ideas [4]. Recently, researchers have devoted every effort to make NILM algorithms become more practical, including the minimization of user involvement in the system deployment process. Consequently, the utilization of smart meter becomes a plausible and attractive approach to minimizing installation effort. In this respect, Green Button data has the potential for supporting NILM applications while minimizing users' effort to install extra sensory equipment. However, the performance of representative NILM algorithms when using Green Button data has not been explicitly discussed. In this report, we investigate the practicality of applying NILM algorithms using Green Button data. Specifically, we evaluate the performance of representative algorithms with data gathered at sampling rates currently defined in the Green Button data standard [5].

We notice that existing NILM algorithms have their limitations when using data generated at low sampling rates, such as 1/900 Hz (i.e., 1 data sample per 15 minutes). Firstly, the accuracy of an NILM algorithm's estimation of the power consumption of an individual appliance decreases as the sampling rate gets lowered. Secondly, the number of appliances also affects the performance of NILM algorithms. Many appliances reside closely with one another in the feature space, making it difficult to distinguish them correctly. From utility customers' point of view, NILM algorithms are useful because they can provide an estimated appliance-specific breakdown of energy consumption, which further directs customers' attention toward potential energy-saving opportunities. However, we also observe that the estimation of NILM algorithms is not accurate enough to replace the results of real-time appliance monitoring [6]. It is more appropriate to think of existing NILM solutions as tools that enable customers to find the less-efficient categories of appliances in terms of energy consumption.

In the field of building energy management and monitoring, researchers are interested in the identification of abnormal energy events [7] [8]. These events can be associated with appliance malfunctioning, failures, or incorrect operations. Techniques of anomaly identification align well with the needs of building operators and property managers: It is laborious to identify energy consumption anomalies manually, and operators' effort should be concentrated on diagnosing and resolving abnormal events. As a result, we also study the implementation of anomaly detection techniques using Green Button data. We identify the performance requirements of anomaly detection techniques and discuss the combination of NILM algorithms and anomaly detection techniques. As illustrated in Fig.1, anomaly detection can help customers narrow down the list of suspicious events. Then, customers have the option of further investigating specific events by setting appropriate time duration for NILM algorithms, which make it easier to identify the categories of appliances that should be carefully examined. The advantage of this approach is twofold. On the one hand, we only need to execute computationally expensive energy disaggregation algorithms on chunks of data containing potential anomalies. On the other hand, we can eliminate the days containing anomalies so that energy disaggregation algorithms can construct more accurate models of appliances. Alternatively, anomaly detection algorithms can also be invoked when the user has doubts on the results of NILM algorithms, which may be provided by utility companies or authorized third parties. In this case, anomaly detection techniques can be conducted on data of a short period of time, such as 1 week or 1 month, which also effectively avoids the impact of seasonal changes on anomaly detection.

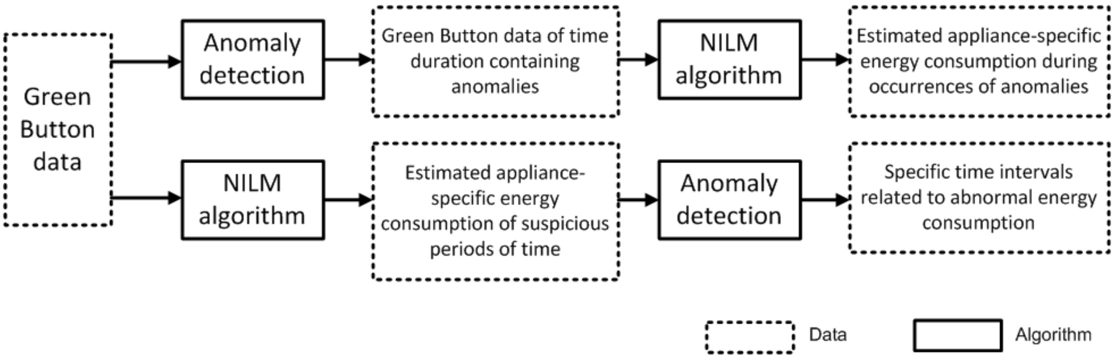

Fig. 1: Two possible ways to combine NILM algorithms and anomaly detection techniques.

This report is organized as follows. In the remainder of this section, we will introduce publicly available data sets for NILM research and describe the procedures we take to convert data from the REDD data set [9] into Green Button format. In the next section, we focus on the evaluation of representative NILM algorithms on simulated Green Button data sets. We identify the restrictions on the applications of these algorithms and notice that these algorithms may be used in combination with anomaly detection techniques. Then, the third section of this report covers anomaly detection using Green Button data. We show that anomaly detection algorithms can identify abnormal events in our simulated Green Button data set. Finally, we conclude the report with a summary of our observations and future research directions.

1.2 Simulated Green Button Data Set

To help customers and developers attain better understanding of the Green Button data standard, many utilities have published samples of energy consumption reports in Green Button format [10] [11] [12]. However, only whole-home energy consumption information is included, whereas the ground truth information for the development of NILM and anomaly detection algorithms is absent. For NILM algorithms based on pattern recognition techniques, status and energy consumption of individual appliances must be presented together with Green Button data. For anomaly detection techniques, ground truth information is also necessary. Intuitively, we need this information to verify whether or not the data points selected by anomaly detection algorithms are related to abnormal energy consumption events.

Moreover, we will show in later section that the ground truth, which is not included in sample reports published by utility companies due to privacy concerns, is indispensible to improve the performance of various anomaly detection techniques. Therefore, it is necessary for us to generate a data set with ground truth, preferably from energy consumption data of real residential houses. In this subsection, we introduce publicly accessible data sets in the NILM research community and describe our simulated Green Button data set generated from the REDD data set [9].

1.2.1 Overview of Publicly Accessible Data Sets

It is widely accepted that the availability of data sets is crucial to the development of practical and efficient NILM algorithms. As a result, several research groups, while approaching NILM problems from diverse perspectives, have published their data sets. Here we focus on the Reference Energy Disaggregation Data set (REDD) [9] due to its popularity. The REDD data set is particularly developed for the development of NILM algorithms that use pattern recognition techniques. Six U.S. residential houses are monitored for time durations ranging from 2 weeks to a few months using three layers of sensors. At the whole-home level, current transducers are deployed to monitor high-frequency whole-home currents at 15 kHz in the power mains. Voltage on a single phase is also monitored at the same frequency. In the circuit breaker panel of each house, current and voltage sensors are also installed and circuit-level data is labeled with the category of attached appliances. This value of circuit level information varies from house to house because the wiring schemes used in different houses may not always be the same. At each wall socket, a smart plug is placed between the socket and its attached appliances. The smart plug wirelessly communicates the energy consumption information of appliances to a data concentrator. Plug-level data is also labeled with the status of the attached appliances. Recent versions of these smart plugs [13] can also communicate with smart phones or personal computers. Since the REDD data set provides ground truth information, we are able to produce a simulated Green Button data set using the labels and sensor readings.

The Smart* data set [14] is generated using both multi-layer energy sensors and miscellaneous event sensors. Similar to the structure of the data collection system for the REDD data set, three houses are monitored using 3 layers of voltage and current sensors. One unique characteristic of the Smart* data set is that renewable energy sources, such as on-site solar panels and wind turbines, are also monitored. Energy-related events, including those collected by motion sensors and thermostat controllers, are also compiled into the data set. Another research group takes on the approach of dense deployment of sensors and the results are organized into the Building-level Fully-labeled data set for Electricity Disaggregation (BLUED) [15]. More energy-related events are monitored in the data collection system for the BLUED data set, such as the change in the level of illumination as well as the emergence and intensity of sounds. Both data sets provide abundant information for the development of event-based NILM algorithms. However, it is hard for customers to glean event-based energy consumption data. In this report, we only consider auxiliary information that can be easily obtained by customers. Hence, we do not choose the Smart* data set or the BLUED data set when generating our simulated Green Button data set. Two other data sets are also relevant to the analysis of residential energy consumption data. The Tracebase data set [16] includes real-power consumption information of 122 appliances from 31 categories sampled at 1 Hz. However, each appliance is monitored separately and usually at different locations. The time duration of monitoring also varies from 1 day to a few days. The Almanac of Minutely Power data set (AMPds) [17] provides both whole-home and circuit-level energy consumption readings from a residential house in Canada. Gas and water consumption information is also gathered, but appliance-level information is not explicitly labeled in the data set.

We choose the REDD data set for the following reasons. First, the information provided in the data set is enough for developers to train classifiers used in NILM algorithms. Second, the REDD data set do not provide auxiliary information on energy-related events that must be collected by specialized sensors. Third, data sets depending on the dense deployment of sensors and data concentrators may not be feasible in residential environments due to the cost and complexity of the data collection systems. Finally, it should be noted that auxiliary information, though not provided in either the REDD data set or sample Green Button data, can be incorporated if the corresponding data collection devices are deployed. Hence, we generate a simulated Green Button data set, which is used in the evaluation of both the NILM algorithms and anomaly detection techniques in this report.

1.2.2 Conversion of REDD Data Set into a Simulated Green Button Data Set

The REDD data set [9] includes energy consumption information gathered using three layers of energy sensors from 6 U.S. homes. To convert this data set into a simulated Green Button data set with ground truth, we need to down-sample the plug-level current and voltage readings in the REDD data set. It should be noted that not all appliances are attached to wall sockets. For example, Heating, Ventilation, and Air-Conditioning (HVAC) systems may be hardwired. Therefore, the corresponding circuit-level signals are down-sampled. Such low-frequency circuit- and plug-level data is included in our simulated data set as labels indicating the operational status and power consumption of individual appliances. To evaluate the performance of NILM algorithms with respect to the number of appliances, we synthesize the plug-level (or circuit-level, when the corresponding wall socket is not monitored or not present) data from different houses into a single data set. As a result, we use the sum of real-power consumption of individual appliances as the whole-home energy readings. The list of appliances included in our data set is provided in the next section.

For the purpose of anomaly detection, we generate abnormal events using the following methods. Our first approach is to replace the individual energy consumption readings of ON appliance with the energy consumption data when it is OFF. The selected appliance is turned on during the same period of time in most days of the same type. In other words, the first type of anomalies is related to ad hoc activities or appliance failures. Our second approach is to change the amplitude of the real-power consumption of an appliance, which may indicate appliance malfunctions or incorrect operations. Examples of anomaly detection are presented in the third section of this report.

2. Energy Disaggregation Using Green Button Data

2.1 Background

According to a recent report by the U.S. Energy Information Administration (EIA) [18], electricity consumption by the residential sector accounts for 29% of the overall consumption of the United States. Compared to the endeavor of energy conservation in commercial and industrial sectors, the effort to help residential customers optimize energy consumption appeared to be limited. The development of the smart grid, especially the large-scale deployment of smart meters, offers the opportunity to develop data-intensive applications and help customers increase energy efficiency. For residential customers, the implementation of Green Button provides secured access to the energy consumption data of their own houses or apartments, which can also be revealed to a trusted third-party under user authorization [1]. Such an open platform catalyzes the development of data-analysis tools that assist a customer to gain better understanding of how energy is consumed by his/her family and whether there is any possibility of energy conservation. From the perspective of utility companies, increased customer awareness and involvement also facilitate further deployment of smart meters and inauguration of policies such as the time-of-use rate option [20]. Therefore, the development of data-analysis tools is the first step towards active user participation in future development of smart grids and associated services.

In a residential house, electric energy is consumed by all the appliances in use. It is sometimes hard for customers to get hold of the whole picture of the energy consumption of the premise. Several field experiments have shown that residential users will consume less energy when timely and accurate feedback on their energy consumption behaviors is provided [20]. A review of field experiments concludes that it is necessary to develop tools that help user better understand the impacts of their own behaviors and decisions on the energy consumption of their houses [21]. All these studies suggest that measures of energy consumption feedback at finer granularity promote energy-awareness and help user identify energy-saving opportunities. In this section, we introduce data-analysis techniques, collectively known as energy disaggregation algorithms, which automatically generate detailed energy usage information of individual appliances. Such information can be used directly as a measure of feedback. It can also support experts in analyzing energy consumption patterns of residential houses.

Energy disaggregation, as known as non-intrusive load monitoring (NILM), converts a time series of energy consumption data into an appliance-specific breakdown. Initially described by G. W. Hart in 1992 [22], this problem attracts the attention of more and more researchers in recent years and has become an important problem in the field of computational sustainability [23]. By decomposing an array of overall energy consumption readings into its constituent parts consumed by individual appliances, energy disaggregation algorithms allow a residential customer to acquire detailed information of energy consumption of all the appliances across the premise and thus better understanding of the electricity bill. An energy disaggregation system is composed of both disaggregation algorithms and necessary hardware for data collection. Though some existing work proposes the deployment of specialized and sometimes sophisticated equipment, minimizing the levels of invasiveness of these systems is believed to be the crucial step towards practical NILM solutions. Hence, economic solutions to energy disaggregation should take advantages of the widespread deployment of smart meters and the implementation of the Green Button by utilities. Such an approach will lower the barriers to the popularization of energy disaggregation systems among residential customers. On the other hand, the current configuration and capabilities of smart meters pose a serious challenge to the development of energy disaggregation algorithms, which is not explicitly addressed by the previous work.

The objective of this section is to evaluate the performance of representative energy disaggregation algorithms on simulated Green Button data. The remainder of this section is organized as follows: In the next subsection, we review general approaches to energy disaggregation. Then, we define the problem of energy disaggregation using Green Button data. Next, we introduce feasible solutions that can be implemented using Green Button data. The algorithms are tested using our synthetic data set and the implementation details are described in their respective subsections. Finally, we conclude this section with the current status and future directions of research in energy disaggregation.

2.2 Related Work

The concept of energy disaggregation was first proposed by G. W. Hart [22] and was termed non-intrusive load monitoring, which emphasized that such a system should introduce little inconvenience for residents and install as few sensors as possible inside a residential house. Though several researchers tried to approach this problem using optimization algorithms [24] [25], most researchers employ pattern recognition techniques. Existing solutions are immensely disparate in their levels of intrusiveness. Some researchers propose the dense deployment of sensors to gather as much information as possible. Sensor-based approaches are easy to implement because signals emitted by appliances and events related to operation or status changes of appliances can all be captured by specialized transducers. Appliances with similar power ratings are probably different in other aspects, such as the time periods of a day when they are used and the user activities associated with these appliances. In other words, most sensor-based approaches simplify the problem of energy disaggregation by collecting auxiliary information. Furthermore, assorted commercial off-the-shelf (COTS) sensors in the market with networking and communication capabilities also facilitate the development of these data collection systems. For example, the system developed in [26] uses acoustic signals accompanying the operations of kitchen appliances. Another system proposed in [27] includes smart phone applications that guide users through the deployment and configuration processes of the data collection system. However, most sensor-based solutions [56] require complicated manual deployment of multiple sensors, ranging from temperature sensors to motion detectors, which raises the barrier to popularization.

Recently, more and more researchers start to develop algorithms that minimize the level of invasiveness of the underlying data collection system. Such approaches are evidently less intrusive than sensor-based solutions. However, it is also more challenging to solve energy disaggregation problems while minimizing the level of intrusiveness of the data collection system. Many appliances consume similar amount of energy and are operated at approximately the same period of time. For instance, the power rating of a toaster is typically 1000~1300 Watts, which overlaps the typical range of the power rating of a microwave (500~1500 Watts) [28]. As a result, researchers look into the possibility of distinguishing appliances using features extracted from electricity consumption data obtained at high frequencies, usually a multiple of the working frequency of the power system such as 15 kHz [9] and 12 kHz [15]. In [29], the authors propose a system that measures the real and reactive power consumption and found that this combination of features cannot distinguish resistive loads with identical power ratings. The system developed in [30] first computes harmonics and then combines these features with the power consumption levels. Signal transformations techniques, including Fast Fourier Transform (FFT) [31] and wavelet transform [32], are employed, while their performance with low-frequency energy sensor data needs further investigations.

To further utilize the information made available by smart meters and reduce the cost and complexity of the data collection systems, recent research on energy disaggregation begins to focus on low-frequency energy consumption data. In [4], the authors studied energy consumption patterns that can be extracted from whole-house energy consumption readings at sampling intervals of 1 hour. Other researchers considered sampling periods ranging from 0.1 second to 3 seconds. Although most authors mentioned that their algorithms can use readings from smart meters, the sampling frequencies used in these papers are still higher than the sampling frequencies defined by the Green Button initiative. The commonly supported reporting frequency in current Green Button implementations of most utility companies is 1/900 Hz, which is equivalent to 1 meter reading per 15 minutes. This motivates us to investigate the feasibility and performance of existing energy disaggregation algorithms using Green Button data.

2.3 Problem Formulation

As summarized in the previous subsection, energy disaggregation problems can be approached using different data collection systems, and we prefer to solve this problem using Green Button data. Besides, the level of intrusiveness of the data collection system should also be minimized to ensure system usability and ease of deployment. In this report, algorithms requiring minimal installation of extra sensors are selected and adapted for Green Button data. The configuration of the data collection system and our definition of the energy disaggregation problem are stated here. As all the algorithms described in the ensuing subsection use the same set of features, we also discuss why those features are considered and how they can be obtained before introducing specific algorithms.

2.3.1 Energy Disaggregation Using Green Button Data

Suppose we are provided with a series of consecutive smart meter readings for a residential house and a list of appliances in the house, our objective is to design an energy disaggregation algorithm that tells us the energy consumption of all the individual appliances. Formally, the smart meter readings are obtained at a regular frequency and organized in the form of Green Button data. We denote the Green Button data by a sequence of real-valued samples $y(0), y(1), y(2), \ldots, y(n)$, where $0 \le n \le \infty$. For example, each sample of a 15-minute Green Button data set is obtained at a sampling rate of 1/900 Hz. We also assume that any auxiliary information fed to the algorithm is also organized as time series obtained at the same frequency. Equivalently, we consider using sensors that are already deployed in residential environment or can be installed with little effort and all the sensors collect auxiliary information in a synchronized fashion. In practical systems, certain sensors, including thermo-sensors and humidity sensors, often generate continuous output and can be arranged into time series of any sampling frequency.

In addition to the assumption that all sensory data can be organized into time series obtained at the same set of sampling instants, the residential house under investigation must also satisfy the following requirements. First, no appliances are brought into or out of the residential house within the period of time when training data is generated. This requirement guarantees that the status of the house does not change so that enough data can be gathered for the training purpose. Since the pattern recognition algorithms we will discuss need training data to construct accurate models of the appliances (or classifiers for different appliances), this requirement also tells us how to choose training data: We desire training data spanning a long period of time when no appliances are added or removed. It should also be noted that this requirement does not apply to testing data, but it is recommended that we test the algorithms with data spanning a period of time when the same set of appliances are used. Second, the smart meter that generates whole-home energy consumption data must monitor the complete set of appliances in the house. Otherwise, our disaggregation algorithms should only consider the set of appliances whose power consumption is actually measured by the smart meter.

