A Mobile System for Annotation of Home Energy Data

Sami Rollins
University of San Francisco
srollins@cs.usfca.edu

Nilanjan Banerjee
University of Maryland,
Baltimore County
nilanb@umbc.edu

Lazeeb Choudhury
University of San Francisco
lfchoudhury@usfca.edu

David Lachut
University of Maryland,
Baltimore County
dlachut1@umbc.edu

ABSTRACT

Home energy management is increasingly important. Though there are a plethora of tools for aiding home energy management, few provide concrete suggestions for helping users to manage energy demand. A key component in developing automated energy management schemes is drawing a connection between energy usage and a user's context. Existing approaches either rely on users to annotate data after the fact, or rely on intrusive and costly sensing systems deployed in the home. This work presents a system for collecting in situ annotations using a mobile application coupled with an existing home energy measurement infrastructure. We use a novel power profiling approach to determine when appliances transition from an idle to active state and aggressively prompt users to provide annotations of their current context. In a five-week study, we were able to collect an average of over 2 annotations per day and users provided a wide range of annotations, however the overall response rate was lower than expected leading to a sparse data set. We conclude by examining the utility of sparse energy data annotations. We first demonstrate that our data set confirms that end user annotation is necessary—a completely automated activity inferencing scheme is implausible. We further demonstrate that sparse annotation data can be used to predict a user's activity with an accuracy of more than 50% in hours where our data set contains an annotation from the user. We finally consider the feasibility of using annotations to predict a user's energy needs.

1. INTRODUCTION

A key component of creating a sustainable world is managing energy in the home. Recent years have seen a marked increase in the number of tools that end users can employ to monitor their energy consumption, but most tools simply provide access to raw numbers. Users can, for example, install energy meters that provide readouts of the watt hours consumed by a particular appliance, however this basic information can be difficult to interpret for a user who simply wants to know how to reduce an energy bill or be friendlier to the environment.

The overarching goal of our work is to provide users with targeted suggestions for reducing home energy consumption, and an important element of our approach is understanding not only how much energy is consumed but *why* a home exhibits a particular energy profile at a given time. Understanding the connection between energy consumed and the activities a resident performs is critical both for developing intelligent energy management algorithms on the system side and presenting helpful information on the user side. Collecting or deriving these annotations, however, is a difficult problem. Existing approaches typically rely on the user to annotate an energy consumption timeline after the fact using a visualization tool—relying on the user to remember what he or she was doing at a particular point in the past [1]—or can require costly and intrusive sensing systems to identify user behavior.

The goal of this work is to collect annotations *in situ* using a mobile application integrated with an energy measurement system used broadly for energy management in the home. Our system requires only the sensors necessary for our Green Homes infrastructure [2]. Green Homes collects energy usage data of several appliances in the home and provides a visualization and control framework. Our energy annotation component sends a push notification to a user's smartphone when the user has likely begun a new activity and the user may reply with an annotation of his/her current context. In this work we explore both annotation collection as well as the utility of the annotations provided by 5 users of our system. There are three primary contributions of this work:

Our first contribution is an approach for using the appliance power signatures we already collect in Green Homes for determining when a change is sufficiently appreciable to suggest that the user may be performing a different or new activity. Many appliances do not have a clear on/off state and determining when the raw power draw indicates that the device has transitioned into a more active state is not straightforward. Our approach uses the DBSCAN clustering algorithm to identify a unique power profile for each individual device. Using this approach, we were able to generate usable profiles for 31 of 39 devices across 5 homes participating in a five-week deployment. Of the remaining devices, two were never used during the experiment period and three were always-on and remained in the same power state.

Our second contribution outlines the results of a five-week deployment in 5 homes. Using the profiles generated by our algorithm, we notified the user anytime any device in the home transitioned into a higher power state, and we also asked users to annotate any time they thought they were performing a relevant activity. We were able to collect an average of over 2 annotations per day, and in one day we saw 13 annotations from one of our subjects. Users provided a wide range of annotations across different times of day and representing different types of activities, however the overall response rate was lower than expected leading to a fairly sparse data set.

Based on the results of our deployment, our third contribution considers the utility of our annotation data. Our analysis considers how we may use annotations to understand a user's energy requirements. We first confirm that manual annotations are necessary—common features such as time of day and power consumption cannot be used to automatically derive annotations using unsupervised techniques. We next demonstrate that using sparse annotation data we can, in most cases, predict a user's activity with an accuracy of more than 50% in hours where our data set contains an annotation from the user. Finally, we explore the feasibility of using annotations to augment a prediction algorithm that identifies the hourly power consumption profile for a user's home.

2. SYSTEM DESIGN

Accurate annotations of energy data are useful for enabling automation of energy management as well as for encouraging improved manual management of energy demand by providing users with a better understanding of how energy usage corresponds to their daily activities. This work presents a novel component for collecting *in situ* annotations integrated into an existing home energy measurement system—Green Homes [2, 3]. Data already collected by the Green Homes energy meters is provided as input to a power profiling algorithm that identifies fine-grained changes in the user's context. The algorithm aggressively prompts the user to annotate his/her context by sending a notification to a smartphone application. In this section, we first provide a brief overview of the Green Homes system, then present the power profiling algorithm and describe the implementation of the notification component.

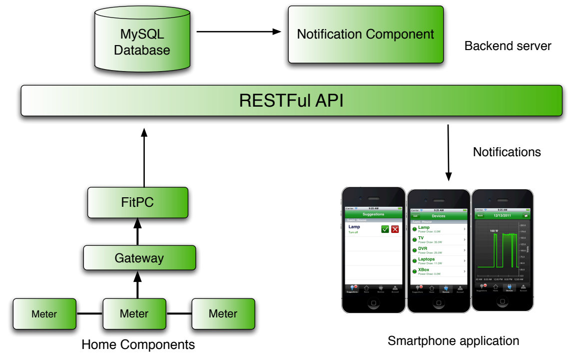

Figure 1: Green Homes architecture.

2.1 The Green Homes System

The Green Homes project [2, 3] is an ongoing effort to understand energy usage in homes, particularly those powered by renewable sources, and to encourage sustainability by developing automated mechanisms for matching energy demand with available supply. At present, 9 homes are participating in our study, including one grid-tied home with solar panels and one off-grid home entirely powered by solar panels. In each home, we have installed between 5 and 10 off-the-shelf energy meters [4] that collect energy usage of appliances include televisions, lamps, microwaves, and computing equipment. Where possible, we also collect energy usage of the entire home.

Figure 1 illustrates the architecture of the Green Homes system. The off-the-shelf energy meters communicate with a dual-radio gateway [5] in each home. Also in each home is a client component that polls the energy meters every 30 seconds and reports readings to a centralized server. Data on the server, including graphs of past usage and current device status, can be accessed by a web or smartphone application (both Android and iPhone are supported). Additionally, every 1 minute, the notification component on the server side executes the power profiling algorithm described below and, if appropriate, pushes a notification to the user's phone. Annotations entered by the user are then stored on the server for postprocessing.

2.2 Power Profiling Algorithm

The notification component uses an aggressive power profiling algorithm to identify possible scenarios when a user's context has changed such that energy usage has increased. We chose not to ask users to annotate decreases in energy consumption to minimize the intrusiveness of the system, and because the primary uses of the annotations are centered around understanding

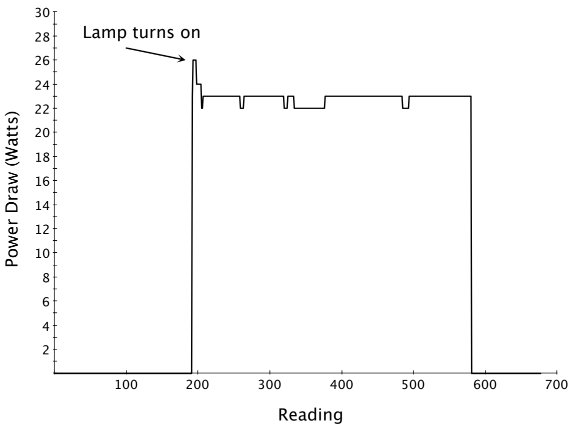

Figure 2: Power draw of a lamp going from off to on and then back to an off state.

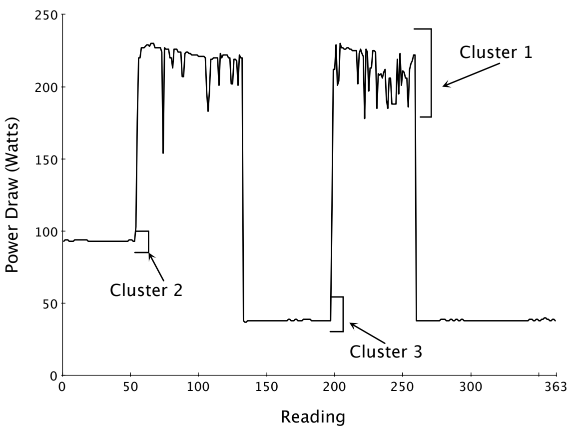

Figure 3: Power draw of a TV and other electronics connected to the same power strip.

periods of high energy consumption. The system is designed to collect annotations at as fine a granularity as possible, therefore the goal is to determine when a device is in use. Recall, the Green Homes system collects instantaneous power draw every 30 seconds, so one might assume that an increase in power draw over a previous reading would indicate that a device was *turned on.* Analysis of our data shows this to be incorrect, however.

To understand why a more sophisticated power profiling technique is required, consider Figures 2 and 3. Figure 2 shows the power draw of a lamp. When off, the lamp shows a power draw of 0 W and it is clear that when the power draw jumps from 0 to over 22 W the device has been turned on. Our first intuition was to use a basic threshold to determine when a device transitions to an on or active state. We ran some initial experiments on one household in our study and, based on the devices in that house, implemented an algorithm that triggered a notification if the power draw reported by a meter jumped by more than 20 W in successive readings. We deployed this preliminary algorithm for two houses in our study and discovered that while the 20 W threshold was appropriate for all devices in the first house, it was not appropriate for all devices in the second. Figure 3 shows the power draw of a power strip that powers a television, cable box, and other electronics in the second house. In this case, the baseline power draw is not a 0 W *off* state, but rather an idle state of nearly 40 W. More interestingly, when the TV is on, the power draw fluctuates between 178 W and 287 W. The occupant discovered that, using the initial algorithm, he received notifications when *changing channels* on his TV! This points to a clear need to perform a smarter analysis to identify on, off, and idle power states of each device.

We have designed a power profiling algorithm that uses the DBSCAN clustering algorithm [6] to produce a unique profile for each device in our study. DBSCAN is a density-based algorithm that identifies clusters while excluding noise. DBSCAN is an ideal choice for this application as it does not require the number of clusters to be provided as input and it can be implemented very efficiently, particularly for one-dimensional data such as ours. The algorithm takes as input two parameters—*eps* specifies the neighborhood of a point, which in our case represents the minimum number of watts separating two distinct power states. We experimentally determined that 2 W yields the best results. The second parameter is *minpts*, which represents the minimum number of points required to form a cluster. We use 10 in our algorithm. For the data from the device shown in Figure 3, the DBSCAN algorithm correctly identifies all readings between 193 and 287 W as belonging to a single cluster. In fact, the algorithm identifies three clusters clearly shown in the figure, plus one additional cluster from 125 W to 139 W.

Once the profile is generated, the final step of the algorithm is to evaluate whether a device is non-interactive and represents only background load for the user. In the case of a refrigerator, for example, the transition into a higher power state likely does not indicate a change in user context. A refrigerator might run twice per hour every hour of the day, hence it is useful to exclude a non-interactive device from notifications. To identify background loads, we apply a heuristic that will classify a device as non-interactive if in more than 80% of the hours for which data was reported for the device there was a change in power state for the device. Effectively, if a device transitions between power states in more than 19 hours of the day then the device is likely not manually controlled by the user and will not trigger notifications.

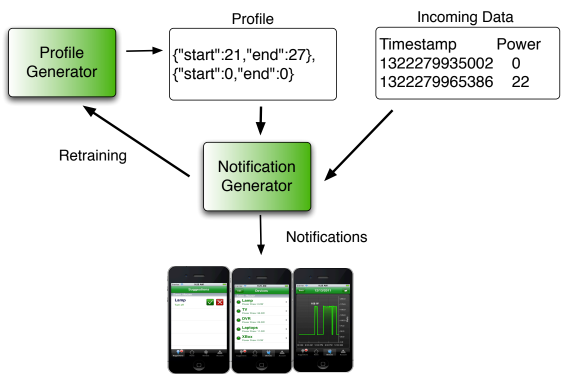

Figure 4: Green Homes notification component.

2.3 Notification Component Overview

The notification component, illustrated in Figure 4, is comprised of a Profile Generator that executes the power profiling algorithm, a Notification Generator that processes incoming data, and a Retraining component that determines whether a profile is stale and should be regenerated.

Profile Generator: The Profile Generator executes the DBSCAN-based profiling algorithm and produces a set of tuples representing the beginning and ending values, in W, for each power state. The background load heuristic is then applied to determine whether the device should be excluded. The resulting profile is stored in the database for use by the Notification Generator.