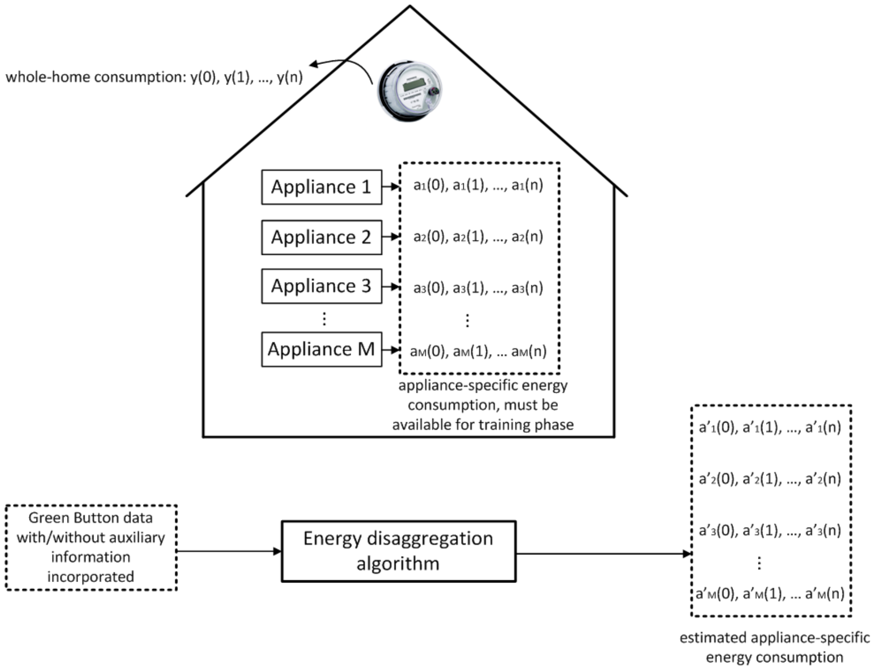

Fig. 2: Energy disaggregation generates estimated appliance-specific energy consumption information from whole-home consumption data, such as those available in the Green Button data. Note that the appliance-specific energy consumption information must be available for the training phases of NILM algorithms.

Assume that there are M appliances in the residential house under investigation, the output of an energy disaggregation algorithm using Green Button data or whole-home energy measurements from specialized sensors as in the REDD data set [9] will be M sequences of estimated energy consumption values of all M appliances. As shown in the Fig. 2, the output sequence $a'_1(0)$, $a'_1(1)$, $a'_1(2)$, …, $a'_1(n)$ represents the sequence of energy consumption values generated by Appliance 1; and a total of M output sequences should be produced by an energy disaggregation algorithm.

2.3.2 Obtaining Features with Few Additional Sensors

In the context of energy disaggregation using Green Button data, the smart meter actually serves as a sensor in our data collection system and produces low-frequency whole-home electricity consumption data. As discussed in previous subsections, using smart meter readings eliminates the necessity of installing current and voltage transducers in the mains panel, which makes such solutions more attractive than those using high-frequency features and a large number of sensors. The design principle of the entire data collection system should be consistent with the reason for which smart meter data is utilized. Therefore, we endeavor to exploit sensors that are already installed or easy to deploy in most residential houses. All the algorithms implemented and tested in this report use the same data collection system (or a subset of the system) described here.

The following table summarizes the extra sensors that may already be available or can be installed easily in most residential houses. Indoor temperature readings from thermo-sensors can be easily obtained from electronic thermometers. In [14], a low-cost in-home weather station is placed inside a residential house, which wirelessly communicates temperature, humidity, and air pressure information at a specified rate. Outdoor weather status can be obtained at no cost from internet weather services [33]. Both indoor and outdoor information can assist energy disaggregation algorithms to infer the status of appliances such as HVAC systems, humidifiers, fans and electric heaters. In addition, weather information also helps in identifying appliances that are only used under particular weather conditions. Some of these appliances are rarely used under other weather conditions, for instance, electric footwear dryers and snow throwers.

Table 1: Auxiliary information obtained at low cost for representative NILM algorithms

Additional Information	Sensor Used	Estimated Cost (Dollars)
Indoor temperature	Temperature sensors, wireless thermometers	\$5~30
Humidity and barometric information	In-home weather station	\$30~50
Outdoor weather information	Internet weather services	\$0 (no cost)
Time of day, day of week, date	Not required	\$0 (no cost)

Time and date information is also encapsulated in Green Button data. Time of day can help in recognizing appliances that are routinely used during specific periods of time every day. The underlying assumption here is that user activities spontaneously present patterns on different days of week and during different time of day. This assumption is usually valid for residential houses and narrows down the list of appliances that are active during different time periods of a day. To recapitulate, data collection systems for energy disaggregation algorithms using Green Button data should utilize information from sensors that are already present or easy to deploy in residential environments so as to minimize the level of intrusiveness and lower the deployment barriers.

2.4 Existing Algorithms

2.4.1 Energy Disaggregation Using Factorial Hidden Markov Models (FHMMs)

Initially proposed in [24], the idea of factorial hidden Markov models naturally matches our objective of energy disaggregation. The original hidden Markov models discussed in [35] are not suitable for energy disaggregation because this approach constructs a multiple-state model for a residential house and the number of states grows exponentially with the increasing number of appliances. A widely accepted observation is that most appliances can be modeled using a small number of states. In fact, many existing pattern recognition algorithms for energy disaggregation, including those using high-frequency features, devise binary-state models for individual appliances. As factorial hidden Markov models allow the construction of models for individual appliances, the total number of states of FHMM-based algorithms is significantly reduced. In this and the next subsections, we discuss the details of the original FHMM algorithm and its variants. The models we use for individual appliances are binary-state models. In other words, we assume that all the appliances have only two states, namely “ON” and “OFF” states. As shown in prior work [36], binary-state models can handle most appliances because an appliance with multiple states normally has a primary state when it is turned on.

In the context of energy disaggregation using residential Green Button data, we assign a binary-state model to each independent appliance. Suppose there are N appliances in the residential house under investigation, we construct a set of N binary-state models $\{M_1, M_2, \ldots, M_N\}$. The value of each model M_i ($i = 1, 2, \ldots, N$) is taken from the set $S_i = \{S_0, S_1\}$, where S_0 represents the “OFF” state and S_1 the “ON” state. For each model M_i, a state transition matrix A_i determines the state transition between S_0 and S_1. For multiple-state models, the set S_i will contain multiple distinct values, each representing a particular state of the corresponding appliance. An entry a_{ij} in the state transition matrix A_i is the probability of state transition from state S_i to state S_j. Formally, this probability is written as

$$a_{ij} = P\left(M_{t+1} = S_j \middle| M_t = S_i\right)$$

where $1 \leq i,j \leq N$. Smart meter readings contained in Green Button data are our observations of the aggregated output of all the models for individual appliances. In other words, we assume that the state of any model M_i cannot be directly observed. Instead, each hidden state S_j of a model M_i emits a certain symbol O_j^i that can be observed. In the context of energy disaggregation, O_j^i is the power consumption of appliance M_i when in state S_j. Thus, for each appliance, we define a set of symbols $V_i=\{V_0, V_1\}$ and our observation O_j^i satisfies that $O_j^i \in V_i$. To elaborate the connections between internal states and emitted symbols of a model M_i, an emission matrix E_i is used. An entry e_{jk} in matrix E_i is defined as the probability that the emitted symbol O_j^i of model M_i takes on the value V_k when it is in state S_j. Formally, this is defined as

$$e_{jk} = P\left(O_j^i = V_k \middle| M_i = S_j\right)$$

To complete our definition of factorial hidden Markov models, we define an array C_i for each model M_i. The number of components in C_i equals the number of states of the model M_i and the value of each component c_k in the array C_i is the probability that the initial state of model M_i is S_k. Therefore, the components of C_i can be written as

$$c_k = P\left(M_{t=0} = S_k\right)$$

where $M_{t=0}$ denotes the initial state of model M_i. The smart meter readings are the aggregated output of the whole set of models observed at regular time intervals. Since we denoted smart meter readings as a sequence y(0), y(1), …, y(T), each sample value in this sequence is the sum of the observed output values of all models of individual appliances at the same time instant.

When using the FHMM algorithm, we assume that each model is independent from all the other models. Furthermore, we also assume that the state of a model M_i at time instant t only depends on the state of M_i at time instant t-1. Thus, a set of factorial hidden Markov models can be fully described by the set of state transition matrices, initial state distribution matrices, emission matrices, all the sets of state symbols, and all the sets of output symbols. Given the input sequence of smart meter readings, we can consider the FHMM approach as a complete set of generative models. Our FHMMs have several parameters that must be determined through training using Green Button data. The parameter estimation process is described in details in [34] [37]. In effect, the parameter estimation process finds the sequences of internal states of all FHMMs that best explain our observation, namely the smart meter readings. The algorithm we used is described in Algorithm 1. To initialize the parameters, we simply assume that all appliances are turned off at the beginning. Since we use binary-state models, the emitted symbols associated with “OFF” states of appliances are chosen as the initial value of the output symbols. Then, we start an expectation maximization (EM) procedure which iterates until the expected value of the log-likelihood function is maximized. This procedure enables us to determine the optimal values for all the parameters of the established FHMMs. After the EM procedure converges, we use maximum likelihood estimation (MLE) to find the best set of sequences of states that explains the observation values. Note that the data used by the EM procedure is the training data while the data used by the MLE procedure is the testing data.

Algorithm 1: Parameter Estimation for FHMMs

Given a sequence of whole-home observations Y

1: λ={A_1,...,A_n,E_i,...,E_n,C_i,...,C_n} $\leftarrow$ initial parameters

2: repeat

3: $\lambda' \leftarrow \lambda$

4: $\lambda \leftarrow \underset{\lambda}{\operatorname{argmax}} \left\{ E\left[\log P\left(Y, M \middle| \lambda\right) \middle| Y, \lambda' \right] \right\}$

5: until λ converges

6: estimated $M^* \leftarrow \underset{M}{\arg\max}\, P\left(M \middle| \lambda, Y\right)$

The algorithm is tested using the simulated Green Button data extracted from the REDD [9] data set. The results are presented at the end of the next subsection so as to give a better understanding of the performance of various FHMM-based algorithms. It should be noted that only real-power consumption signals are used in the FHMM algorithm. To train the FHMMs, we need to label every whole-home energy consumption data sample with the vector of power consumption of individual appliances. After the training procedure converges, we only need to provide whole-home energy consumption data to obtain the estimated appliance-specific energy consumption information.

2.4.2 Conditional Factorial Hidden Markov Models (CFHMMs) and Conditional Factorial Hidden Semi-Markov Models (CFHSMMs)

So far our factorial models are developed solely based upon the energy consumption information. As mentioned previously, incorporating extra information, such as time of day or indoor temperature information, can help us further distinguish appliances having similar power ratings and usage patterns. Under the original FHMM framework, the values of state transitions probabilities, which may change under different external conditions, are fixed once Algorithm 1 converges. Hence, we need to extend FHMMs for scenarios where the state transition probabilities are not constant but conditioned on extra features. This leads us to the CFHMMs described in [37].

We can further incorporate the idea of FHSMMs into CFHMMs to formulate a more complex set of appliance models, which is termed conditional factorial hidden semi-Markov models (CFHSMMs). The FHSMMs allow us to use state occupancy distributions other than exponential distributions. The EM procedure for both CFHSMMs and CFHMMs is computationally intractable. As a result, we use the same approximation approach used in the FHSMM algorithm. The estimation of external-feature conditioned state transition probabilities is computationally expensive but also provides the flexibility of incorporating extra features if possible. One of the extra features considered in [37], namely the interdependence between appliances is not included in our implementation because our simulated data set combines energy consumption data of appliances from different residential houses. Even in a single residential house monitored in [9], the interdependence between appliances is not as obvious as that of the appliances studied in [37]. The reason for this is that the authors in [37] studied the set of appliances closely related to entertainment activities, such as the TV set and PlayStation3.

The testing results of all 4 FHMM-based algorithms are presented in the following figures. F-measure is used in our evaluation, which is consistent with existing work such as [37] [38]. The reason that F-measure is used in energy disaggregation literature is that the ON events of appliances are rare events. Accuracy is not a good metric because any algorithm that always guesses that an appliance is in

OFF state will have a high accuracy. F-measure is computed as the harmonic mean of precision and recall. Formally, we have

$$\text{Precision}=\frac{TP}{TP+FP} \qquad \text{Recall}=\frac{TP}{TP+FN}$$
$$F-measure=\frac{2\cdot\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}$$

where TP (true positive) is the number of ON-state estimation results generated by an NILM algorithm that are consistent with the ground truth. TN (true negative) is the number of OFF-state estimation results generated by an NILM algorithm that are consistent with the ground truth. FP (false positive) represents the number of ON-state estimation results that are incorrectly generated by an NILM algorithm, whereas FN (false negative) is the number of OFF-state estimation results that should have been classified as ON events. The CFHSMM algorithm outperforms all the other FHMM-based algorithms because all the features readily available in residential environment are incorporated. However, the estimation of external-feature conditioned state transition probabilities is computationally prohibitive. Since both CFHMMs and CFHSMMs effectively expand the set of state transition probabilities for estimation, we need to collect more data for the training phase. The FHSMMs, on the other hand, only brings slight improvement over FHMMs. It should also be noted that the testing phases of all 4 FHMM-based algorithms produce estimated sequences of states of factorial models that best explain the overall output. Such estimation may correctly explain the amount of energy consumed over a relatively long period of time, but the estimated states of appliances at specific time instants can be wrong. Therefore, it may be hard for utility customers to recall their actual power consumption behaviors based on the estimation generated by FHMM-based algorithms, even though these algorithms do produce appliance-specific breakdowns of energy consumption for any time intervals.

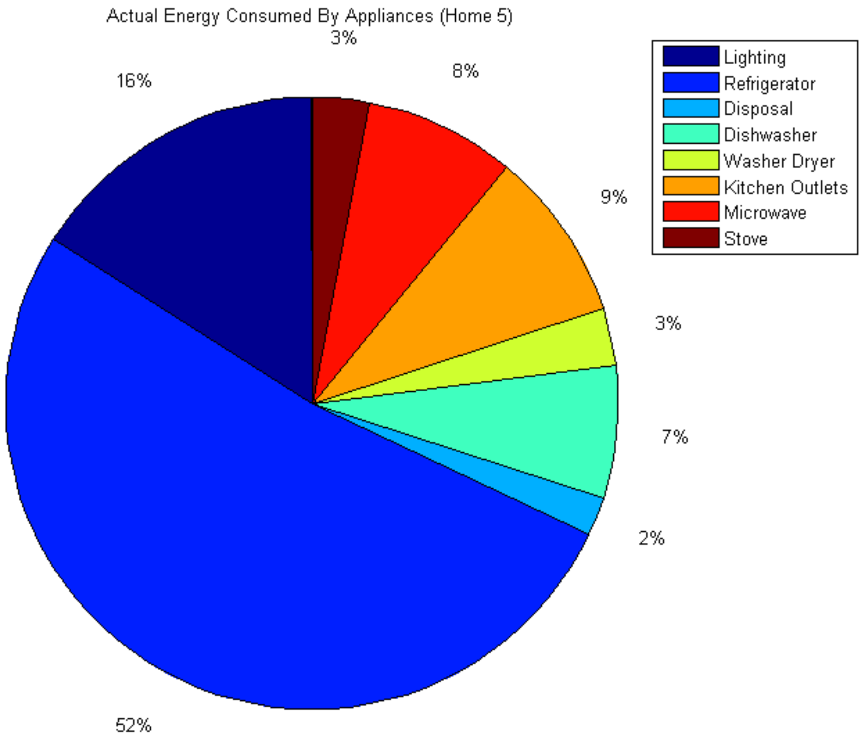

Fig. 3: True distribution of appliance-specific energy consumption over a week for Home 5 from the REDD [9] data set. This figure can only be obtained when we generate a simulated Green Button data set with ground truth.

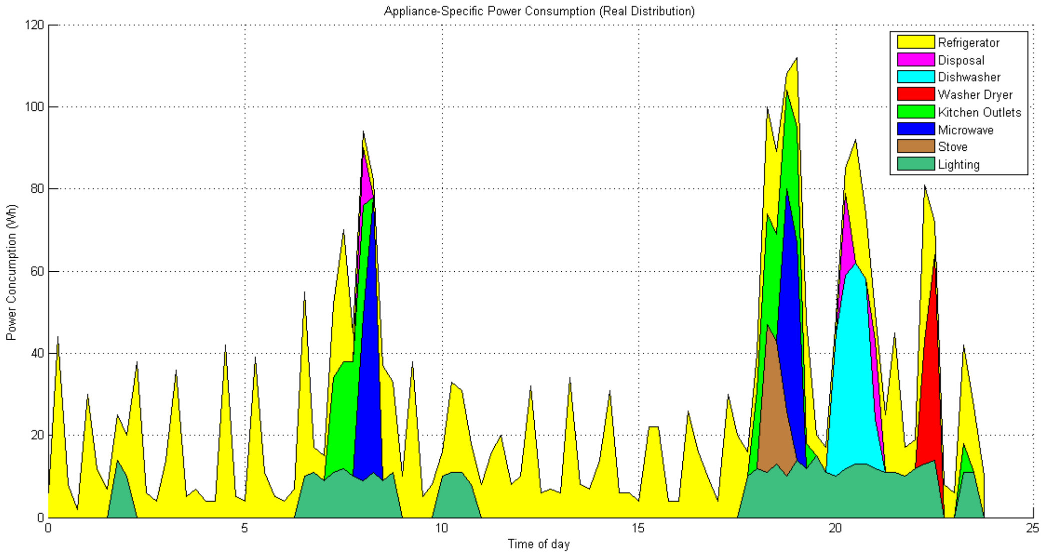

Fig. 4: Appliance-specific power consumption of Home 5 in REDD [9] data set on a Monday. This figure shows the true distribution among appliances and is used to compared with the results generated by energy disaggregation algorithms.