Notification Generator: The Profile Generator will train on 3 days to 2 weeks of data to produce the initial power profiles. Once trained, the Notification Generator will evaluate incoming data every 1 minute. If, in the past 1 minute, a device has transitioned into a higher power state *and* if no notification has been sent for the given device in the past 1 hour, a notification is triggered. The latter condition is helpful in case of a device such as a washing machine that oscillates between several power states over the course of a single load of laundry.

Retraining: At the end of notification generation, the notification component will determine whether it is necessary to retrain to produce an updated power profile. Retraining will occur if one of two conditions is met. The first condition is that the number of power draw readings that fall outside of the states identified in a device's current power profile exceeds a threshold, in our current implementation 25. This indicates that the power profile of the device has changed and that it is possible that a new state has been introduced. The second condition is the opposite—a state has been eliminated. We determine that this is the case if a given state in the power profile has not been visited for some period of time, in our case 2 weeks. Though 2 weeks seems like a significant period of time, a device like a washing machine may be used infrequently.

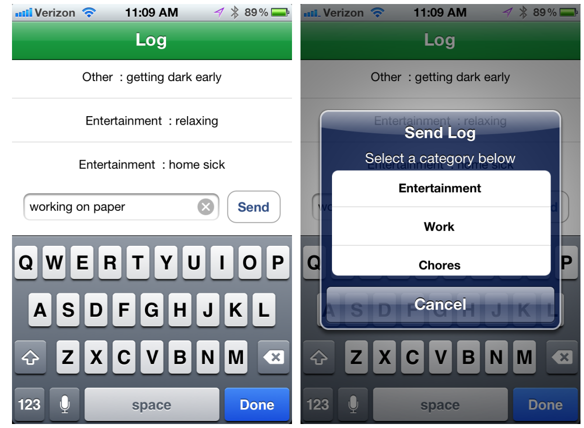

Figure 5: iPhone user interface for providing an annotation.

3. DEPLOYMENT

In order to understand whether the Green Homes notification component is an effective means of collecting contextual annotations for home energy data we report the results of a five-week deployment. The goal of the study is to understand whether the power profiling algorithm works broadly for different types of devices in different homes; whether the system generates a manageable number of notifications at optimal times; and whether subjects respond when they receive notifications.

Setup: To conduct the study, we enabled notifications for 5 of the 9 homes currently participating in the Green Homes project and collected data from the period between November 1, 2012 and December 6, 2012[1]. Table 1 reports the number of days of data collected in each home, as our deployment was rolled out incrementally. Participants were asked to provide feedback on their activities using our iPhone or Android application. In some cases, subjects used their own personal devices and in other cases we provided subjects with an iPod Touch specifically for the experiment. Subjects were asked to log their activity (raw annotations), using a free-form text field, and were also asked to select a category to represent the activity from the following set of categories: `cooking`, `entertainment`, `work`, `chores`, and `other`. Logging activity categories in addition to raw annotations has two advantages. First, it helps us collect more uniform data across subjects. Since raw annotations can vary widely across users, categories provide a common framework to compare data collected from different homes. Secondly, since our subjects were

[1]Our study is ongoing and additional data will be available for a camera-ready version of this paper.

not trained for this project, the raw annotations logged were sometimes obvious and not useful. For instance, one of our subjects logged "Watching TV" when he switched on his television—an event that triggered the notification and can be inferred without any user annotation. We chose these specific categories since home users understand these activity categories well [1]. A screen shot of the application is shown in Figure 5. In addition to asking subjects to log activities when notified, we encouraged them to provide annotations whenever they felt they were performing a relevant activity, even if the system had not notified them. We note that three of our participants, subjects 3, 4, and 5, are researchers involved with this project.

Subject ID	Days of Data
1	35
2	33
3	35
4	25
5	34

Table 1: The table shows the number of days of data collected used in this analysis.

3.1 Power Profiles

The goal of the Profile Generator is to produce profiles that accurately represent the power states of each device in the study and exclude devices that are not interactively controlled by the user. To evaluate this component, we consider the profiles generated for the 39 devices in the study. Our algorithm produced usable profiles for 31 of 39 devices. Among the devices for which no profile was produced, we saw several scenarios. First, we consider training incomplete if the algorithm is unable to identify at least two distinct clusters. In one case, for example, the data came from a DVR that was always on with a power draw within a roughly 2 W range. The notification algorithm identified only one power state, hence it could not be used to generate any notifications. This is expected behavior as there is never a transition from on to off or vice versa. Another example is a power strip that powers two laptops and a printer. In this case, the algorithm again generated a single cluster, but one that went from 4 W, the baseline power draw of the printer, to roughly 70 W, the value to which the power draw spikes when a laptop is plugged in to be charged. Though there are no distinct power states for this device, it does appear that when a laptop is plugged in there is a jump followed by a gradual decrease in power draw. This points to a possible improvement in our algorithm: reverting to a threshold-based approach that generates a notification in case the power draw suddenly increases by more than some threshold. Finally, in several cases, it seemed that the device was simply not used during the experiment period.

Most of the devices in our study yielded profiles with 3 or fewer states, though we did see up to 7 states in some cases. We rely on the user to provide us with a label describing each device (e.g., TV or Lamp), and labels were missing for 3 devices. Table 2 highlights the labeled devices that were classified as having 3 or fewer states, greater than 3 states, or did not produce a usable profile. Note that there were multiple instances of most of the devices described, including TVs and lamps. The classification is as expected for devices such as the lamps, which typically have two states, and TVs, which often have three states representing off, on, and a power-saving mode. We were somewhat surprised that in some cases, for instance in case of the two washers in our study, the same class of device fell into different categories. One factor impacting this classification is the sample rate of the Green Home system. It samples the power draw of each device every 30 seconds, which is too coarse grained to capture fine-grained changes in power draw of devices such as microwaves and washing machines.

Of the four refrigerators in our study, our algorithm initially correctly identified two as non-interactive background loads. Recall that a device will be excluded if it exhibits a power state transition in more than 19 of 24 hours in the day. There were two reasons the algorithm failed to identify the other cases. In the first case, the meter attached to the fridge was misbehaving and reported a power draw of 0 about half of the time. We have found that the off-the-shelf hardware we use is not as reliable as we expected and we must periodically run a network healing procedure to repair broken links in the mesh network formed by the meters. In this case, we asked the subject to heal the network and once this occurred the algorithm was able to correct itself and identified the refrigerator as non-interactive midway through the experiment period. In the second case, the refrigerator simply ran much less frequently—on the order of once every two hours in contrast to the other refrigerators we have measured that run once and sometimes twice per hour. Though we continue to explore additional heuristics, there are other devices in our study, such as the space heater, that are also likely candidates for exclusion but for which we would need to develop a different set of heuristics. Ultimately, we have decided that this is one place where we must have the user in the loop. Midway through our experiment we deployed an update to our iPhone application that allows users to enable and disable notifications as they prefer.

3 or Fewer States	Lamp, TV, XBox, Washer, Oven, Microwave, PC, Electronics Power Strip, Dryer, Laptop, Treadmill
Greater than 3 States	Dishwasher, Space Heater, Washer, TV, Fridge
Unable to Train	DVR, Laptops, Dryer, Router+Gateway+FitPC

Table 2: The table shows a taxonomy of the devices in the study.

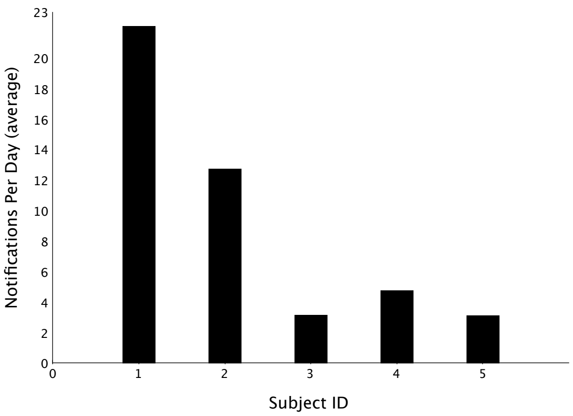

Figure 6: The average number of notifications per day sent to each subject.

3.2 Notifications and Annotations

Subject ID	Number Days with No Notification	Notifications/Day Excl. No Notification Days
1	2	23.4
2	9	16.2
3	8	4.1
4	0	4.8
5	4	3.5

Table 3: The table shows the number of days when no notifications occurred and the mean notifications per day excluding days when no notifications occurred.

We next consider the effectiveness of the system in using the power profiles identified to solicit annotations from our users. We first examine the number of notifications per day sent to each subject in our study in Figure 6. The figure illustrates the mean number of daily notifications for each subject. There was considerable variation in the frequency with which the system requested annotations, with subjects 1 and 2 receiving significantly more notifications than the other subjects. In the homes of both subjects 1 and 2 a refrigerator was measured and accounts for a significant percentage of the daily notifications. As discussed above, in the home of subject 2 the refrigerator was identified as a non-interactive device on November 24 and, after that point, the number of notifications decreased to fewer than 10 per day. Moreover, excluding the refrigerator and space heater—another background load—from the notifications for subject 1, the average daily notifications drops to below 8. Across all homes, the mean number of notifications for interactive devices is 4.6, though this is skewed by a number of days when our subjects were traveling and no notifications were generated. Table 3 shows, for each subject, the number of days when no notifications occurred and the mean number of daily notifications excluding the zero notification days when subjects were likely not home. This analysis demonstrates, first, that identifying background loads is a key component of maintaining a manageable number of potentially intrusive push notifications and, second, that in homes where we generate notifications for interactive devices only, our system is able to use device power profiles to identify 4–5 potentially important changes in user context each day.

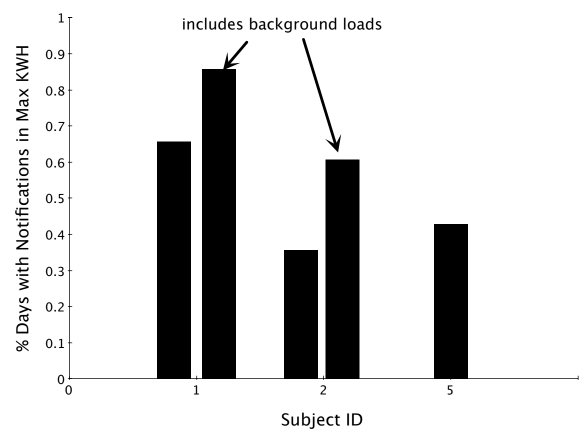

Figure 7: Percentage of days a notification is sent during hours of maximum energy consumption.

The next question that arises is whether our system identifies the most relevant contextual changes for the user, in other words whether the notifications are generated at optimal times. To evaluate this metric, we look at whether notifications were generated during hours of the day when the subject's home had the highest energy demand. Unfortunately, we were only able to collect whole-home data in three homes, as it often requires modifications to a user's main fuse box. Subjects 1 and 2 installed a metering device that reports home energy consumption every 30 seconds via the Green Home infrastructure, and subject 5 was able to provide hourly energy consumption data acquired offline via her utility company.

Figure 7 shows the percentage of days for which a notification was generated during one of the top two peak demand hours for that day. Because subjects 1 and 2 both received notifications for non-interactive devices we report results both including and excluding those notifications. For interactive devices, the notification algorithm identifies hours of peak demand between 36% and 66% of the time. There are two factors that contribute to this result. First, the system does not consider the whole-home energy consumption when determining when to send a notification because this information is not available in most homes in our study. Second, in each home we measure fewer than 10 devices and this number is largely limited by accessibility to the power outlet for the device. In the home of subject 5, for example, the clothes dryer is a significant energy sink but is not measured. Interestingly, however, subject 5's home has only 3 devices that generate notifications but our system is able to identify periods of peak demand nearly 43% of the time. Ultimately, this result points to a main tradeoff of our system— though our algorithm could be improved by hiring an expert to deploy sensors more pervasively, we are still able to identify hours of peak demand about half the time using only a plug-and-play system deployed by the end user.

Subject ID	Annotations Per Day (Mean)	Response Rate/ Excl. Background Notifications
1	2.9	6% / 13%
2	1.2	3% / 7%
3	2.0	33%
4	2.7	15%
5	1.9	43%

Table 4: The table shows the average number of annotations per day for each subject as well as the percentage of notifications that yielded a response from the user within a 10-minute window.

The final question we consider is whether the subjects responded to the notifications sent by the system. Table 4 reports the mean number of annotations per day for each subject as well as the response rate. The response rate is calculated as the number of notifications for which there is a corresponding annotation within a 10 minute window divided by the total number of notifications. For subjects 1 and 2 we again report results including and excluding the notifications generated by background loads.

Overall, the mean number of annotations per day was lower than expected, however Figure 8 shows the distribution of the number of daily annotations for each user. In several cases, the mean is highly skewed by a large number of days for which there was no annota-

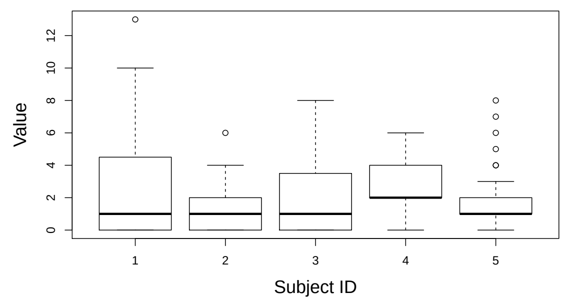

Figure 8: Distribution of the number of annotations per day for each subject.