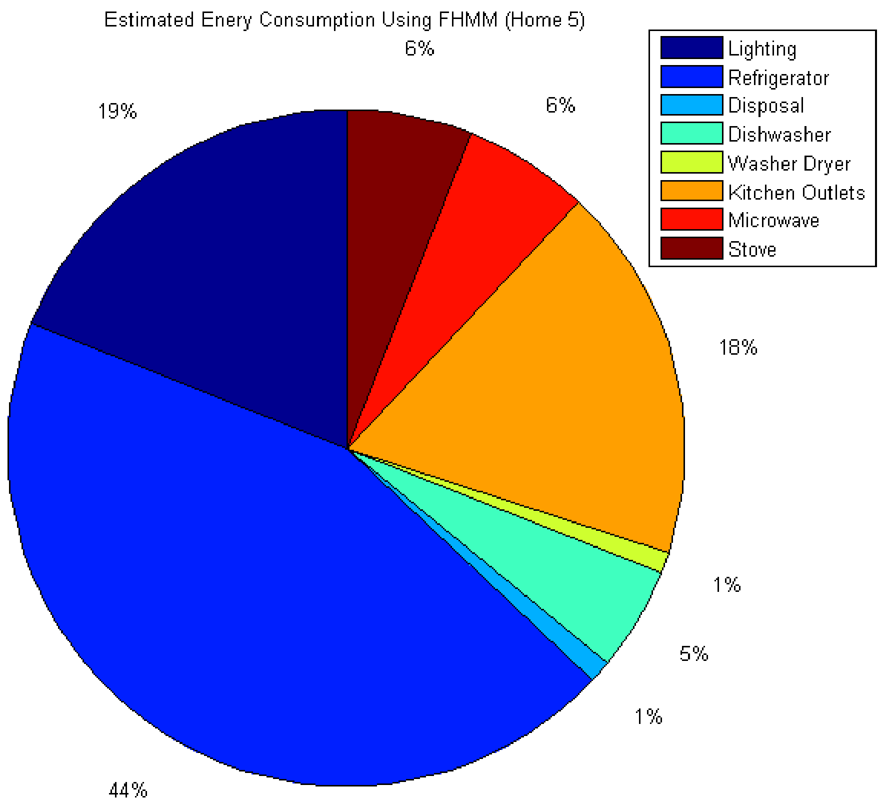

Fig. 5: FHMM-estimated distribution of appliance-specific energy consumption over a week for Home 5 from the REDD [9] data set. The simulated Green Button data set is used.

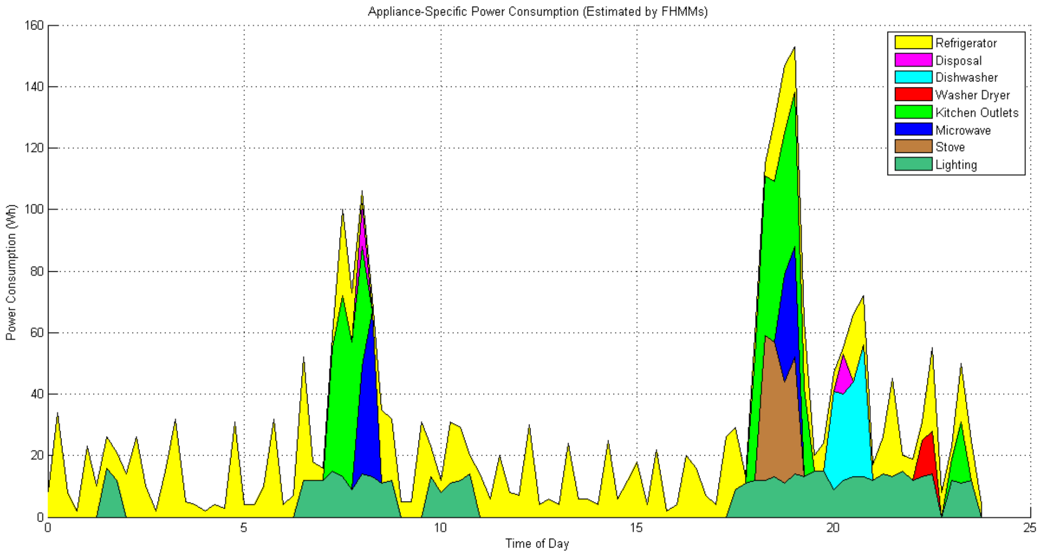

Fig. 6: Appliance-specific power consumption of Home 5 of the REDD [9] data set on a Monday, estimated by the FHMM algorithm.

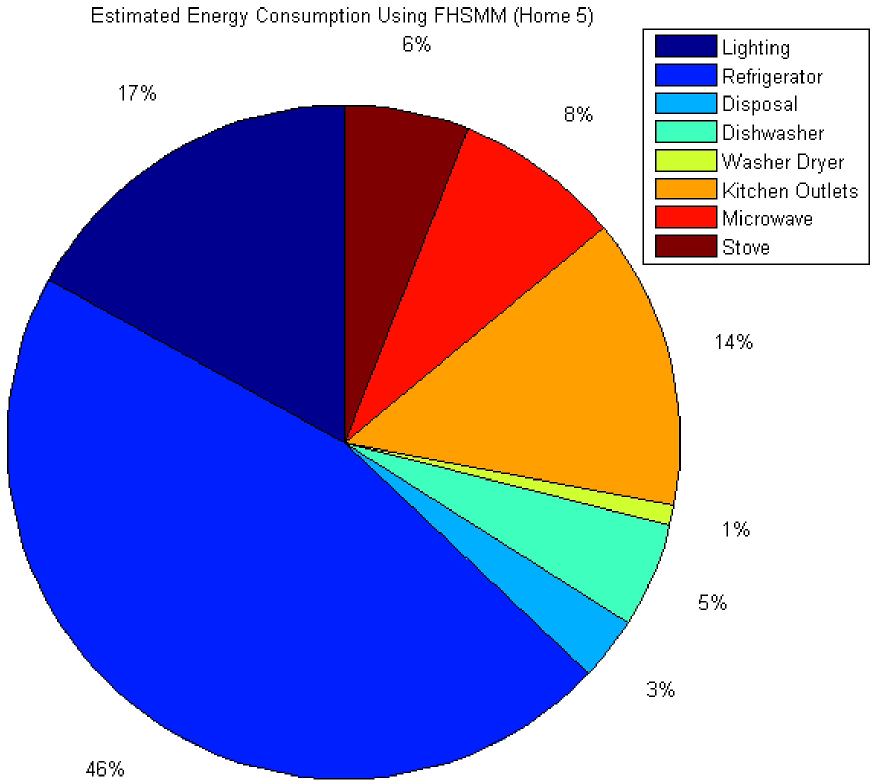

Fig. 7: FHSMM-estimated distribution of appliance-specific energy consumption over a week for Home 5 from the REDD [9] data set.

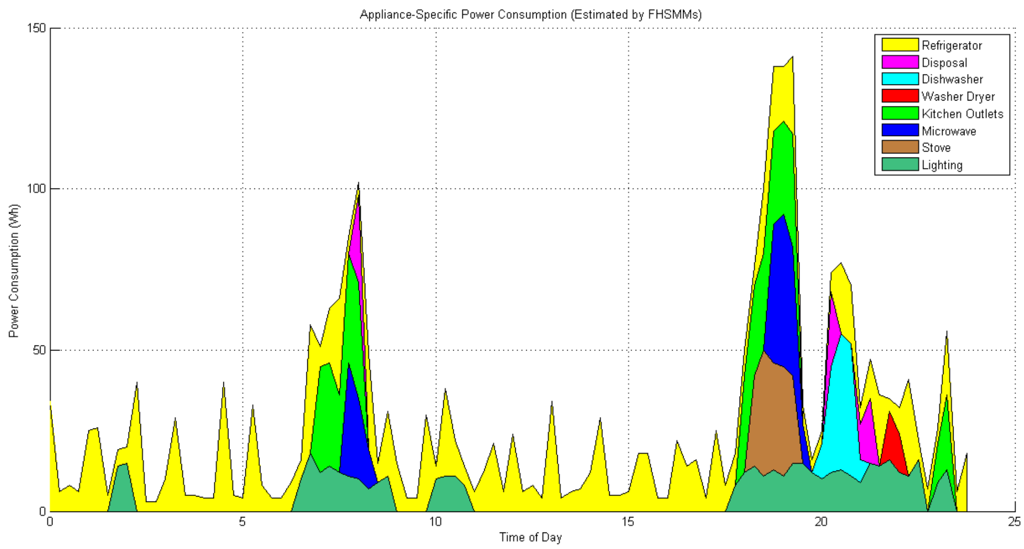

Fig. 8: Appliance-specific power consumption of Home 5 in the REDD [9] data set on a Monday, estimated by the FHSMM algorithm.

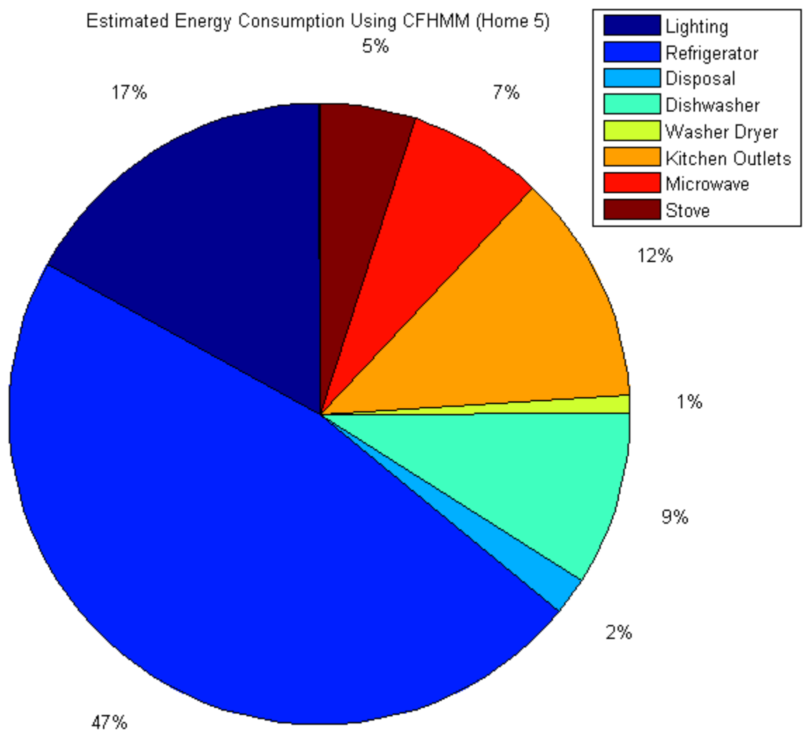

Fig. 9: CFHMM-estimated distribution of appliance-specific energy consumption over a week for Home 5 from the REDD [9] data set. Note that the auxiliary information used here is time of day, which can be easily extracted from Green Button data.

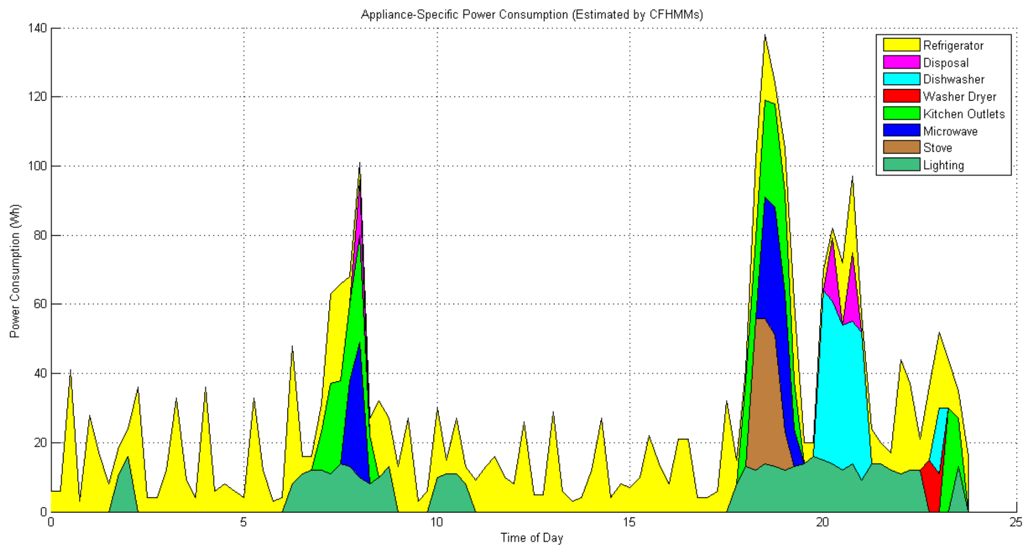

Fig. 10: Appliance-specific power consumption of Home 5 in the REDD [9] data set on a Monday, estimated by the CFHMM algorithm.

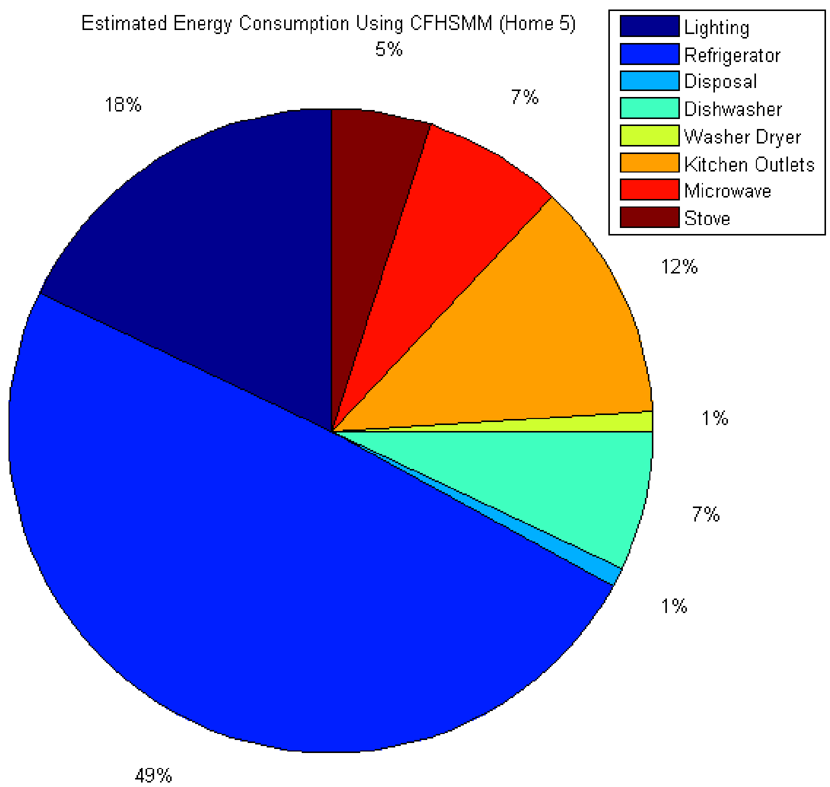

Fig. 11: CFHSMM-estimated distribution of appliance-specific energy consumption over a week for Home 5 from the REDD [9] data set. The extra feature incorporated here is also time of day.

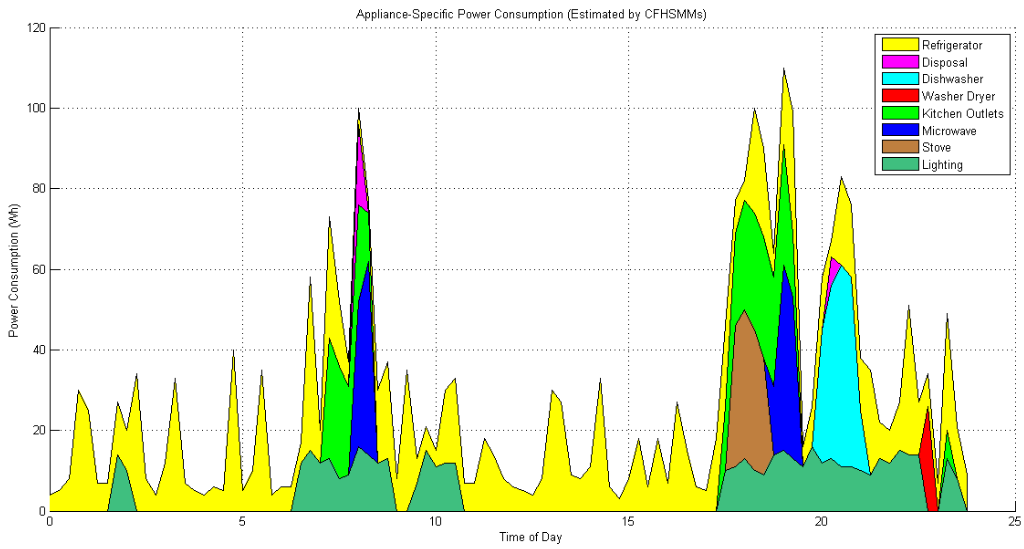

Fig. 12: Appliance-specific power consumption of Home 5 in the REDD [9] data set on a Monday, estimated by the CFHSMM algorithm.

2.4.3 Energy Disaggregation Using the K-Nearest Neighbor (KNN) Algorithm

K-nearest neighbor algorithms are also popular in energy disaggregation applications [39]. In the context of energy disaggregation, we obtain observations of a residential house at regular time intervals. The observation obtained at any specific time instant can be a scalar value of whole-home energy consumption or a vector of smart meter reading and extra information previously described. The generic KNN algorithm is described in Algorithm 2. The training samples for each class can be regarded as the patterns of the class. The fundamental idea of KNN algorithm is to find the pattern that is nearest to a new data sample. Similar to the training data used by FHMM-based algorithms, the training data for our KNN algorithm also requires that whole-home energy consumption readings be labeled by a vector of status and energy consumption of individual appliances obtained at the same time. To determine the distances between a testing sample and the training patterns, we need to select an appropriate distance metric. For the purpose of energy disaggregation, many authors [22] [40] choose Euclidean distance as the distance metric.

Algorithm 2: KNN Algorithm with Majority Voting (K=5)

Given: n training patterns c_1, c_2, ..., c_n, labeled with status of individual appliances and a testing sample s

1: i $\leftarrow$ 0
2: do i $\leftarrow$ i + 1; for each training pattern c_i
3: compute Euclidean distance d_i
4: until i = n
5: find the 5 nearest training patterns $d_1^{'}, d_2^{'}, d_3^{'}, d_4^{'}, d_5^{'}$
6: find the class c that owns the most training patterns from the 5 nearest patterns
7: label s with the label of c

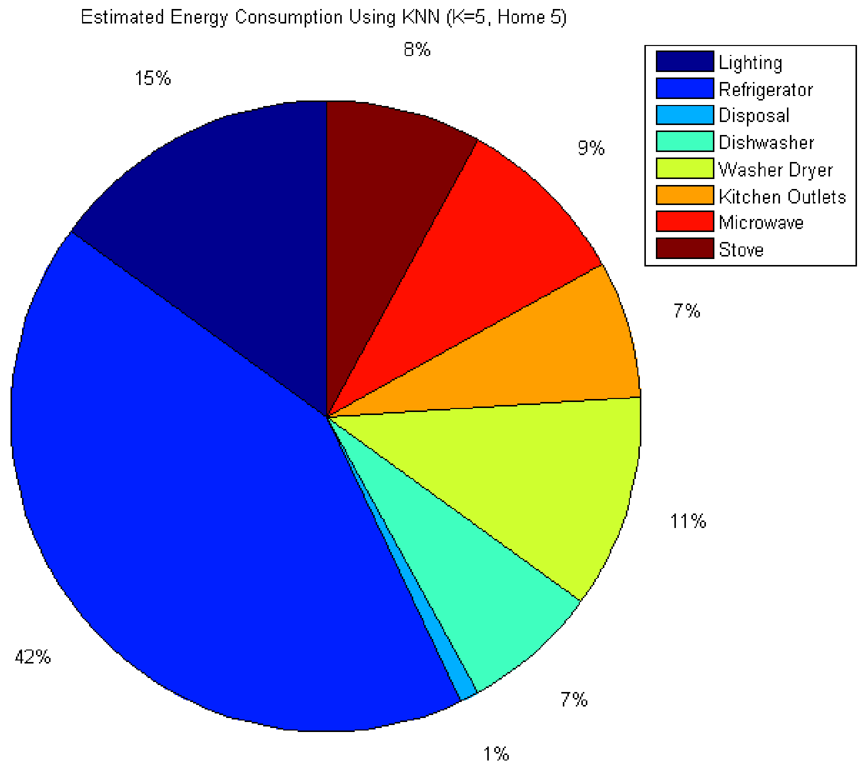

Fig. 13: Distribution of appliance-specific energy consumption over a week for Home 5 from the REDD [9] data set, estimated by KNN algorithm (K=5). The extra feature incorporated here is the time of day.

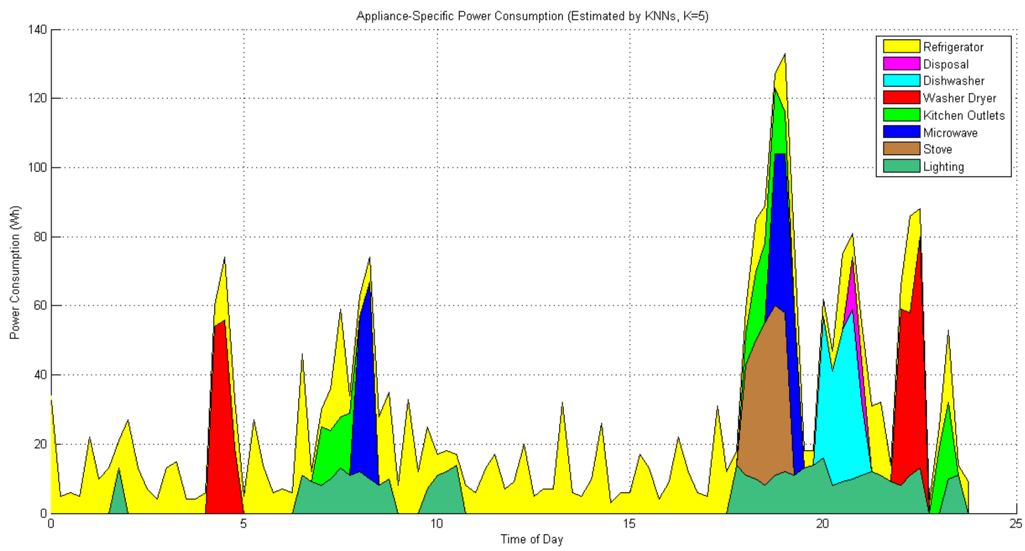

Fig. 14: Appliance-specific power consumption of Home 5 in the REDD [9] data set on a Monday, estimated by KNN algorithm (K=5).

As discussed in [41], increasing the value of K improves our estimation of the probability distribution of the state of nature in the feature space. However, a larger value of K also indicates higher cost of computation. In our KNN implementation, we choose K = 5. A majority voting procedure is used: First, we compute the distances between a new data sample and all the training patterns. Then, we find the 5 training patterns that are the nearest to the data sample. The class that owns the largest number of training patterns out of the nearest 5 will be used to label the new data. Ties are resolved arbitrarily.

Testing samples are presented to the KNN algorithm one at a time. Our testing results are presented in Fig. 13~14.

The performance of KNN algorithm is comparable to that of FHMMs and FHSMMs. The performance of KNN algorithm can be further enhanced by increasing the value of K or choosing a better distance metric [41]. However, this is also a computationally expensive approach and improving the computational efficiency of KNN algorithms has been extensively discussed in pattern recognition literature [42] [43]. Since our primary objective is to investigate the practicality of NILM algorithms with Green Button data, we do not implement these improvement techniques.

2.4.4 Energy Disaggregation Using Support Vector Machines (SVMs)

Support vector machines (SVMs) can also be used in solving the problem of energy disaggregation [44] [45]. Similar to KNN algorithms, we find the feature space for pattern classification by organizing features into a series of vectors obtained at regular intervals. An SVM algorithm attempts to find decision boundaries that maximize the separation between classes. Instead of conducting classification directly in the feature space, an SVM algorithm first transforms the feature space into a higher dimensional space. This transformation can be defined by a kernel function, which is chosen by algorithm designer based on their knowledge of particular problems. In our implementation, a third-order polynomial is chosen as the kernel function. After the transformation, an SVM algorithm will look for the best decision surfaces that separate different classes. The preprocessing step transforming the training and testing samples into the higher dimensional space simplifies the representations of the decision surfaces, so parameter estimation in the training phase will be more effective [46].

The parameters need to be estimated are those of the decision surfaces, which determine the locations and shapes of decision surfaces in the transformed feature space. The training procedure can be described as follows. First, we use the kernel function, denoted by $\varphi(\cdot)$, to transform all training samples $x_1, x_2, \ldots, x_n$ into $y_1, y_2, \ldots, y_n$, where

$$y_i = \varphi\left(x_i\right) \qquad \text{i=1,2,...,n}$$

Our objective here is to find a linear discriminant function $g(y) = a^T y$ that completely describes the decision surface for each class. This linear discriminant function must correctly classify all training samples, which depends on the correct choice of the values of the components of the weight (parameter) vector a. For each of the n training samples, k = 1, 2, …, n, we assign either +1 or -1 to z_k depending on whether or not the training sample x_k belongs to the class. The distance from any separating hyperplane to a transformed training sample y is $|g(y)|/||a||$, and minimizing the norm of a (i.e., $||a||$) leads to the best separation between classes. Using the method of Lagrange undetermined multipliers, we construct the function

$$L\left(a,\alpha\right) = \frac{1}{2}\left\|a\right\|^2 - \sum_{k=1}^{n} \alpha_k \left[z_k a^T y_k - 1 \right]$$

Now, we need to minimize $L(\cdot)$ with respect to the weight vector a and maximize it with respect to the multipliers $\alpha_k \geq 0$ [47]. The problem can be reformulated as a constrained maximization problem and can be solved by several alternative schemes such as quadratic programming [48]. Testing is straightforward: New data samples are first transformed before classification. As all the decision boundaries are completely determined, we compute the distances from the testing sample to the decision surfaces and determine the region where the testing sample falls. Given a residential house with M appliances with N possible combinations of status, SVMs allow us to compute a set of decision surfaces that distinguish one class from all the other classes, thus resulting in N classifiers. Alternatively, we can also determine the decision surfaces of dichotomizers for every pair of classes. In our implementation, the dichotomizer approach is taken and N·(N-1) dichotomizers are designed using SVM algorithm, which results in a smaller ambiguous region [47] [48].

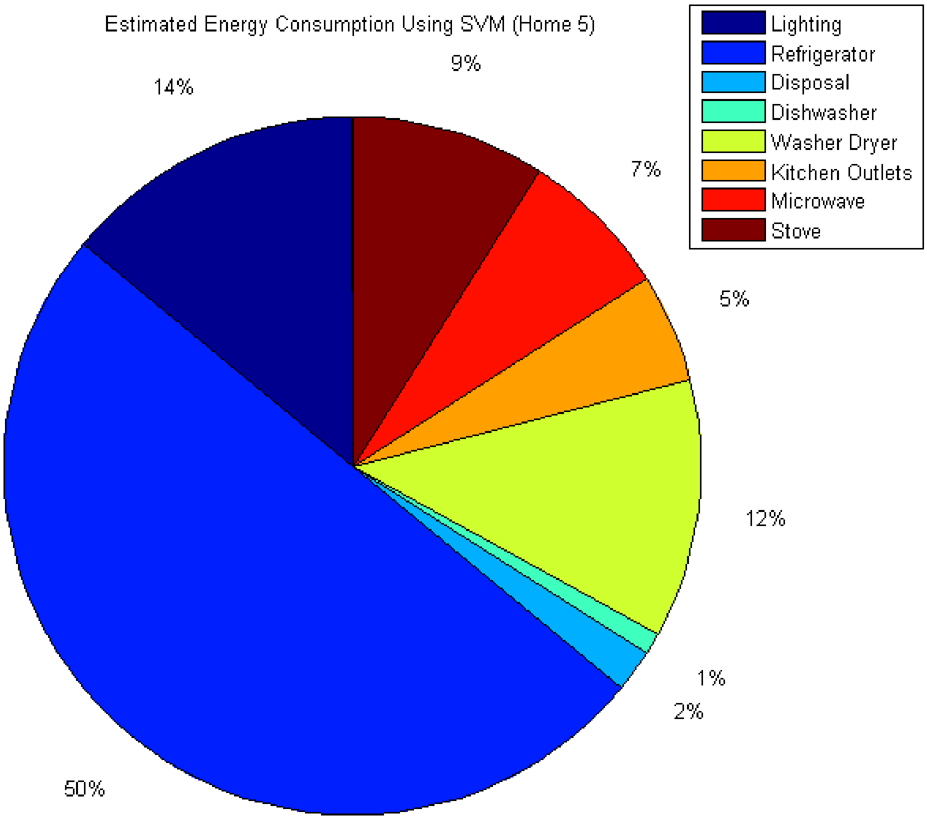

Fig. 15: Distribution of appliance-specific energy consumption over a week for Home 5 from the REDD [9] data set, estimated by our SVM algorithm. The extra feature incorporated here is the time of day.

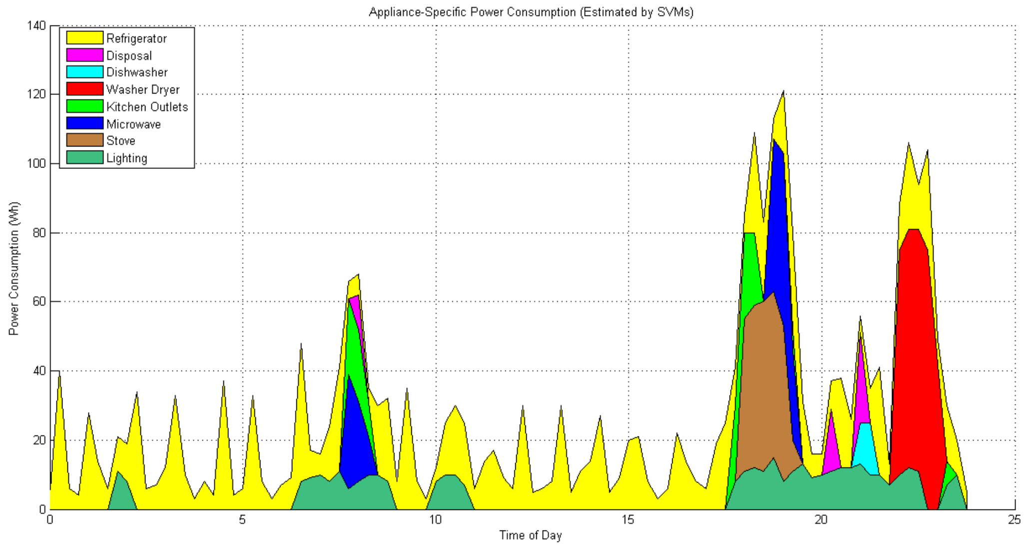

Fig. 16: Appliance-specific power consumption of Home 5 in the REDD [9] data set on a Monday, estimated by SVM algorithm.

The testing results of our SVM algorithm are shown in Fig. 15~16. The performance of SVMs is comparable to KNN and FHMMs, and the variants of FHMMs described in previous subsection only outperform SVMs marginally. It should be noted that the performance of SVMs can be further improved by choosing a better kernel function and that the purpose of our SVM-, KNN- and FHMM-based implementation is to show the baseline performance of existing techniques when low-frequency Green Button data is used. From Figs. 3~16, we observe that each algorithm has its own strength in distinguishing certain types of appliances. For example, KNN, FHSMMs and CFHMMs algorithms give better estimates of the power consumed by lighting. However, little is known about whether algorithms based on different pattern recognition techniques can complement one another. In fact, authors using algorithms based on the same pattern recognition technique [37] [49] also report results that reflect strengths in classifying disparate types of appliances, which suggests that the performance of energy disaggregation algorithms over different types of appliances is related to the decision boundaries and can be adjusted. In the next subsection, we further explore whether existing energy disaggregation approaches are feasible for practical applications by increasing the number of appliances and decreasing the sampling frequency.

2.4.5 Practicality of Energy Disaggregation Using Green Button Data

In the previous subsections, we evaluate FHMM-, KNN- and SVM-based energy disaggregation algorithms using our simulated data. The sampling frequency used in the evaluation is 10Hz (i.e., 10 readings per second). Currently, most utility companies provide customers with 15-minute data. Therefore, it is necessary to evaluate the performance of these algorithms at lower sampling frequency. 7 copies of the simulated Green Button data set are generated at sampling intervals of 0.1 seconds, 1 second, 10 seconds, 1 minute, 5 minutes, 10 minutes and 15 minutes from Home 5 in the REDD [9] data set. Figs. 17~18 present the performance of the three categories of NILM algorithms we described in previous subsections. As the sampling frequency decreases, performance of all the NILM algorithms deteriorates. The performance of CFHMMs and CFHSMMs is slightly better than that of the other algorithms. However, none of the algorithms perform well when the sampling rate is lowered.

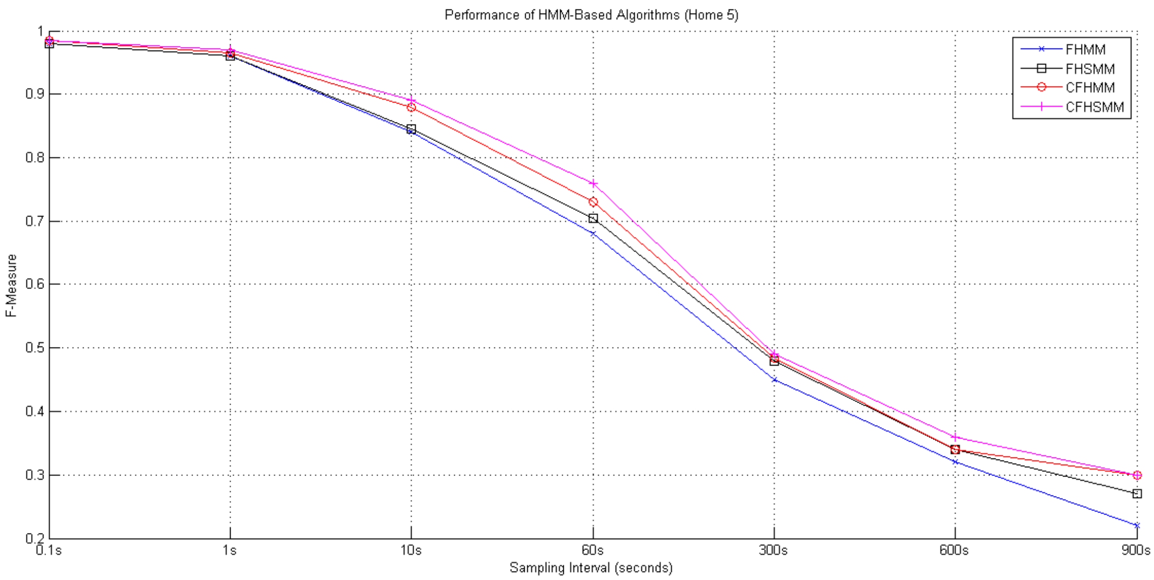

Fig. 17: Performance of FHMM-based algorithms at different sampling frequency. The simulated Green Button data is obtained from the same week as that of Fig. 3~9.

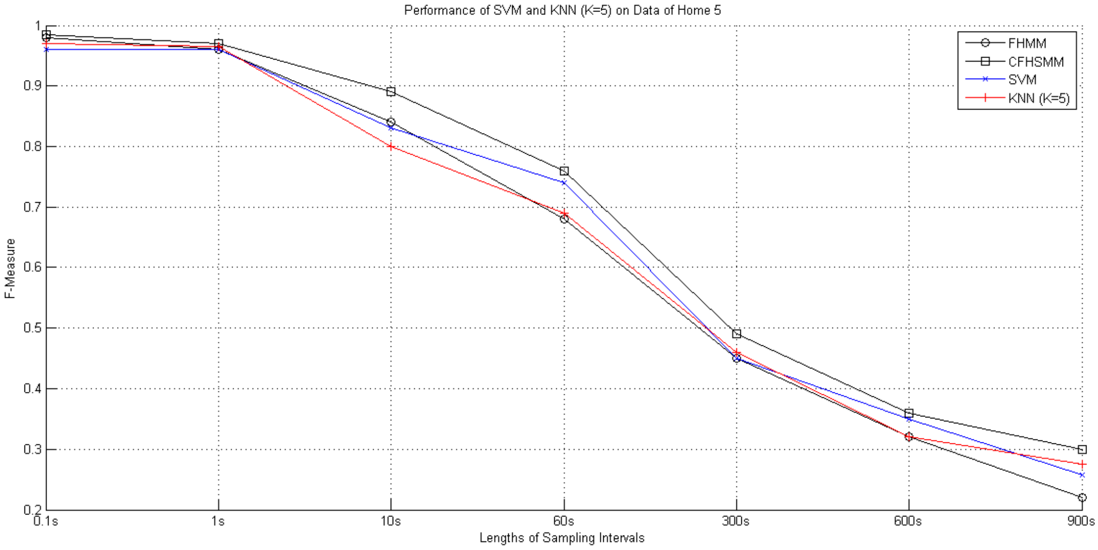

Fig. 18: Performance of SVM- and KNN-based algorithms at different sampling frequency, in comparison with FHMM and CFHSMM algorithms.

In addition the deterioration in performance at low sampling frequency, the number of appliances also affects the performance of NILM algorithms. Using the voltage and current waveforms of appliances from Home 1~5 in REDD [9] data set, synthetic Green Button data sets are generated by adding new appliances. The order in which appliances are added into the data set is shown in Table 2. Note that the sampling frequency used here is 10Hz (i.e., 10 samples per second). As the number of appliances increases, the performance of each algorithm deteriorates. The performance of FHMM-, KNN- and SVM-based NILM algorithms is shown in Figs. 19~20. In the original papers where the algorithms we implement are first proposed, data is collected from a small number of appliances. For instance, only 5 appliances are used to evaluate the FHMM-based algorithms in [37]. Our evaluation results suggest that existing energy disaggregation algorithms still need further enhancement so as to be applied in residential houses with many appliances.