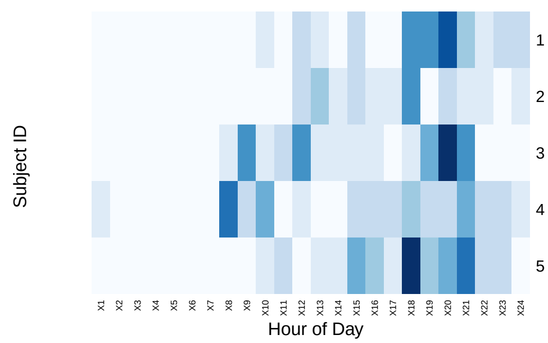

Figure 9: Heatmap showing the number of annotations in each hour of the day. The lightest cells represent 0 annotations with the darkest cells representing a maximum of 12 annotations.

tion, again the likely result of the Thanksgiving holiday when many subjects were traveling. The distribution shows that subjects were willing to provide up to 13 annotations per day, though the response rate was low, particularly for the subjects who received the largest number of notifications. Recall that subjects 3–5 are researchers involved with this project, however subject 1 provided the largest number of annotations. Interestingly, subject 1 recorded a total of 54 annotations where 13% were not in response to a notification while subject 2 recorded a total of 39 annotations where 64% were not in response to a notification. This suggests that users' motivation to annotate varies. Finally, Figure 9 further examines the distribution of annotations through a heatmap of the number of annotations that occurred in each hour of the day. Using our approach, we were able to collect at least one annotation in a minimum of 13 of the 24 hours of the day, with one subject providing annotations in 19 different hour slots.

Summary: Using off-the-shelf energy measurement components and our novel power profiling algorithm our system is able to identify a significant number of im-

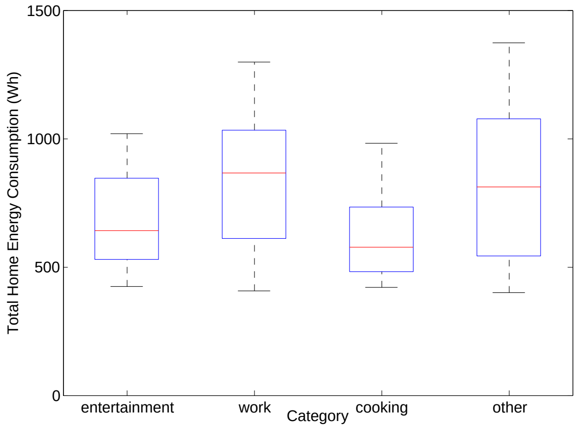

Figure 10: Energy consumption of the entire home when an activity was performed by subject 1.

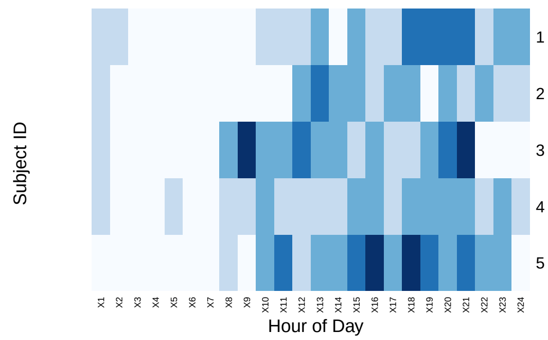

Figure 13: Heatmap showing the number of categories annotated in each hour of the day. White cells represent 0 categories with the darkest cells representing a maximum of 4 categories.

portant energy consumption events. Our data indicate, however, that user responsiveness is inconsistent. Though we were able to collect a measurable number of total annotations during the study, the mean number of annotations per day we collected is consistent with that seen in previous work [1]. This suggests that users are not willing to provide comprehensive annotation of their energy data. In the next section we consider how energy annotations may be used to predict future behavior and energy needs.

4. USING SPARSE ANNOTATIONS

The Green Homes project seeks to predict future energy demand in order to provide users with targeted suggestions for demand management. The previous section demonstrates that users are willing to provide annotations, but not consistently. In this section, we explore how the sparse annotations collected may be used to predict a user's activity and further consider the feasibility of using annotations to predict user energy needs.

4.1 Automated Activity Prediction

We begin by taking a step back to answer the question of whether it would be plausible to devise a completely automated mechanism for inferring a user's activity. Ideally, an unsupervised learning mechanism would exploit common features such as total home energy consumption, appliance power consumption, or time of the day to predict a user's activity. Our analysis, however, confirms the premise of our work—collecting annotations from the user is necessary.

We first consider whether total home energy consumption is sufficient to distinguish between two activities. In this experiment, we use data collected from subject 1 during the entire experiment period. Figure 10 shows the energy consumption of the entire home over one-hour window for each of the four coarse-grained categories logged by this subject. We observe a large variance in the energy consumption for each of the categories. Moreover, the energy consumption profiles across categories have large overlap. We conclude that it would be challenging for an unsupervised classification algorithm to use this feature to infer a category of activity for a user.

We next examine whether the power consumption of individual appliances can be used to distinguish activity categories. In this experiment we consider categories `entertainment`, and `work`. We again use data collected from subject 1 and Figure 11 and Figure 12 show the power consumption of his individual appliances when he logged activity categories `entertainment` and `work`. We again derive the data by looking at a one-hour window and, in this case, we use the maximum power consumption value seen in that window. Looking at these figures side by side, it is clear that there is overlap for all appliances but the treadmill. This suggests that the power draw of an appliance cannot be used to reliably differentiate between these two categories.

Subject ID	Hours with One Annotation Category
1	27%
2	27%
3	15%
4	46%
5	0%

Table 5: The table shows the percentage of hours for which there is only one annotation category. Hours with fewer than 2 annotations are excluded.

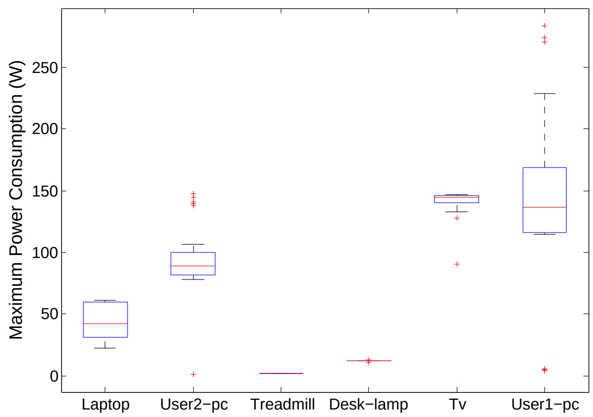

Figure 11: The maximum power consumption of appliances in a one-hour window when subject 1 performed an activity in the category `work`