Number of Appliances Included	Appliance(s) Added	Home
2	Lighting and Refrigerator	5
3	Dishwasher	5
4	Disposal	5
5	Washer Dryer	5
6	Kitchen Outlets	5
7	Microwave	5
8	Stove	5
9	Furnace	3
10	Outdoor Outlets	3
11	Oven	2
12	Bathroom GFCI	2
13	Air Conditioning	4
14	Smoke Alarms	4
15	Electric Heat	3
16	Electronics	1

17	Microwave	3
18	Stove	4
19	Washer Dryer	4

Table 2: The order in which appliances from different homes in REDD [9] data set is added to form the synthetic Green Button data set. The sampling frequency used is 10Hz.

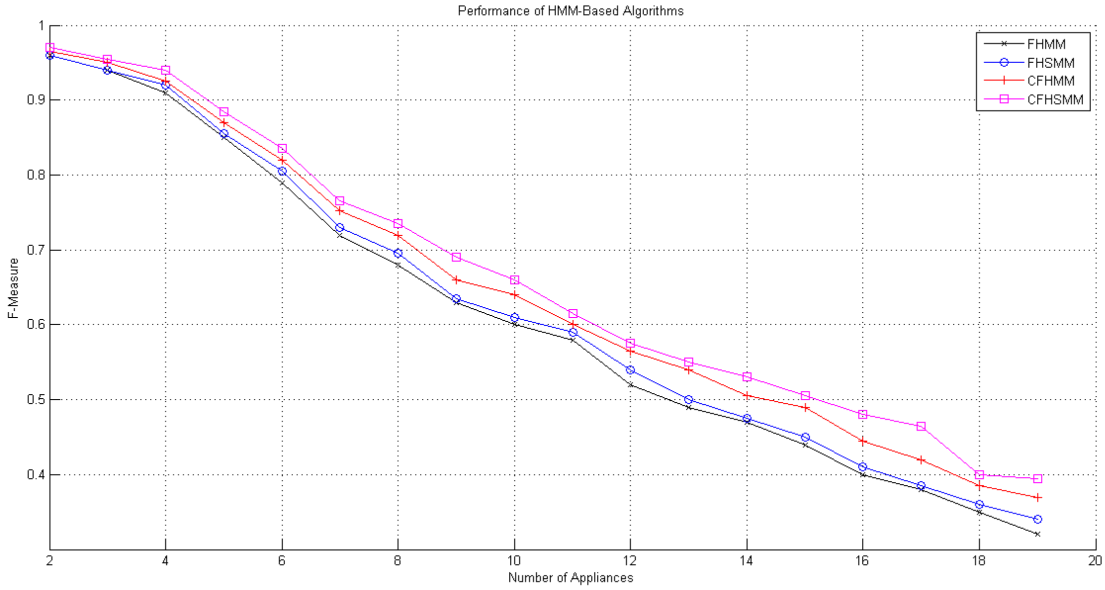

Fig. 19: Performance of FHMM-based algorithms with respect to the number of appliance.

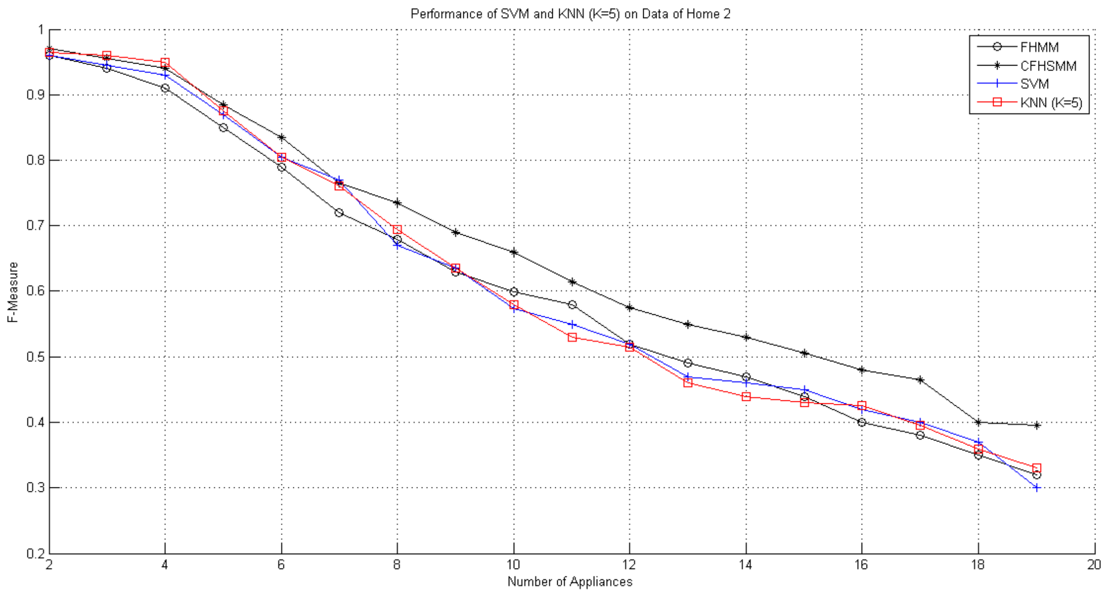

Fig. 20: Performance of KNN- and SVM-based NILM algorithms with respect to the number of appliances, in comparison with CFHMM and CFHSMM algorithms.

In summary, the performance of NILM algorithms implemented in this report deteriorates when the number of appliances increases or the sampling frequency decreases. In order to make use of these algorithms in real residential environment, researchers have proposed two categories of solutions. Some researchers resort to the deployment of extra sensors. To avoid raising barriers to residential installation, the problem is formulated as the deployment of a minimum number of sensors to achieve the desired performance [50] [51] [52]. Other researchers propose that future Green Button data standard support higher sampling frequency, either by upgrading the firmware of the smart meters or by adding extra hardware module for local data processing [53]. However, these solutions are still at the stage of development and are only evaluated using a small number of appliances.

2.5 Discussion and Future Work

From the testing results of different disaggregation algorithms, we observe that different approaches present their strengths in differentiating certain particular appliances. However, such difference is not significant. We can also adjust the decision boundaries of each algorithm to obtain performance gain on specific types of appliances, so it is not a good practice to use combine multiple computationally expensive algorithms. The most serious challenge posted by the problem of energy disaggregation using Green Button data is that the quantities of useful samples we can obtain for various categories of appliances are quite different. In our simulated Green Button data set, kitchen appliances are frequently used and enough samples can be collected. However, other appliances, such as outdoor outlets and smoke alarms, are rarely used. Hence, most samples we obtained for these appliances are related to the "OFF" state, and only very few samples associated with "ON" state are collected. As we further lower the sampling frequency down to 1 sample every 15 minute or even 1 sample per hour [4] [58], it is highly probable that no samples associated to "ON" state can be gathered. A possible solution to the problem of skewed distributions and imbalance between numbers of samples for different status and appliances is to simulate the sampling process and increase the number of samples for underrepresented states [54] [55]. This can be accomplished by reducing the number of overrepresented states by under-sampling methods and simultaneously increasing the number of underrepresented states by over-sampling methods. In particular, over-sampling can be done by synthesizing a new sample corresponding to a minority class by randomly choosing from the nearest neighbor of the minority sample point: First, we compute the difference between feature vector (of the minority sample point) under consideration and its nearest neighbor. Then, this difference is multiplied by a random value uniformly distributed between 0 and 1. Finally, the product is added to the feature vector under consideration. As a result, a new sample for the minority class is generated, which is not a replica of the existing data. In the future, we consider using this method to produce synthetic data sets that have a better balance between "ON" and "OFF" states of individual appliances.

The output of energy disaggregation algorithm is usually presented as estimated time series of power consumption of individual appliances. However, we note that both parametric and non-parametric disaggregation algorithms produce an explanation of observed whole-home energy consumption. Depending on the criteria used by different algorithms to determine the best explanation, existing energy disaggregation algorithms are not able to figure out the correct energy consumption of individual appliances at all time instants. The F-measure shown in our testing results gives the overall performance of algorithms, but the state estimation at a specific time instant can be completely wrong. More importantly, we sometimes do not have fine-grained control over the classification criteria we use. For example, it is well known that the localized constraint function for FHMMs leads to a completely different set of sequences of state transitions [34] [35], which can also explain the overall observation. This is one of the major obstacles to applying energy disaggregation techniques in real residential house: Customers may be confused by the state estimation generated by disaggregation algorithms at some specific time instants. In other words, the overall estimation results, in the form of energy consumption by specific appliances within a relatively long period of time (in our implementation, 1 week), will be helpful because it helps customers to understand appliance-specific energy consumption of the residential house quantitatively in a relatively long period of time. This may help customers identify the appliance that requires further attention. However, estimation results at specific time instants produced by disaggregation algorithms can only serve as a reference when a customer try to identify the specific time period of a day when an appliance malfunctions or fails.

3. Anomaly Detection Using Green Button Data

3.1 Background

Green Button data, as well as monthly electric bills, gives a detailed account of both regular and abnormal electric power usage. Abnormal power consumption patterns are usually caused by inefficient appliances or faulty operations. The identification of abnormal power consumption patterns can make customers more aware of overworn appliances; recognition of these irregular usage patterns also allows customers to reflect on the energy impacts of their behavior patterns and seize energy-saving opportunities that are not immediately apparent. However, the abstract concept of energy consumption seems to be difficult for customers to grasp, which makes it hard for customers to discern abnormal energy consumption patterns by themselves [56] [57]. Therefore, it is necessary to develop data analysis algorithms for anomaly detection and reveal specific energy-conserving opportunities to customers.

Anomaly detection techniques can bridge the gap between the availability of detailed energy usage and a customer's knowledge of his property and power consumption behaviors. There are various causes for unusual energy usage patterns. On the one hand, the aging of equipment sometimes results in a higher probability of malfunctioning. Customers may immediately realize the failures of certain appliances, such as a TV set. However, more and more appliances come with error recovery mechanisms, making it increasingly difficult to sense temporary faults. Since the impacts of overlooked malfunctions are implicitly included in the energy consumption records of a residential house, anomaly detection techniques should be developed and assist customers to capture and diagnose such issues. Furthermore, anomaly detection also reveals patterns related to human activities. For instance, trace of energy thief can be discovered if energy usage information of time intervals containing these malicious behaviors is available. Inadvertent operations, such as leaving the ventilation system on when it should have been turned off, may lead to energy consumption patterns that deviate from normal ones. Anomaly detection algorithms can pick out such energy events for further user inspection.

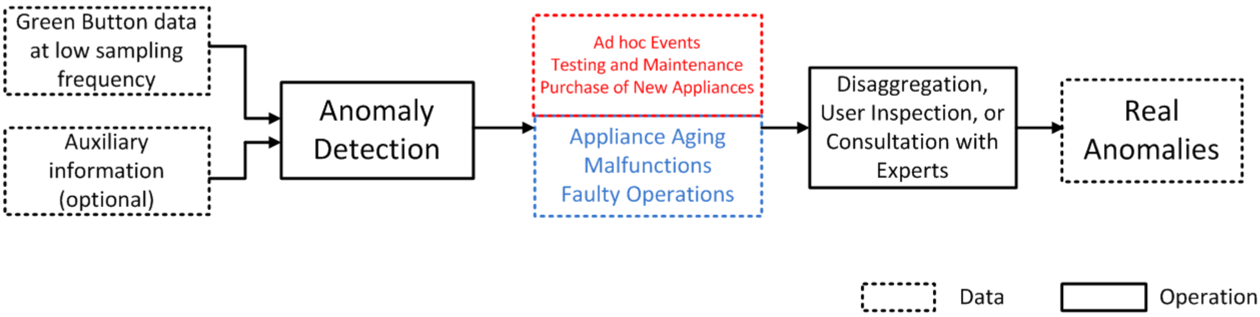

Fig. 21: Results generated by anomaly detection algorithms need to be further analyzed so as to identify real anomalies.

Anomaly detection algorithms can be regarded as an independent data-analysis module and can be integrated into existing building energy management systems [59]. By presenting energy events captured by anomaly detection algorithms, data analysis tools ease the burden of manual identification of suspicious energy usage patterns. Customers will be able to devote more time and effort to investigating possible links between suspicious patterns and user activities. It should be noted that the essence of anomaly detection [60] [61] is to find deviations from the majority of energy usage patterns. These deviations may also include patterns associated with ad hoc events such as parties. Therefore, user inspection is still necessary and hopefully minimized when anomaly detection is implemented.

Fig. 21 illustrates the idea of anomaly detection and how it may be incorporated into energy data analysis tools. User intervention is needed so as to reject outliers associated with relatively rare events, such as parties and purchase of new appliances. In this section, we first formulate the problem of anomaly detection and review the principles and requirements for the development of anomaly detection techniques. Possible approaches to anomaly detection are then reviewed. Representative solutions that can be implemented using Green Button data are discussed in details. To conclude this section, we look into the feasibility of applying existing techniques to residential data analysis using Green Button data.

3.2 Problem Statement and Related Work

3.2.1 Problem Statement

To capture unusual energy consumption patterns, we formulate the problem of anomaly detection as an outlier detection problem: Given real-power consumption readings y_0, y_1, y_2, …, y_i, …, y_n measured at equi-length time intervals ending at time instants t_0, t_1, t_2, …, t_i, …, t_n, find energy consumption reading and time instant pairs (y_i, t_i) that are related to abnormal power consumption events. The identified value pairs (y_i, t_i) are called outliers [60], which are statistically defined as data points widely separated from the majority of data points from the same group. In the context of anomaly detection using Green Button data, all energy consumption readings are directly associated with the energy consumption events happening at the same time. Consider a house with a set of m appliances, the states of which are represented by a vector of symbols E = (a_1, a_2, …, a_m). Each symbol in the vector denotes the state of an appliance. For example, for a reading lamp denoted by a_1, a_1 may take on the value from the set {'ON', 'OFF'} if the lamp has only two states. On the other hand, a_1 may take on any value ranging from 0 to 1 if the brightness of the lamp is adjustable (probably by a dimmer switch). The energy event completely characterizes the states of all the appliances at any time instant. Ideally, outliers are energy consumption readings associated to a specific energy consumption event that deviate from the main cluster of readings associated to the same event. Raw readings may be further processed into feature vectors, and our objective becomes outlier detection in the feature space.

When using Green Button data, we are usually not able to use the above-mentioned vector representations of energy consumption events because we typically do not deploy sensors within a residential house to capture the exact states of every individual appliance. Instead, we associate energy consumption readings with time instants at which they are generated. The underlying assumption for using time instants is that the power consumption events at the same time of different days are usually similar due to the habitual behavior patterns of customers.

Associating power consumption readings to time instants also introduces ambiguity. Grouped by time of day, some outliers can actually be caused by slight changes of daily routines. A customer who routinely turns on the air-conditioner shortly after his/her arrival at home may choose to switch on the ceiling fan instead. The customer has the freedom of choosing the particular way of reducing the room temperature. However, if the air-conditioner is the usual choice, energy consumption readings obtained when the ceiling fan is used may be considered as outliers. Such ambiguity does not exist if we use the vector representation of energy events because the two scenarios correspond to two different vector representations. Furthermore, time fluctuations of user behaviors also introduce ambiguity when time instants are used. For example, an occupant who prefers to take a shower at 9 pm has an important phone call to answer and takes the shower at 10 pm. This energy consumption pattern may also be considered as an outlier because it may generate an energy consumption reading significantly different from the others obtained at 10 pm. In short, the association between time instants and energy consumption readings exploits the implicit assumption that user behaviors exhibit regular patterns with respect to time of day. If an energy consumption and time instant pair (y_i, t_i) is related to a change of daily routines or time deviation but captured by an outlier detection algorithm, a false positive is generated by anomaly detection algorithm.

To minimize the amount of effort that customers need to devote to inspecting suspicious energy consumption patterns, we should develop outlier detection algorithms with a low false positive ratio. The false positive ratio is defined as the ratio of the number of false positive (FP) events to the sum of the numbers of false positive (FP) and true negative (TN) events. In the context of outlier detection, normal energy consumption patterns should be categorized as negative events. Normal patterns recognized as outliers by an algorithm consist of the set of false positives, whereas those correctly categorized as negative events are true negatives. A low false positive ratio ensures that most the captured outliers are associated to abnormal energy consumption patterns. Thus, customers can identify anomalies effectively by investigating value pairs presented by outlier detection algorithms. On the other hand, we also prefer a high ratio of true positives because we may otherwise miss the rare occurrences of anomalies. True positive ratio is defined as the ratio of the number of true positives (TF) to the sum of the numbers of true positives (TF) and false negatives (FN). Abnormal energy consumption patterns may be overlooked and reported as negative samples. Since the occurrences of anomalies are rare, a good outlier detection scheme should detect as many abnormal patterns as possible. In summary, practical anomaly detection techniques have both a low false positive ratio and a high true positive ratio.

3.2.2 Related Work

Outlier detection is an important field of statistical data analysis [61]. It deals with the rare occurrences of samples that evidently differ from most of the other samples. A number of fault detection techniques have been developed for heating, ventilating, and air-conditioning (HVAC) systems in buildings. There are two approaches to fault detection and diagnostic problems [62]. The first approach starts from faults in individual components, such as the outdoor air ventilation system [63]. The other approach looks for abnormal behavior at a premise level. Two early research efforts sponsored by the International Energy Agency are the Annex 25 [64] and Annex 34 [65] projects. It has been shown by these projects that daily energy consumption patterns can be rather distinctive from day to day. In [66], the authors present a method for detecting energy problems of buildings using an Energy Consumption Index (ECI). Their work reveals that there is a weekly cycle of ECI, and the authors develop a threshold-based outlier detection technique.

Recently, Chen et al. [67] present their field experiments of residential energy anomaly detection. In their experiments, sensors are densely deployed to monitor the status of appliances, record environmental information, and keep track of user activities. Although the focus of their work is to study the impacts of customers’ behaviors and establish the links between anomalies and user activities, the authors notice that outliers that are actually associated with abnormal energy consumption are rarely encountered. Several approaches have been applied to energy consumption data, including K-nearest neighbor (KNN) clustering and self-organizing maps [68]. However, these algorithms either demand expensive computations or require the deployment of multiple sensors inside a customer’s house. In this report, we review and evaluate algorithms that are computationally efficient for Green Button data.

3.3 Anomaly Detection Using Green Button Data

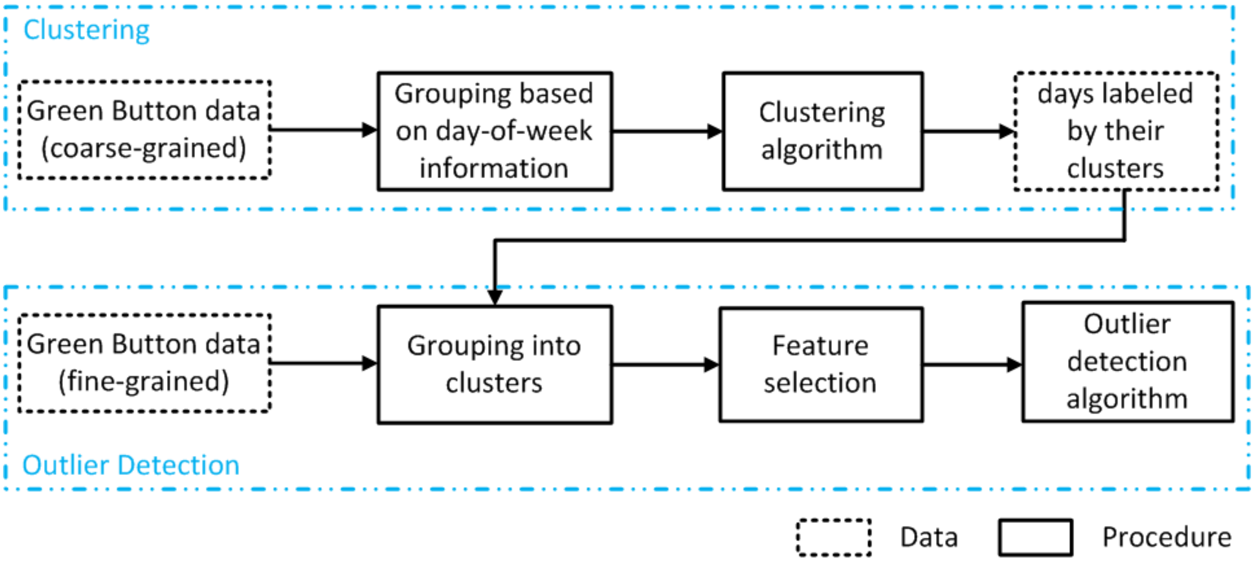

Fig. 22: Organization of clustering and outlier detection modules for anomaly detection using Green Button data

Typically, a practical solution to anomaly detection using Green Button data includes two components. An outlier detection algorithm is the core of the system that screens out suspicious energy consumption patterns. A clustering algorithm provides outlier detection algorithm with standards for grouping the data and effectively reducing the false positive ratio. In this report, we discuss all the necessary steps to complete the task of outlier detection, including feature selection, data grouping, outlier detection, and clustering. We test the algorithms on simulated Green Button data set and discuss possible applications of anomaly detection techniques.

3.3.1 Feature Selection

To fully utilize the information in Green Button data, we consider both direct energy consumption readings and statistical features [68]. Readings from smart meters, including peak demand and average consumption over time intervals of user-specified length, reveal the energy consumption patterns during the corresponding time intervals and can be used as features for outlier detection. Although these features are easy to obtain, one must be careful with the length of time intervals. All the outlier detection techniques discussed in this report allow users to specify a fixed value as the length for all the features, for instance, in a certain multiple of 15 minutes. If the user instructs the algorithms to detect outliers at time intervals of 1 hour, both peak demand and average consumption data should all use 1-hour intervals.

The energy consumption of a residential house has dynamic characteristics, and we can introduce statistical features using an auto-regression (AR) model to capture dynamical changes [68]. Suppose we have a sequence of 15-min energy consumption readings y(1), y(2), y(3), …, y(N), the next data y(N+1) can be predicted as a linearly weighted sum of p sample values immediately preceding y(N+1), namely y(N), y(N-1), …, y(N-p+1). The parameter p is the order of the AR model and is generally much smaller than the sequence length N. Formally, we have

$$y(t) = w + \sum_{i=1}^{p} a_i \cdot y(t-i) + e$$

where a_i is the ith coefficient of the pth-order AR model, w is the intercept variable, and e is a noise parameter. To estimate the parameters in a pth-order AR model, a stepwise least square procedure can be employed. The model order p, on the other hand, is not known in advance and needs to be determined for specific applications. For the purpose of outlier detection, it has been proposed that choosing p=2 should allow us to achieve a balance between computational complexity and model accuracy [69]. Therefore, it is sufficient to include only a_1 and a_2 in the feature vector.

The outlier detection algorithms discussed in this section provide customers with the flexibility to specify the set of features and the length of time intervals. This is necessary for the following reasons. First, future smart meters may have the capability to provide measurements other than peak demand and consumed real power, such as reactive power [53]; customers may deploy extra monitoring equipment that complement the smart meter. If extra data is available and can be processed using the same set of time intervals, outlier detection should exploit such information. Second, the statistical features are computationally expensive. In most cases, using only peak demand and average energy consumption can help customers find outliers efficiently without a significant loss of performance. Therefore, the outlier detection algorithms considered in this report can be used to provide expedited as well as in-depth analyses.

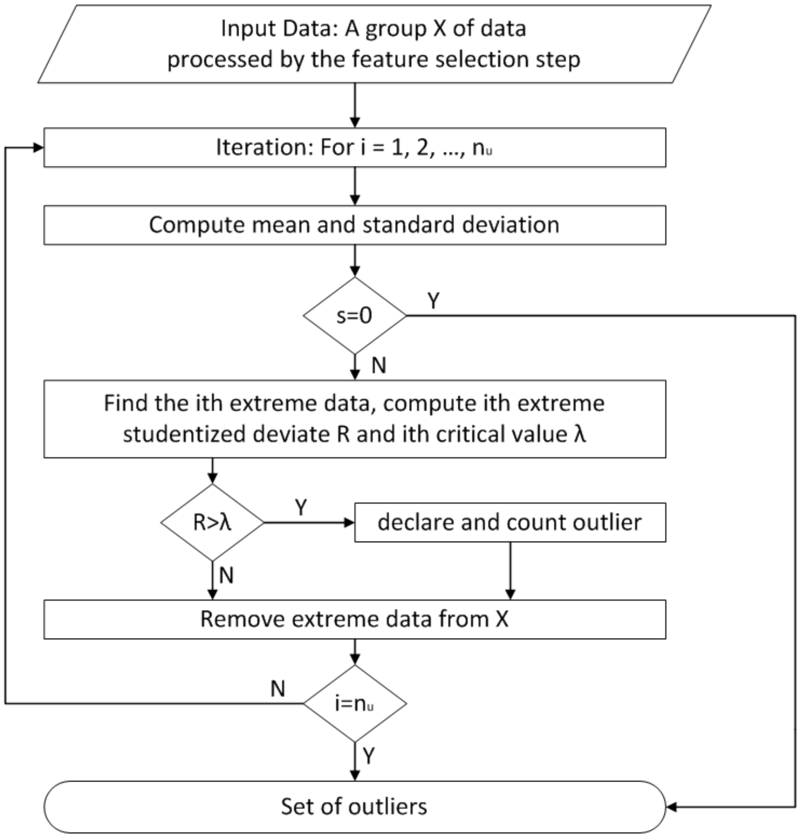

Fig. 23: Flowchart of GESD many-outlier procedure

3.3.2 Outlier Detection Algorithms

To make use of the data of Home 5 in the REDD [9] data set, we first sum up the power consumption values and generate simulated 15-minute energy consumption data. The simulated data is then processed using the feature selection schemes described in the previous subsection. In this section, we introduce two outlier detection algorithms and test them on simulated Green Button data. We first introduce the generalized extreme studentized deviate (GESD) many-outlier procedure [70], which works on a single feature at a time. Wilk's multivariate many-outlier procedure [71] is then discussed. This algorithm works on a feature vector and identifies the data vectors that may indicate abnormal power consumption events. In fact, practical anomaly detection systems typically rely on a specific outlier detection technique.

Therefore, we compare the two approaches to help system designer decide on the appropriate algorithm for implementation.

Both many-outlier procedures require two user-specified parameters α and n_u. The parameter α is the probability of incorrectly declaring one or more outliers when no outlier exists, and n_u gives the upper bound on the number of outliers. Typically, α is assigned to be 0.1; n_u can be determined by finding the largest integer that satisfies the inequality $n_u \leq 0.5(n-1)$, where n is the number of data points, such as energy consumption readings. Moreover, both algorithms require that data be divided into groups based on the types of days. For example, the energy consumption patterns of residential houses on weekends are usually different from those of weekdays. Such a preprocessing step is necessary because patterns of weekends will probably be considered as anomalies if data of weekends and weekdays is simultaneously presented to outlier detection algorithms. We will discuss how to classify energy consumption data according to the energy profiles of different types of days in the next subsection.

3.3.2.1 Generalized Extreme Studentized Deviate (GESD) Many-Outlier Procedure

The flowchart in Fig. 23 gives a stepwise illustration of the GESD many-outlier procedure [70]. Suppose we simply choose energy consumption within time intervals of equal length as the only feature and preprocess the data into 7 groups, namely Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday. When using Green Button data, we can easily organize data into groups based on the encapsulated time and date information. However, this arrangement has the issue of insufficient data especially when large time intervals are selected or data over a short period of time is available. In this section, we assume that our data set is sufficiently large and is divided into 7 groups based on types of days. We will discuss possible ways of classifying data into groups with similar energy profiles in the next subsection, and those techniques can be incorporated with our outlier detection algorithms without adaptation.

The GESD many-outlier procedure picks out at most n_u outliers from the given data set. Since we now assume that we divide energy consumption data into 7 groups, each group is fed to the GESD many-outlier procedure separately. Let $X=\{x_1, x_2, \ldots, x_n\}$ denote the data set of some specific day type, for instance, Saturday. The GESD many-outlier procedure processes one feature at a time and picks out an outlier candidate per iteration. The parameter n_u specified by the user determines the number of iterations of the procedure. The outlier candidate will then be removed from the data set, reducing the size of X by 1. At the beginning of each iteration, we first compute the mean m and standard deviation s of data set X as follows

$$m = \frac{\sum_{i=1}^{n} x_i}{n} \qquad s = \sqrt{\frac{\sum_{j=1}^{n} \left(x_j - m\right)^2}{n-1}}$$

where n is the number of sample in data set X during the current iteration. Note that the size of X is decreased by 1 after each iteration. When the standard deviation s equals 0, the GESD procedure terminates immediately because all the data points have the same value and there are no outliers.

After computing the mean and standard deviation, a data point with extreme feature value is chosen to be the outlier candidate for the current iteration. An extreme value is an element x_i in data set X that maximizes the function $|x_i-m|$. The extreme value, denoted by x_e, is then used to calculate the extreme studentized deviate R, which is given by

$$R = \frac{\left| x_e - m \right|}{s}$$

For an outlier candidate, the extreme studentized deviate R measures the normalized distance between the outlier candidate and the mean value of data set X of the current iteration. Next, we compute the critical value associated with the data set X of the current iteration, which helps us determine whether or not the selected extreme data point is an outlier. The critical value λ is given as follows

$$\lambda = \frac{(n-i) \cdot t_{n-i-1,p}}{\sqrt{(n-i+1)\left(n-i-1+t_{n-i-1,p}^2\right)}}$$

where $t_{n-i-1,p}$ is the Student's t-distribution with (n-i-1) degrees and a tail area probability p, which is determined using

$$p = \frac{\alpha}{2(n-i+1)}$$

The parameter α is the user-specified probability of incorrectly announcing one or multiple outliers when no outliers exist.

Once we obtain the extreme studentized deviate R and the critical value λ, we are able to determine whether the outlier candidate is an actual outlier by evaluating the inequality R> λ. If R is larger than λ, the selected extreme data point is regarded as an outlier. Otherwise, the selected data point is not an outlier. In both cases, the selected extreme data point is removed from the data set X before the algorithm proceeds with the next iteration. After a total of n_u iterations, the algorithm terminates and a set of outliers are ready for customer inspection. It should be noted that the GESD many-outlier procedure must work on the energy consumption data of different days of a week in order to find the outliers in respective groups of data. For each group of data, at most n_u outliers are found after exactly n_u iterations.

After all the outliers in a data set X are detected, an optional procedure assessing the level of severity of each outlier is recommended [70]. Using robust statistics, the proposed procedure computes a modified z-score, which quantifies how far and in which direction an outlier deviates from the mean value of typical observations. Ranking outliers by their levels of severity provides more information to help customers decide on which the specific outlier for investigation when multiple outliers are presented. The z-score procedure accepts two data sets as input, namely the set of outliers X_{out} and the set of non-outliers X_{normal}. X_{out} is the set of outliers detected by the GESD many-outlier procedure, but X_{normal} is not the remaining data set X when the GESD algorithm terminates. Instead, X_{normal} contains all the non-outliers, including those selected outlier candidates that are simply removed from data set X without being placed into outlier set X_{out}.

Once the set X_{normal} of normal observation is obtained, the mean m_{robust} and standard deviation s_{robust} of X_{normal} are computed as follows

$$m_{robust} = \frac{\sum_{i=1}^{r} x_i}{r} \qquad s_{robust} = \sqrt{\frac{\sum_{j=1}^{r}\left(x_j - m_{robust}\right)^2}{r-1}}$$

where r is the number of non-outliers in the set X_{normal}. The modified z-score of an outlier is determined from

$$z = \frac{x_{outlier} - m_{robust}}{s_{robust}}$$

where $x_{outlier}$ is an outlier taken from the set X_{out}. Anomaly detection software may rank outliers by their modified z-scores.

The GESD many-outlier procedure can also be applied to energy data in the form of feature vectors. Each feature vector consists of observations of residential energy consumption from different aspects, such as auxiliary information from low-cost sensors discussed in the previous section. When both power consumption and peak demand values are used, it is possible that a data point that is not considered as an outlier in terms of total energy consumption is still presented as an outlier due to inconsistent peak demand value. The GESD many-outlier works with a single feature at a time, but it can also be used to identify abnormal patterns observed in a specific dimension of the feature space. In other words, we need to run the GESD many-outlier procedure for $2n_u$ iterations if 2-dimensional data is fed to the algorithm. In this case, two set of outliers are generated, each containing outliers from the perspective of a specific feature. Anomaly detection software can take the union of the two sets and present the union set as the set of indentified outliers. When a huge volume of multi-dimensional data for a particular type of day is presented to this algorithm, the union set may become too large for user inspection. Anomaly detection software may allow users to rank the features by their perceived importance and only display anomalies of the most concern. In the next section, we discuss how to effectively detect outliers from multi-dimensional data.

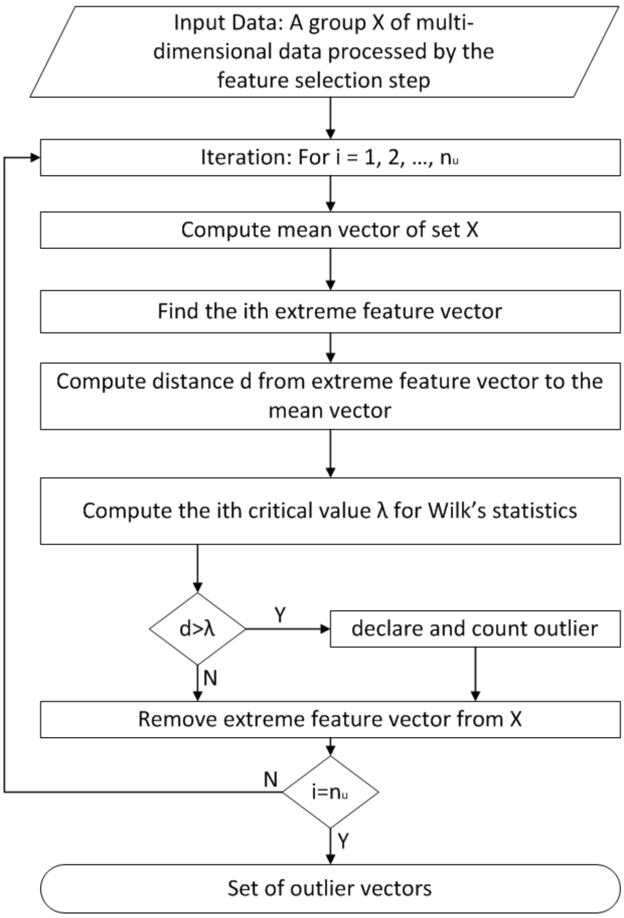

Fig. 24: Flowchart of Wilk's multivariate many-outlier procedure

3.3.2.2 Wilk’s Multivariate Many-Outlier Procedure

Suppose m features are extracted from Green Button data and a consecutive sequence of n data samples is available in the form of feature vectors. Thus, we have an n-by-m matrix X. Each row of X is an observation vector, while each column of X represents a dimension of the feature space. Similar to the GESD many-outlier procedure, users are required to specify two parameters α and n_u for the Wilk’s multivariate many-outlier procedure [71]. It is worthy of emphasis that the Wilk’s multivariate many-outlier procedure recognizes outliers in a circumspective way, considering all dimensions in the feature space. Therefore, no union operations are needed and the algorithm is well-suited for multi-dimensional data.

Wilk’s many-outlier procedure also works on data grouped by types of days. For data set X of a certain type of day, the algorithm identifies at most n_u outliers from X after exactly n_u iterations. In addition to the number of iterations, there is another termination condition for Wilk’s multivariate many-outlier procedure that must be checked at the beginning of every iteration. When the number of remaining data vectors in set X equals the number of dimensions plus 1, the algorithm must terminate immediately because the data becomes too sparse.

The flowchart of Wilk’s procedure is shown in Fig. 24. For each iteration, the sample mean vector **u** is computed using the data vectors in current data set X

$$u = \frac{1}{n-i+1}\sum_{j=1}^{n-i+1} x_j$$

where i denotes the current number of iteration. Note that n represents the number of elements of data set X, which is reduced by 1 after every iteration. Then, an extreme data vector is chosen using Mahalanobis distance d_j given by

$$d_j = \sqrt{\left(x_j - u\right)^T \Sigma^{-1}\left(x_j - u\right)}$$

where $\mathbf{x_j}$ is the data vector in the data set X, **u** is the mean vector of set X and **Σ** is the covariance matrix. The Mahalanobis distance gives us a measure of dissimilarity between a data vector and the mean vector of data set X. The extreme feature vector $\mathbf{x}_{\text{extreme}}$ is the data vector that has the maximum Mahalanobis distance from the mean vector.