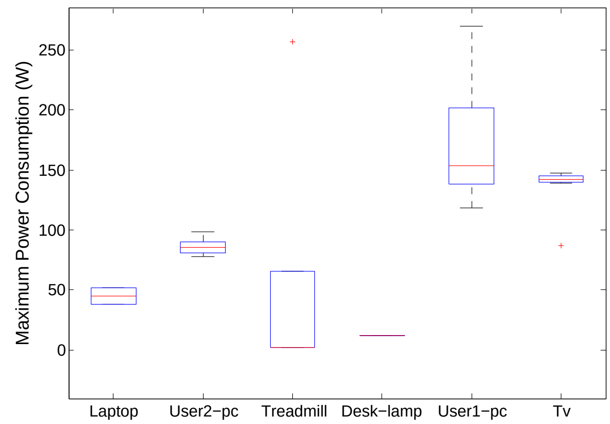

Figure 12: The maximum power consumption of appliances in a one-hour window when subject 1 performed an activity in the category `entertainment`

We finally consider the time of the day feature using Figure 13 and Table 5. Figure 13 is a heatmap that reports the number of categories logged by all subjects in each one-hour time slot across a day. All subjects report more than one category in several time slots and, in some time slots, up to four different categories are reported. We examined a similar heatmap for day of the week and found consistent results—two or more categories are reported each day for all subjects. Table 5 considers, for each subject, the percentage of time slots for which a user reported at least two annotations and both were of the same category. We observe that it is rare for most subjects to report only one category during a given time slot—the percentage of hours when a single category is annotated is less than 50% , and is 0% for subject 5. We conclude that an unsupervised learning scheme that uses time of the day as a feature to derive an annotation category is likely to get confused.

4.2 Semi-Automated Activity Prediction

We next explore whether the sparse annotations we are able to collect with our notification system are sufficient to predict a user's future activities, which provide insight into the user's energy needs. We consider whether a supervised learning algorithm identifies activities with reasonable accuracy. We find that in cases where our test set contains annotations that enable us to verify our predictions, our predictions are generally accurate. However, the sparseness of the annotations in our test set results in the inability to verify our predictions in many cases.

First, we provide an overview of the supervised learning algorithm we use for activity predictions. Our approach uses the Least Square Support Vector Machine (LS-SVM) as a supervised classifier. The algorithm uses a RBF kernel [7] and the simplex optimization framework to determine a minimal error matching between features and classes. We chose a SVM algorithm as they have been shown to perform well in predicting energy demand and appliance power consumption [8, 9]. The classifier is trained on two features: time of the day, and day of the week. It predicts the user's activity for every one-hour time slot in the test period. We use data from our subjects between November 5—November 19 for training and November 20—December 3 for testing.

The classifier is trained on only valid annotation data and the accuracy is calculated by looking at only the one-hour slots for which we have a valid annotation in the test set. This approach effectively assumes that we know *apriori* when a user is performing some activity. The accuracy is the percentage of one-hour time slots where we predict the activity category correctly.

Subject	Accuracy
1	75%
2	45%
3	36%
4	52%
5	65%

Table 6: Classification accuracy of the LS-SVM algorithm that only considers time slots when a valid activity annotation exists in the data set. The experiment assumes that the algorithm knows apriori that a valid activity is being performed.

Table 6 shows the mean classifier accuracy for 20 independent runs. In most cases, we can predict nearly half or more of the activity categories. Subject 3 has the lowest accuracy, however, we observed a 9% standard deviation in accuracy across the 20 independent

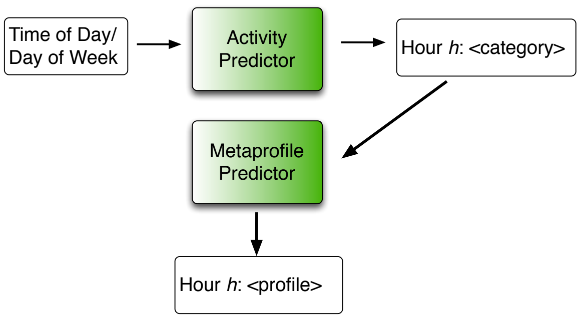

Figure 14: Overview of the algorithm used to predict the power profile of a user.

runs of the classifier. The variance occurs when the training data is noisy and the simplex optimization algorithm does not converge to the same optima. The overall observation is that accuracy is reasonable, but clearly impacted by the problem identified in Figure 13. Because users annotate several categories during most times of day and days of week, there is a tradeoff between accuracy and the goal of minimizing intrusiveness by using only features easily accessible by an energy measurement system.

Though the LS-SVM learning algorithm has reasonable accuracy when we consider only the hours when we have user annotations, we cannot overlook the question of what happens during hours when annotations are not present. We evaluated an approach that classified all unannotated time slots as a generic `category 6` and found that our algorithm performed quite well, but only because the skewed number of `category 6` entries in the training set prompted it to trivially predict `category 6` for all hours in the sparsely-annotated test period. One key question is how to determine whether the lack of an annotation implies that the user is not performing an activity (e.g., was not home), or is simply unable or unwilling to provide an annotation in that hour. We explored several algorithms that attempt to filter hours where the user is likely performing no activity, but ultimately concluded that with only sparse data available to test our algorithms we are unable to determine whether many of our predictions are correct or incorrect.

4.3 Using Annotations for Profile Prediction

We next turn our attention to whether it is feasible to use activity predictions to improve prediction of a user's energy needs. In this section, we consider whether our activity predictions can be used to improve the performance of an algorithm that predicts a *metaprofile* comprised of the power states of each measured device in a user's home. The metaprofile provides insight into a user's energy requirements, which can be used as the basis for intelligent demand management.

The **metaprofile** for an hour is a tuple $< A_1, ..., A_k >$, where A_i takes the value *high* or *low* depending on whether the the power consumption reported by device i was high or low in the corresponding hour. To determine the high or low value, we consider the maximum power draw reported by a device during the hour and the overall maximum power draw observed by the power profiling algorithm discussed in Section 3.1. If the maximum in the current hour exceeds one half of the overall maximum the value will be high, otherwise the value will be low. This provides a coarse estimate of whether each measured device in a user's home is on or off during a particular hour. Our analysis produced between 3 and 12 metaprofiles for each subject.

The goal of this analysis is to determine whether we are able to more accurately predict a user's metaprofile when using annotations versus using only time of day and day of week as features. The metaprofile prediction algorithm works in two phases as shown in Figure 14. First, time of day and day of week are used to train the activity prediction algorithm described in the previous section, and activities are predicted for the test period. Similarly, annotated activity is used to train the metaprofile prediction algorithm. Next, the *predicted* activity is fed into the metaprofile prediction algorithm, which predicts the metaprofile for each hour of the testing period.