Next, a matrix A is computed, which equals (n-i) multiplied by the sample covariance matrix. We can compute the matrix A from a mean-corrected data matrix D which is determined by

$$D = \left[\left(x_1 - u\right) \quad \left(x_2 - u\right) \quad \ldots \quad \left(x_{n-i+1} - u\right)\right]^T$$

Matrix A can then by calculated as follows

$$A = D^T D$$

It should be noted that we can also use matrix A to determine the distance from the extreme data vector to the mean vector. The distance using matrix A is given by

$$d_{extreme} = \left(x_{extreme} - u\right)^T A^{-1}\left(x_{extreme} - u\right)$$

where A^{-1} is the inverse of matrix A. The extreme data vector also maximizes the distance from the mean vector when matrix A is used. This distance is then used to compute the Wilk’s statistics w for the current iteration

$$w = 1 - \frac{n-i+1}{n-i} d_{extreme}$$

where $d_{extreme}$ is the distance between the extreme data vector and the mean vector using matrix A as the scaling matrix.

To determine the whether the chosen extreme data vector is an actual outlier, we need to compute the critical value λ for the current iteration, which is given as follows

$$\lambda = \left(1 + \frac{m}{n-m-i} F_{m,n-m-i,[1-\frac{\alpha}{n-i+1}]} \right)^{-1}$$

where $F_{m,n-m-i,[1-\alpha/(n-i+1)]}$ is the percentile value for an F-distribution with m degrees of freedom for the numerator, (n-m-i) degrees of freedom for the denominator, and a right hand tail area of $[1-\alpha/(n-i+1)]$. If the distance between the outlier candidate and the mean vector is larger than the critical value λ, an outlier is detected and removed from data set X. Otherwise, the outlier candidate is directly removed from X. After n_u iterations, a set of outliers is found and the algorithm terminates. The Wilk's multivariate many-outlier procedure is computationally more expensive than the GESE many-outlier procedure, but it performs well when multi-dimensional data is involved. It is a better choice for automatic anomaly detection because the user only needs to specify the parameters α and n_u and no union operation is involved.

3.3.3 Determination of Types of Days by Clustering

Residential energy consumption shows distinctive patterns during different types of days, such as weekdays, weekends, and holidays. As mentioned in the previous subsection, regular energy consumption patterns for weekends may be captured as outliers since there are always more weekdays than weekends. Hence, it is necessary to categorize energy data based on the types of days. Experienced building managers can manually put Green Button data into the groups based on his knowledge of energy-related routines of the building. However, this task can be challenging for residential customers. Furthermore, the energy consumption patterns of different residential premises are sometimes significantly different from one another. This motivates us to develop automatic clustering techniques that classify energy consumption data into several categories. Such algorithms should determine the number and types of data categories for a residential premise without customer intervention. Specifically, cluster analysis helps to resolve the following problems. First, the techniques introduced in this section automatically choose the number of categories, such as weekdays and weekends, for classification. Second, days with similar energy consumption profiles are organized into the same category. Finally, the impact of seasonal variations should be minimized. For instance, we prefer to classify all data of Wednesdays into a single category instead of multiple ones.

Essentially, the purpose of cluster analysis is to label the days of a yeas that are used in Green Button data so that energy consumption data can be organized based on these labels. Therefore, we use coarse-grained data instead of the raw data sequence. Customers may choose from the following ways of generating data of coarse granularity for cluster analysis: (1) Daily energy consumption data. (2) Daily Energy consumption during the same time interval. (3) Daily Energy consumption of selected time intervals organized as a vector. For the first two options, the feature extraction techniques we discussed in the previous section can still be applied, changing data scalars into feature vectors. When using energy consumption data from multiple time intervals of a day, we typically select the peak demand, average energy consumption, or the total energy consumption since data points have already been organized into vectors. In this section, we introduce two clustering techniques that can be incorporated with the outlier detection algorithms discussed in the previous section. Both techniques are tested using simulated Green Button data.

3.3.3.1 Modified Agglomerative Hierarchical Clustering

As a commonly used technique, agglomerative hierarchical clustering [71] begins with clusters of individual data scalars or vectors. This technique then merges nearest clusters until only a single cluster remains. For our problem, the data should be partially clustered because we want to obtain data groups containing days with similar energy consumption patterns instead of a single cluster. We modified the agglomerative hierarchical clustering algorithms by introducing termination rules, allowing days with similar energy consumption patterns to be organized into the same cluster while those with distinct patterns to be put into different groups. It should be noted that date and day-of-week information in Green Button data is maintained but not actually used in the clustering process. These labels help customers to better understand their macroscopic power consumption patterns. For instance, a customer may learn about which day of week presents distinctive power consumption patterns and adjust his/her behavior accordingly.

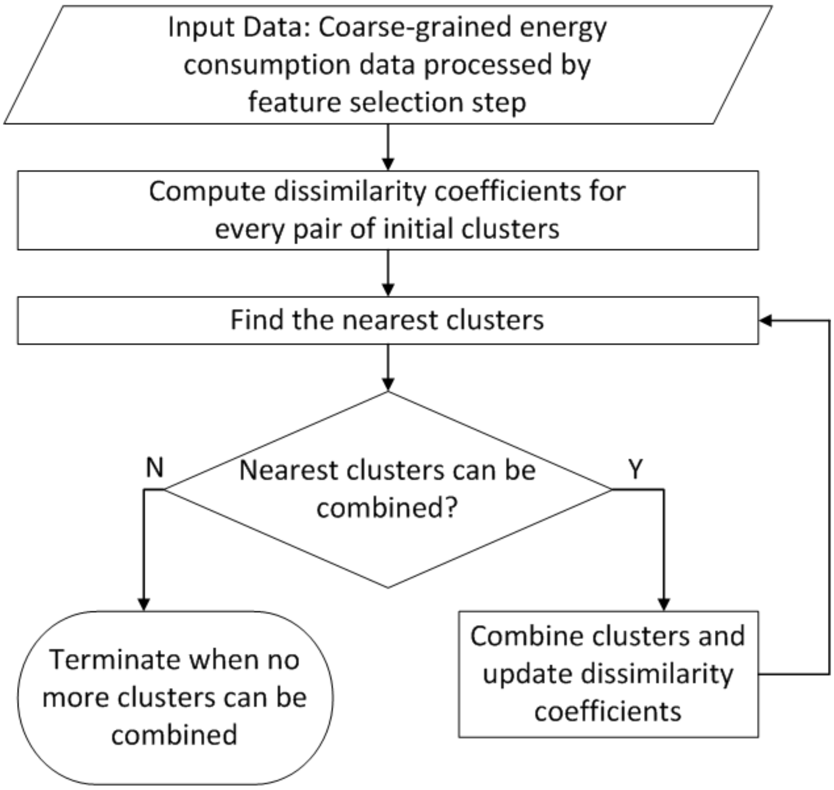

Fig. 25: Flowchart of the modified agglomerative hierarchical clustering

A dissimilarity coefficient based on the average linkage method proposed by Kaufman and Rousseeuw [72] is employed to quantify the extent to which two clusters are different from one another. The average linkage method defines the dissimilarity coefficient between two clusters C_i and C_j as the average distance between every pair of feature vectors, where one vector of the pair comes from C_i and the other data vector is taken from C_j. Therefore, the dissimilarity coefficient we use can be formally defined as

$$d\left(C_i, C_j\right) = \frac{1}{n_i \cdot n_j} \sum_{\substack{x \in C_i \\ y \in C_j}} d\left(x, y\right)$$

where n_i is the number of feature vectors in C_i, n_j is the number of feature vectors in C_j, and d(**x**, **y**) is the distance between the pair of feature vectors. In our modified agglomerative hierarchical clustering algorithm, Euclidean distance is used. For two feature vector **x** and **y**, d(**x**, **y**) is defined as

$$d\left(x, y\right) = \sqrt{\sum_{i=1}^{n_{features}} \left(x_i - y_i\right)^2}$$

where x_i and y_i are the ith components of vector **x** and **y**, respectively.

The algorithm iterates until the termination condition is satisfied. At the beginning of an iteration, the distance between each pair of clusters must be evaluated. The nearest clusters are then identified. To determine whether the two clusters should be merged or not, we use the termination rule proposed by Duda and Hart [47]

$$z > \frac{1 - \left(2/\pi \cdot n_{features} \right) - \left(\left(SS_i + SS_j\right) / SS_{i \cup j} \right)}{\sqrt{\left(2\left[1 - 8/\left(\pi^2 \cdot n_{features}\right) \right] \right) / \left(n_i + n_j\right) \cdot n_{features}}}$$

where z is a critical value from a standard normal distribution, $n_{features}$ represents the number of features used, n_i and n_j are the number of feature vectors in clusters C_i and C_j, respectively. The SS_i and SS_j terms are the sums of squared distances from member feature vectors of clusters C_i and C_j to their corresponding mean feature vectors. SS_{iUj} represents the sum of squared distances between feature vectors in cluster C_i as well as C_j and the new mean feature vector computed using all feature vectors from C_i and C_j. The general formula for SS_i, SS_j, and SS_{iUj} is as follows

$$SS = \sum_{x \in C} \left(x - u\right)^T \left(x - u\right), \text{ where } u = \frac{1}{n} \sum_{x \in C} x$$

Immediately after two clusters are combined, the dissimilarity coefficient between the new cluster and all the other clusters can be updated using the following equation

$$d\left(C_i \cup C_j, C_k\right) = \frac{n_i}{n_i + n_j} d\left(C_i, C_k\right) + \frac{n_j}{n_i + n_j} d\left(C_j, C_k\right)$$

Using this equation, we only need to compute all the dissimilarity coefficients once at the very beginning of the modified agglomerative hierarchical clustering algorithm, saving the cost of expensive computation of Euclidean distance. The clustering result is then used to group the data for outlier detection.

3.3.3.2 Clustering using Canonical Variables (CVs) and Linear Discriminant Analysis

This approach has better clustering performance at the expense of intensive computation [69]. It usually results in a slightly smaller number of clusters. The data we use is still coarse-grained feature scalars or vectors. The data is organized as an m-by-n matrix Xi, where m is the number of samples and n the number of features. Note that the day-of-week labels in Green Button data must be employed so as to divide the data into groups. Suppose we have K groups of data and each group is placed in consecutive rows of matrix X. The matrix X is first passed through canonical variable analysis (CVA), which projects the original data onto new axes called canonical variables. The CVA method maximizes interclass separation while minimizing intra-class variance. Interclass variance can be characterized by between-group covariance matrix C_B, which is defined as

$$C_B = \frac{1}{K-1}\sum_{i=1}^{K} n_i \left(u_i - u\right)\left(u_i - u\right)^T$$

where u is the mean vector computed from all the samples and $\mathbf{u_i}$ the mean vector of the ith group. The intra-class variance is characterized by within-group covariance matrix C_W, which is given by

$$C_W = \frac{1}{N-K}\sum_{i=1}^{K}\sum_{j=1}^{n_i}\left(x_{ij} - u_i\right)\left(x_{ij} - u_i\right)^T$$

where x_{ij} is the jth sample of the ith group. Using the following equation, we first found the eigenvalues λ and eigenvectors W. The columns of W are the directions of axes onto which we will project the data. With K groups of data, there should be K-1 CVs. The CVs are defined as

$$CV = X_C W$$

where X_c is the mean-centered data matrix obtained using every row of matrix X.

For every group of date, evaluate the linear discriminant functions for each feature vector and the data is categorized into the class whose linear dicriminant function is the largest. The linear discriminant function is given as follows

$$F_i\left(CV\right) = \log\left(\pi_i\right) - \frac{1}{2}\left(CV - CV_i\right)^T C_{W,CV}^{-1}\left(CV - CV_i\right) + \log\left|C_{W,CV}\right|$$

where i=1, 2, …, K is the group number, π_i is the prior probability and equals 1/K, CV is the canonical variable of the sample data to be classified, CV_i is the average of CVs for group i, and $C_{w,CV}$ is the covariance matrix of the CVs. Although we can obtain fewer clusters using this approach, it should be pointed out the algorithm does not work if the matrix C_W is singular [73].

3.3.4 Evaluation Results

The anomaly detection schemes described in previous subsections are evaluated using our simulated data set. Since the REDD [9] data set is not developed for anomaly detection, we modify the simulated Green Button data set to introduce some anomalies by changing some power consumption readings of the refrigerator. Some readings are changed into values that are smaller than the minimum ones of the same day, while some other readings are replaced by values that are larger than the maximum ones of the same day. Data of a whole month is used, and our evaluation is performed in two steps. Both agglomerative hierarchical clustering and CV-based clustering are tested. Both algorithms put the days into two groups, namely weekdays and weekends. Fig. 26 shows the daily power consumption of the days of a week, and both clustering algorithms correctly classify Sunday and Saturday into weekends and the rest into weekdays. This figure also shows that a number of power consumption readings of weekends (and probably holidays) may be considered as anomalies and clustering is necessary.

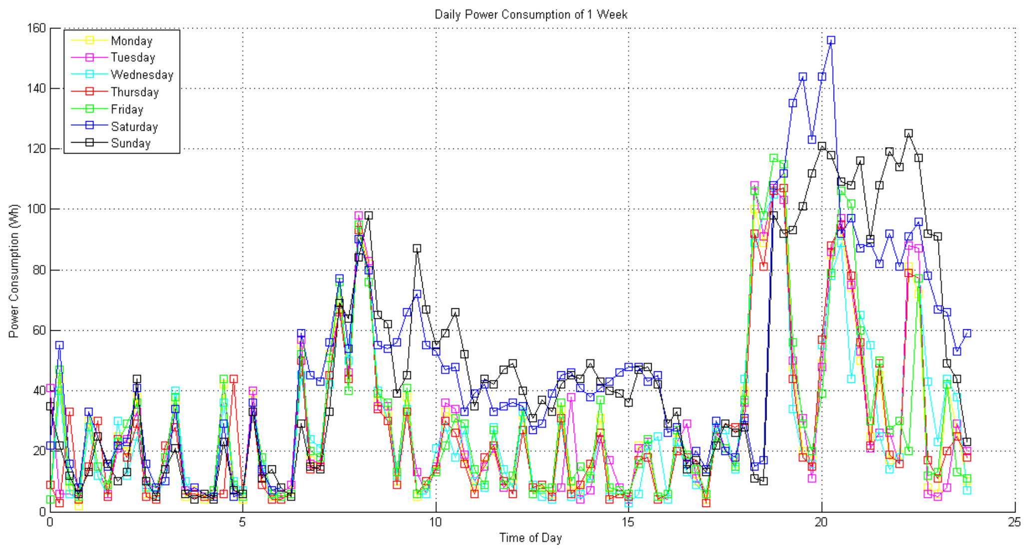

Fig. 26: Daily power consumption of Home 5 in REDD [9] data set in 1 week.

After clustering, each day is labeled as either "weekdays" or "weekends". The two groups of data are fed to outlier detection algorithms separately. Both GESD and Wilk's many-outlier procedures are able to identify all the outliers we introduced. However, it should be noted that Wilk's algorithm is more effective in processing vector-based data, whereas GESD algorithm requires union operation and generates more outliers. The weekdays of 1 week of Home 5 in the REDD [9] data set with manually injected outliers are shown in Fig. 27.

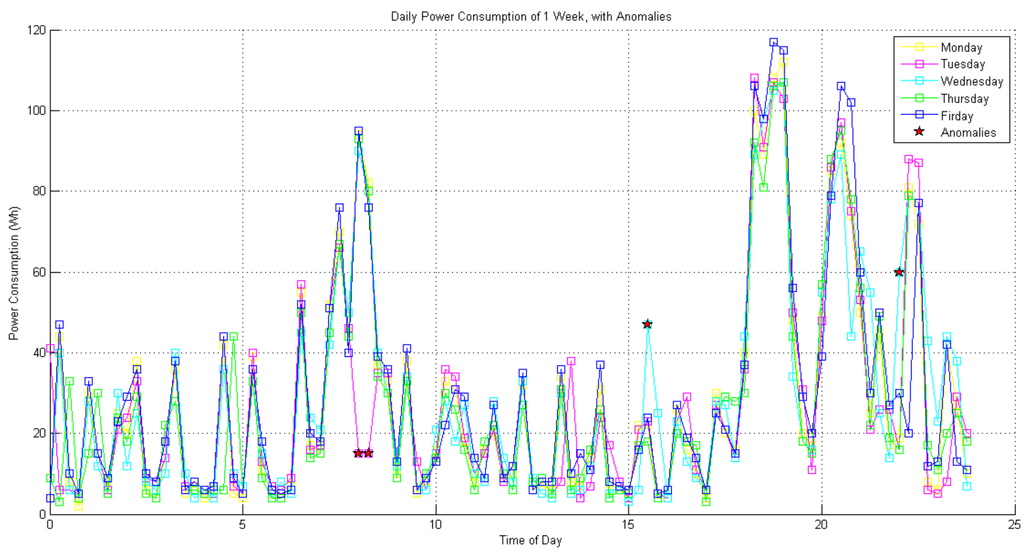

Fig. 27: Daily power consumption of the weekdays in one week. Outliers are introduced by modifying some power consumption readings of the refrigerator.

3.4 Possible Organization of Practical Anomaly Detection Systems

With individual modules for anomaly detection, we are now able to construct an anomaly detection system that detects outliers from Green Button data. The task of anomaly detection can be done in two steps. First, an appropriate clustering technique is adopted to classify calendar days with Green Button data into several types of days with similar energy consumption patterns. The clustering techniques discussed in this section may also be used to compare energy consumption profiles of different households. Although the approach of comparing energy profiles between neighboring households is generally not available to individual customers, this approach may still be regarded as another way of anomaly detection. For instance, building managers may be interested in the comparisons between the power consumption patterns of various housing units so as to provide general energy-saving guidelines for residents.

The ensuing step of outlier detection can be incorporated with the clustering step in two ways. The data fed to outlier detection algorithms is grouped according to the labels generated by clustering techniques. Alternatively, we can also use outlier detection before clustering. This allows us to first screen out outliers in the set of coarse data. Most outliers identified at this stage are associated with vacations, holidays or special events. The elimination of these outliers allows us to cluster and analyze fewer days of data, and we can thus choose smaller values for α and n_u. For pre-screening purpose, the GESD many-outlier procedure is usually used. For the outlier detection step, we can implement both GESD and Wilk’s algorithms. The anomaly detection system will then choose an appropriate algorithm based on the features specified by the user.

4. Discussion and Future Work

In this report, we study the feasibility of energy disaggregation and anomaly detection using Green Button data. Representative techniques are implemented and evaluated on simulated Green Button data generated from the REDD [9] data set. In order to make energy disaggregation practical for residential applications, there are two major challenges that need to be addressed. The low sampling frequency of Green Button data makes it hard to detect frequent state transitions of appliances. Auxiliary information collected by extra sensors or extra smart meter readings gathered locally by inexpensive hardware modules [53] can help improve the accuracy of energy disaggregation. In the future, we will look into the solvability issue of energy disaggregation problems. In particular, we plan to further investigate the degree to which appliances can be distinguished using different type of features such as energy consumption readings and signals from motion sensors. Knowing the theoretical limits [74] of a given NILM solutions is important because this will allow us to select the appropriate set of features in terms of computational efficiency and disaggregation performance. A possible theoretical formulation of the solvability of energy disaggregation problem is to use the receiver operating characteristic (ROC) curves [75]. On the other hand, we note that only a small number of appliances are used in the evaluation of existing algorithms. As the number of appliances increases, it becomes increasingly difficult to distinguish the appliances due to the overlapping of the distributions of features in the feature space. This issue also requires that we attain better understanding of the theoretical limits of NILM schemes.

For anomaly detection, we show that preprocessing Green Button data using clustering techniques improves the accuracy of outlier detection. It should be noted that statistical techniques for outlier detection are not able to differentiate patterns related to ad hoc activities from those related to real anomalies. User intervention is required in anomaly detection systems, and some researchers have begun to devise the data representation and management solutions to support data-intensive residential energy management systems [76]. The performance requirements we identify, namely the false positive ratio and true positive ratio, are important performance metrics because the primary objective of anomaly detection is to help users focus on real issues. To further assist customers in recalling energy consumption activities, researchers develop activity detection and annotation systems combining smart meter and smart phones [77]. To accurately identify anomalies, consultation with experts may be required, which also raises the privacy issues. Research on privacy preservation, which protects sensitive energy consumption data and other necessary information for anomaly diagnoses, such as records of user activities, also makes steady progress [78]. Finally, we note that anomaly detection techniques are capable of handling data obtained at different sampling frequency, and the selection of appropriate sampling frequency can help us preserve customer privacy [79].

Finally, we notice that publicly accessible data sets only provide data of a limited number of appliances with ground truth and some appliances are not actually turned on during the entire period of monitoring. This motivates us to develop our own data collection platform in a controlled environment and develop scenarios drawn from the activities of real residential customers. Our effort to develop the test bed is described in the next part of the report. To make residential data analysis practical, further research is required to enhance the Green Button data standard to support auxiliary information, monitor a larger number of appliances for longer periods of time in real residential houses, and conduct comprehensive study of low-frequency features of appliances.

References

[1] http://greenbuttondata.org/developers/

[2] Najmeddine, H.; El Khamlichi Drissi, K.; Pasquier, C.; Faure, C.; Kerroum, K.; Diop, A.; Jouannet, T.; Michou, M., "State of art on load monitoring methods," Power and Energy Conference, 2008. PECon 2008. IEEE 2nd International , pp.1256-1258, 1-3 Dec, 2008.

[3] Zoha, A.; Gluhak, A.; Imran, M.A.; Rajasegarar, S., "Non-Intrusive Load Monitoring Approaches for Disaggregated Energy Sensing: A Survey," Sensors 2012, vol. 12, pp.16838-16866. 2012.

[4] Kolter, J. Z.; Batra, S.; Ng, A. Y., "Energy disaggregation via discriminative sparse coding," In Advances in Neural Information Processing Systems, pp.1153-1161. 2010.

[5] http://collaborate.nist.gov/twiki-sggrid/bin/view/SmartGrid/GreenButtonInitiative

[6] Yaqub, R.; Hamid, B.; ul Asar, A., "Appliance performance monitoring and warranty alert system," Open Source Systems and Technologies (ICOSST), 2013 International Conference on , pp.13-17, 16-18 Dec, 2013.

[7] Linda, O.; Wijayasekara, D.; Manic, M.; Rieger, C., "Computational intelligence based anomaly detection for Building Energy Management Systems," Resilient Control Systems (ISRCS), 2012 5th International Symposium on , pp.77-82, 14-16 Aug, 2012.

[8] Wijayasekara, D.; Linda, O.; Manic, M.; Rieger, C., "Mining Building Energy Management System Data Using Fuzzy Anomaly Detection and Linguistic Descriptions," Industrial Informatics, IEEE Transactions on , vol.PP, no.99, pp.1-12, June 2014.

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

[10] http://energy.gov/downloads/green-button-sample-data-nstar-monthly

[11] http://energy.gov/downloads/green-button-sample-texas

[12] http://energy.gov/downloads/green-button-sample-data-pge

[13] http://www.dlink.com/us/en/home-solutions/share/network-attached-storage/dsp-w215

[14] Barker, S.; Mishra, A.; Irwin, D.; Cecchet, E.; Shenoy, P.; Albrecht, J., "Smart*: An open data set and tools for enabling research in sustainable homes," In Workshop on Data Mining Applications in Sustainability (SIGKDD), Beijing, China, August 2012.

[15] Anderson, K.; Ocneanu, A.; Benitez, D.; Carlson, D.; Rowe, A.; Berges, M., "BLUED: a fully labeled public dataset for Event-Based Non-Intrusive load monitoring research," In Proceedings of the 2nd KDD Workshop on Data Mining Applications in Sustainability, pp. 12-16, Beijing, China, August 2012.

[16] Reinhardt, A.; Baumann, P.; Burgstahler, D.; Hollick, M.; Chonov, H.; Werner, M.; Steinmetz, R., "On the accuracy of appliance identification based on distributed load metering data," Sustainable Internet and ICT for Sustainability (SustainIT), 2012 , pp.1-9, 4-5 Oct, 2012.

[17] Makonin, S.; Popowich, F.; Bartram, L.; Gill, B.; Bajic, I. V., "AMPds: A public dataset for load disaggregation and eco-feedback research," In Electrical Power & Energy Conference (EPEC), 2013 IEEE, pp. 1-6, August 2013.

[18] U.S. Energy Information Administration (EIA), "Annual Energy Outlook 2014," U.S. Department of Energy, May 2014.

[19] https://www.pplelectric.com/at-your-service/electric-rates-and-rules/time-of-use-option.aspx

[20] Abrahamse, W.; Steg, L.; Vlek, C.; Rothengatter, T., "The effect of tailored information, goal setting, and tailored feedback on household energy use, energy-related behaviors, and behavioral antecedents," Journal of Environmental Psychology, vol. 27(4), pp. 265-276, December 2007.

[21] Darby, S., "The effectiveness of feedback on energy consumption," A Review for DEFRA of the Literature on Metering, Billing and direct Displays, no. 486, 2006.

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

[23] Fisher, D.H., "Recent Advances in AI for Computational Sustainability," Intelligent Systems, IEEE , vol.27, no.4, pp.75-79, July-Aug. 2012.

[24] Baranski, M.; Voss, J., "Genetic algorithm for pattern detection in NIALM systems," Systems, Man and Cybernetics, 2004 IEEE International Conference on , vol.4, pp.3462-3468, 10-13 Oct, 2004.

[25] Suzuki, K.; Inagaki, S.; Suzuki, T.; Nakamura, H.; Ito, K., "Nonintrusive appliance load monitoring based on integer programming," SICE Annual Conference, 2008, pp.2742-2747, 20-22 Aug, 2008.

[26] Schoofs, A.; Guerrieri, A.; Delaney, D. T.; O'Hare, G. M. P.; Ruzzelli, A. G., "ANNOT: Automated Electricity Data Annotation Using Wireless Sensor Networks," Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on, pp.1-9, 21-25 June, 2010.

[27] Weiss, M.; Helfenstein, A.; Mattern, F.; Staake, T., "Leveraging smart meter data to recognize home appliances," Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on, pp.190-197, 19-23 March, 2012.

[28] http://www.wholesalesolar.com/StartHere/HowtoSaveEnergy/PowerTable.html

[29] Laughman, C.; Kwangduk Lee; Cox, R.; Shaw, S.; Leeb, S.; Norford, L.; Armstrong, P., "Power signature analysis," Power and Energy Magazine, IEEE, vol.1, no.2, pp.56-63, Mar-Apr 2003.

[30] Srinivasan, D.; Ng, W. S.; Liew, A.C., "Neural-network-based signature recognition for harmonic source identification," Power Delivery, IEEE Transactions on, vol.21, no.1, pp.398-405, Jan 2006.

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

[32] Chan, W. L.; So, A. T. P.; Lai, L. L., "Harmonics load signature recognition by wavelets transforms," Electric Utility Deregulation and Restructuring and Power Technologies, 2000. Proceedings. DRPT 2000. International Conference on, pp.666-671, 2000.

[33] http://w1.weather.gov/obhistory/KABE.html

[34] Ghahramani, Z.; Jordan, M. I., "Factorial hidden Markov models," Machine learning, vol. 29(2-3), pp. 245-273, Nov 1997.

[35] Rabiner, L., "A tutorial on hidden Markov models and selected applications in speech recognition," Proceedings of the IEEE, vol.77, no.2, pp.257-286, Feb 1989.

[36] Zeifman, M.; Roth, K., "Nonintrusive appliance load monitoring: Review and outlook," Consumer Electronics, IEEE Transactions on, vol.57, no.1, pp.76-84, February 2011.

[37] Kim, H.; Marwah, M.; Arlitt, M.; Lyon, G.; Han, J., "Unsupervised Disaggregation of Low Frequency Power Measurements," Proceedings of the 2011 SIAM International Conference on Data Mining, pp. 747-758, 2011.

[38] Shao, H.; Marwah, M.; Ramakrishnan, N., "A Temporal Motif Mining Approach to Unsupervised Energy Disaggregation: Applications to Residential and Commercial Buildings." In Twenty-Seventh AAAI Conference on Artificial Intelligence, June 2013, Available at http://www.aaai.org/ocs/index.php/AAAI/AAAI13/paper/view/6420.

[39] Shaw, S. R., "System identification techniques and modeling for nonintrusive load diagnostics," Doctoral dissertation, Massachusetts Institute of Technology, 2000.

[40] Saitoh, T.; Osaki, T.; Konishi, R.; Sugahara, K., "Current sensor based home appliance and state of appliance recognition," SICE J. Contr. Meas. Syst. Integrat., vol. 3, pp. 86-93, 2010.

[41] Liangxiao Jiang; Zhihua Cai; Dianhong Wang; Siwei Jiang, "Survey of Improving K-Nearest-Neighbor for Classification," Fuzzy Systems and Knowledge Discovery, 2007, Fourth International Conference on, vol.1, pp.679-683, 24-27 Aug, 2007.

[42] Angiulli, F., "Fast Nearest Neighbor Condensation for Large Data Sets Classification," Knowledge and Data Engineering, IEEE Transactions on, vol.19, no.11, pp.1450-1464, Nov 2007.

[43] Garcia, S.; Derrac, J.; Cano, J.R.; Herrera, F., "Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.34, no.3, pp.417-435, March 2012.

[44] Patel, S. N.; Robertson, T.; Kientz, J. A.; Reynolds, M. S.; Abowd, G. D., "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 (UbiComp 2007), pp. 271-288, Innsbruck, Austria, September 16-19, 2007.

[45] Gu-yuan Lin; Shih-chiang Lee; Hsu, J.Y.-J.; Wan-rong Jih, "Applying power meters for appliance recognition on the electric panel," Industrial Electronics and Applications (ICIEA), 2010 the 5th IEEE Conference on, pp. 2254-2259, 15-17 June, 2010.

[46] Burges, C. J., "A tutorial on support vector machines for pattern recognition," Data mining and knowledge discovery, vol. 2, no. 2, pp. 121-167, 1998.

[47] Duda, R. O.; Hart, P. E.; Stork, D. G., "Pattern classification 2nd edition," John Wiley & Sons, 2001.

[48] Decoste, D.; Schölkopf, B., "Training invariant support vector machines," Machine Learning, vol. 46, pp. 161-190, 2002.

[49] Wang, B.; Dong, H.; Boedihardjo, A. P.; Chen, F.; Lu, C. T., "A Hierarchical Probabilistic Model for Low Sample Rate Home-Use Energy Disaggregation," In SIAM International Conference on Data Mining (SDM), pp. 704-712, 2013.

[50] Froehlich, J.; Larson, E.; Gupta, S.; Cohn, G.; Reynolds, M.S.; Patel, S.N., "Disaggregated End-Use Energy Sensing for the Smart Grid," Pervasive Computing, IEEE, vol.10, no.1, pp.28-39, Jan.-March 2011.

[51] Zoha, A.; Gluhak, A.; Nati, M.; Imran, M.A., "Low-power appliance monitoring using Factorial Hidden Markov Models," Intelligent Sensors, Sensor Networks and Information Processing, 2013 IEEE Eighth International Conference on, pp. 527-532, 2-5 April, 2013.

[52] Phillips, D.E.; Rui Tan; Moazzami, M.; Guoliang Xing; Jinzhu Chen; Yau, D.K.Y., "Supero: A sensor system for unsupervised residential power usage monitoring," Pervasive Computing and Communications (PerCom), 2013 IEEE International Conference on, pp. 66-75, 18-22 March, 2013.

[53]Carrie Armel, K.; Gupta, A.; Shrimali, G.; Albert, A., "Is disaggregation the holy grail of energy efficiency? The case of electricity," Energy Policy, vol. 52, pp. 213-234, 2013.

[54] Chawla, N. V.; Japkowicz, N.; Kotcz, A., "Editorial: special issue on learning from imbalanced data sets," ACM Sigkdd Explorations Newsletter, vol. 6, pp. 1-6, June 2004.

[55] Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; Kegelmeyer, W. P., "SMOTE: Synthetic Minority Over-sampling Technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321-357, 2002.

[56] Kazmi, A. H.; O'grady, M. J.; Delaney, D. T.; Ruzzelli, A. G.; O'hare, G. M., "A Review of Wireless-Sensor-Network-Enabled Building Energy Management Systems," ACM Transactions on Sensor Networks (TOSN), vol. 10(4), Article 66, June 2014.

[57] Fischer, J. E.; Ramchurn, S. D.; Osborne, M.; Parson, O.; Huynh, T. D.; Alam, M.; Pantidi, N.; Moran, S.;

Bachour, K.; Reece, S.; Costanza, E.; Rodden, T.; Jennings, N. R., "Recommending energy tariffs and load shifting based on smart household usage profiling," In Proceedings of the 2013 international conference on Intelligent user interfaces (IUI '13), ACM, pp. 383-394, New York, USA, March 2013.

[58] Eibl, G.; Engel, D., "Influence of data granularity on nonintrusive appliance load monitoring," In Proceedings of the 2nd ACM workshop on Information hiding and multimedia security, pp. 147-151, ACM, June 2014.

[59] Linda, O.; Wijayasekara, D.; Manic, M.; Rieger, C., "Computational intelligence based anomaly detection for Building Energy Management Systems," Resilient Control Systems (ISRCS), 2012 5th International Symposium on, pp. 77-82, 14-16 Aug, 2012.

[60] Chandola, V.; Banerjee, A.; Kumar, V., "Anomaly detection: A survey," ACM Computing Surveys (CSUR), vol. 41(3), Article 15, July 2009.

[61] Hodge, V. J.; Austin, J., "A survey of outlier detection methodologies," Artificial Intelligence Review, vol. 22(2), pp. 85-126, Oct. 2004.

[62] Katipamula, S.; Brambley, M. R., "Review article: methods for fault detection, diagnostics, and prognostics for building systems—a review, Part I," HVAC&R Research, vol. 11(1), pp. 3-25, 2005.

[63] Katipamula, S.; Pratt, R. G.; Chassin, D. P.; Taylor, Z. T.; Gowri, K.; Brambley, M. R., "Automated fault detection and diagnostics for outdoor-air ventilation systems and economizers: Methodology and results from field testing," Transactions-American Society Of Heating Refrigerating And Air Conditioning Engineers, vol. 105, pp. 555-567, 1999.

[64] Hyvarnen, J., "IEA ANNEX 25, building optimization and fault diagnosis source book," International Energy Agency, Paris, France, 1995.

[65] Salsbury, T.; Diamond, R., "Performance validation and energy analysis of HVAC systems using simulation," Energy and buildings, vol. 32(1), pp. 5-17, 2000.

[66] Dodier, R. F.; Kreider, J. F., "Detecting whole building energy problems," ASHRAE Transactions, vol. 105(1), pp. 579-589, 1999.

[67] Chen, C.; Cook, D. J.; Crandall, A. S., "The user side of sustainability: Modeling behavior and energy usage in the home," Pervasive and Mobile Computing, vol. 9, no. 1, pp. 161-175, 2013.

[68] Räsänen, T.; Ruuskanen, J.; Kolehmainen, M., "Reducing energy consumption by using self-organizing maps to create more personalized electricity use information," Applied Energy, vol. 85, no. 9, pp. 830-840, 2008.

[69] Xiaoli Li; Bowers, C.P.; Schnier, T., "Classification of Energy Consumption in Buildings With Outlier Detection," Industrial Electronics, IEEE Transactions on, vol.57, no.11, pp. 3639-3644, Nov. 2010.

[70] Seem, J. E., "Using intelligent data analysis to detect abnormal energy consumption in buildings," Energy and Buildings, vol. 39, no. 1, pp. 52-58, 2007.

[71] Seem, J. E., "Pattern recognition algorithm for determining days of the week with similar energy consumption profiles," Energy and Buildings, vol. 37, no. 2, pp. 127-139, 2005.

[72] Kaufman, L.; Rousseeuw, P. J., "Finding groups in data: an introduction to cluster analysis," Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons Inc., New York, 2009.

[73] Nørgaard, L.; Bro, R.; Westad, F.; Engelsen, S. B., "A modification of canonical variates analysis to handle highly collinear multivariate data," Journal of Chemometrics, vol. 20, no. 8-10, pp. 425-435, 2006.

[74] Dong, R.; Ratliff, L.; Ohlsson, H.; Sastry, S. S., "Fundamental limits of nonintrusive load monitoring," In Proceedings of the 3rd international conference on High confidence networked systems, pp. 11-18, ACM, April 2014.

[75] Fawcett, T., "An introduction to ROC analysis," Pattern recognition letters, vol. 27(8), pp. 861-874, June

2006.
[76] Batra, N.; Kelly, J.; Parson, O.; Dutta, H.; Knottenbelt, W.; Rogers, A.; Singh, A.; Srivastava, M., "NILMTK: An Open Source Toolkit for Non-intrusive Load Monitoring," In Proceedings of the 5th international conference on Future energy systems (e-Energy '14), pp. 265-276, ACM, New York, 2014.
[77] Saha, M.; Thakur, S.; Singh, A.; Agarwal, Y., "EnergyLens: Combining Smartphones with Electricity Meter for Accurate Activity Detection and User Annotation," In Proceedings of the 5th international conference on Future energy systems (e-Energy '14), pp. 289-300, ACM, New York, 2014.
[78] Chu, C. K.; Liu, J. K.; Wong, J. W.; Zhao, Y.; Zhou, J., "Privacy-preserving smart metering with regional statistics and personal enquiry services," In Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security (ASIA CCS '13), pp. 369-380, ACM, May 2013.
[79] Tudor, V.; Almgren, M.; Papatriantafilou, M., "Analysis of the impact of data granularity on privacy for the smart grid," In Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society (WPES '13), pp. 61-70, ACM, November 2013.