Subject	Accuracy (annotations)	Accuracy (td, dw)
1	35%	35%
2	54%	54%
3	32%	30%
4	55%	33%
5	80%	80%

Table 7: Accuracy of predicting metaprofiles using user annotations versus just time of the day and day of the week. Annotations help for two subjects, but in other cases inaccuracies stem from misclassified categories and overlap of disjoint metaprofiles and categories.

Similar to the experiments in the previous section, the classifier is trained on only valid annotation data and the accuracy is calculated by looking at only the one-hour slots for which we have a valid annotation in the test set. Table 7 presents the results of this experiment. We present the mean of 10 independent runs. The results demonstrate that using activity annotations as a feature improves metaprofile prediction in 2 of 5 cases. For other subjects, the annotations produce accuracies equivalent to using time of the day and day of the week features. There are two reasons why annotations do not produce better results for three of our subjects. First, there is considerable overlap between activity categories and metaprofiles. The overlap between `work` and `entertainment` categories for subject 1, for example, is illustrated in Figures 11 and 12. Due to this overlap, even though 60% of the misclassified metaprofiles correspond to predicted activities that were correctly classified during the first phase of our algorithm, the predicted metaprofiles where incorrect. Second, there are several instances where activities are mispredicted during the first phase of the algorithm, leading to poor metaprofile classification. In subject 3's case, for instance, 75% of the misclassified profiles correspond to incorrect activity prediction during the first phase of the algorithm. This implies that if the accuracy of predicting activities can be improved, the metaprofiles can also be better predicted. Finally, for subject 4, the annotations produce a 22% improvement. A closer look at the results show that the time of the day and day of the week features mispredict metaprofiles for a power strip that has five devices attached—TV, Roku, Playstation, DVD, and Lamp. The poor classification using time of day and day of week implies that the power strip is not consistently used at the same times. This is likely because it is used in different ways, for example sometimes to watch TV, some times to play video games, and sometimes just to light the room. It is, however, used in the same way for a given activity. Hence our algorithm that predicts a user's activity can identify how the power strip is used at those times.

Discussion and Future Work: The results in this section demonstrate that deriving annotated data requires user involvement. Unsupervised learning algorithms that use raw data to derive activities are implausible. Since users are willing to provide annotations, but not consistently, it is difficult to collect a dense annotation data set. Nonetheless, with an average of only a few annotations per day, we can predict a user's future activity with reasonable accuracy. Moreover, in some cases, using only sparse energy annotation data yields an advantage in determining a user's future energy requirements.

The insights we have gained in this study suggest several avenues for future work. A key challenge we encountered is the sparseness of our data set and the inability to derive contextual meaning for periods with no annotation. We plan to explore whether we can, in some cases, automatically derive annotations given a sparse data set and, further, whether we can identify the most critical periods with missing annotations. This would allow us to augment our system to let the user know when he/she should be most vigilant about providing annotations. Similarly, we plan to provide future users with more specific guidance regarding the types of annotations that are most useful. Finally, we plan to further explore the parameters that impact when annotations are useful in understanding energy needs.

5. RELATED WORK

Our paper builds on previous work on context aware energy management, methods to collect labeled energy data, and demand-response systems in a home environment. Here we we compare and contrast our contributions with the most relevant literature.

5.1 Context-aware Energy Management

Augmenting energy data with contextual information can help in better demand prediction of home appliances and whole home energy consumption [10, 11, 12, 13, 14]. Most of the focus in this area has been on collecting auxiliary data on temperature, humidity, weather conditions, ambient light, and time. Collecting these additional dimensions of data require deploying sensors such as temperature, motion, light, and humidity [15, 16, 17, 18]. Unfortunately, dense deployment of sensors can be expensive and intrusive [19]. Moreover, these sensor sources provide fairly generic data and deriving usage context and relationship with energy consumption is challenging. Our work focuses on collecting context through user activities. Activities such as cooking, entertainment, and work are directly related to home energy consumption and provide an easy to use context for demand prediction [20]. Moreover, users can relate to and understand such activities [1] and energy conservation recommendations made akin to these activities have a higher chance of being adopted. We have designed an annotation system that reminds users to enter their activities based on energy level changes for specific appliances. The system uses energy meters that are apriori installed in homes to collect appliance and home specific energy measurements, and hence does not require any additional hardware installations. Our work also relates to aging in place research [21], and activity recognition in smart homes [22, 20, 23]. Aging in place research focuses on movement and fall detection for the elderly while our focus is on activities like cooking, entertainment, and work that are directly related to energy consumption. Additionally while certain activities described in these papers (Kasteren et al. [23], Tapia et al. [22], and Szewcyzk et al. [20]) can be used in the energy management context, the focus of these papers are on methods to use sensors like cameras and motion detectors to automatically infer activities.

5.2 Energy Annotations

Collecting energy annotations and providing feedback to users on energy conservation are open areas in the human-computer interaction community. For instance, eco-feedback systems use visualization techniques to engage the user in the energy conservation process [24, 25, 26]. A primary goal is to make the user cognizant to potential energy bottlenecks. For example, Costanza et al. [1] propose a time series based web interface called

FigureEnergy where users can label their activities. They also provide a visualization interface where users can understand the impact of performing certain activities on the total home energy consumption. Unfortunately, the annotations are based on the user remembering what activities he performed during the day. Our system *in-situ* monitors the energy consumption of appliances and notifies users to log activities. Additionally, our goal is to use the activities as context to provide feedback on how to conserve energy both at the granularity of appliances and whole home energy consumption. Chen et al. [27] use manually collected annotations to predict energy consumption of different activities. They show that the prediction algorithm has lower than expected accuracy. Our system focuses on collecting activity annotations when the power consumption of appliances change.

5.3 Demand-response in Homes

A primary goal of the Green Homes project is to devise techniques that balance energy supply with demand, especially in renewable energy driven homes. To this end, energy annotations and labeled energy data provide better ways to understand energy demand, predict future energy consumption, and provide timely energy saving recommendations to home users. Hence, our energy annotation techniques are complementary to several demand-response systems in homes [28, 29] and can be used to improve their performance. These include systems that flatten peak energy consumption [30], methods to predict energy generation and consumption [31], balance energy demand with supply [3], and minimize whole building energy consumption.

6. CONCLUSION

Understanding correlations between user activities and power consumption is key to designing better home energy management systems. Existing approaches for activity inferencing rely on expensive and potentially intrusive sensor deployments in the home. In this paper, we present a system for collecting user annotations using a mobile application. Our system requires only the sensors necessary to monitor energy usage of home appliances. We present a novel technique that utilizes changes in appliance power signatures to push notifications to a user's mobile phone to solicit activity annotations. Through a five-week deployment of our system in 5 homes, we show that users are willing to provide a large variety of annotations, but response rates are lower than expected, creating a sparse data set. Based on the data collected, we show that manual activity annotations are mandatory—unsupervised learning techniques that derive annotations from unlabeled data are implausible. Moreover, sparse annotations can be used to predict a user's activity 50% of the time when it is known that some activity is being performed.

Acknowledgments

We wish to thank all of our study participants. We also thank Kevin Moran, Neal Xiong, and Simon Piel for their contributions to the Green Homes system. This material is based upon work supported by the National Science Foundation under Grant No. 1115798, 1055061, 1018112. Any opinions, findings, and conclusions are those of the authors and do not necessarily reflect the views of the NSF.

7. REFERENCES

[1] Enrico Costanza, Sarvapali D. Ramchurn, and Nicholas R. Jennings. Understanding domestic energy consumption through interactive visualisation: a field study. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pages 216–225, New York, NY, USA, 2012. ACM.

[2] David Lachut, Lazeeb Choudhury, Kevin Moran, Simon Piel, Yucheng Xiong, Nilanjan Banerjee, and Sami Rollins. Minimizing Intrusiveness in Home Energy Measurement. In *ACM Workshop On Embedded Systems For Energy-Efficiency In Buildings*, November 2012.

[3] Nilanjan Banerjee, Sami Rollins, and Kevin Moran. Automating Energy Management in Green Homes. In *ACM Sigcomm Workshop on Home Networks*, August 2011.

[4] `http://www.aeon-labs.com/site/products/view/5/`. Aeon labs smart energy switch.

[5] `http://micasaverde.com/vera.php`. Micasaverde vera 2.

[6] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231, 1996.

[7] J.A.K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9:293–300, 1999.

[8] Chao Chen and D.J. Cook. Behavior-based home energy prediction. In *Intelligent Environments (IE), 2012 8th International Conference on*, pages 57 –63, june 2012.

[9] Richard E. Edwards, Joshua New, and Lynne E. Parker. Predicting future hourly residential electrical consumption: A machine learning case study. *Energy and Buildings*, 49(0):591 – 603, 2012.

[10] Corinna Fischer. Feedback on household electricity consumption: a tool for saving energy? *Energy Efficiency*, 1:79–104, 2008.

[11] Geraldine Fitzpatrick and Greg Smith. Technology-enabled feedback on domestic energy consumption: Articulating a set of design concerns. *IEEE Pervasive Computing*, 8(1):37–44, January 2009.

[12] Tom Hargreaves, Michael Nye, and Jacquelin Burgess. Making energy visible: A qualitative field study of how householders interact with feedback from smart energy monitors. *Energy Policy*, 38(10):6111 – 6119, 2010.

[13] James Pierce, Chloe Fan, Derek Lomas, Gabriela Marcu, and Eric Paulos. Some consideration on the (in)effectiveness of residential energy feedback systems. In *Proceedings of the 8th ACM Conference on Designing Interactive Systems*, DIS '10, pages 244–247, New York, NY, USA, 2010. ACM.

[14] Yolande A.A. Strengers. Designing eco-feedback systems for everyday life. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2135–2144, New York, NY, USA, 2011. ACM.

[15] Thomas Schmid, David Culler, and Prabal Dutta. Meter any wire, anywhere by virtualizing the voltage channel. In *BuildSys*, 2010.

[16] Jay Taneja, David Culler, and Prabal Dutta. Towards cooperative grids: Sensor/actuator networks for promoting renewables. In *SmarGridComm*, 2010.

[17] Vijay Srinivasan, John Stankovic, and Kamin Whitehouse. WaterSense: Water Flow Disaggregation using Motion Sensors. In *BuildSys*, 2011.

[18] Shwetak N. Patel, Sidhant Gupta, and Matthew S. Reynolds. The design and evaluation of an end-user-deployable, whole house, contactless power consumption sensor. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 2471–2480, New York, NY, USA, 2010. ACM.

[19] Timothy W. Hnat, Vijay Srinivasan, Jiakang Lu, Tamim I. Sookoor, Raymond Dawson, John Stankovic, and Kamin Whitehouse. The hitchhiker's guide to successful residential sensing deployments. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, SenSys '11, pages 232–245, New York, NY, USA, 2011. ACM.

[20] S. Szewcyzk, K. Dwan, B. Minor, B. Swedlove, and D. Cook. Annotating smart environment sensor data for activity learning. *Technol. Health Care*, 17(3):161–169, August 2009.

[21] Julie A. Kientz, Shwetak N. Patel, Brian Jones, Ed Price, Elizabeth D. Mynatt, and Gregory D. Abowd. The georgia tech aware home. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '08, pages 3675–3680, New York, NY, USA, 2008. ACM.

[22] Emmanuel Munguia Tapia, Stephen S. Intille, and Kent Larson. Activity recognition in the home using simple and ubiquitous sensors. In *In Pervasive*, pages 158–175, 2004.

[23] Tim van Kasteren, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse. Accurate activity recognition in a home setting. In *Proceedings of the 10th international conference on Ubiquitous computing*, UbiComp '08, pages 1–9, New York, NY, USA, 2008. ACM.

[24] Jon Froehlich, Leah Findlater, Marilyn Ostergren, Solai Ramanathan, Josh Peterson, Inness Wragg, Eric Larson, Fabia Fu, Mazhengmin Bai, Shwetak Patel, and James A. Landay. The design and evaluation of prototype eco-feedback displays for fixture-level water usage data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2367–2376, New York, NY, USA, 2012. ACM.

[25] Jon Froehlich, Leah Findlater, and James Landay. The design of eco-feedback technology. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1999–2008, New York, NY, USA, 2010. ACM.

[26] Yolande A.A. Strengers. Designing eco-feedback systems for everyday life. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2135–2144, New York, NY, USA, 2011. ACM.

[27] Chao Chen, Barnan Das, and Diane J. Cook. Energy prediction based on resident's activity. In *SensorKDD*, 2010.

[28] Tanuja Bapat, Neha Sengupta, Sunil K. Ghai, Vijay Arya, Yedendra B. Shrinivasan, and Deva Seetharam. User-sensitive scheduling of home appliances. In *GreenNets*, 2011.

[29] Thomas Weng, Bharathan Balaji, Seemanta Dutta, Rajesh Gupta, and Yuvraj Agarwal. Managing plug-loads for demand response within buildings. In *In Proceedings of the ACM Workshop on Embedded Sensing Systems For Energy-Efficiency*, Seattle, October 2011.

[30] Sean Barker, Aditya Mishra, David Irwin, Prashant Shenoy, and Jeannie Albrecht. SmartCap: Flattening Peak Electricity Demand in Smart Homes. In *IEEE International Conference on Pervasive Computing and Communications*, March, 2012.

[31] Ting Zhu, Aditya Mishra, David Irwin, Navin Sharma, Prashant Shenoy, and Don Towsley. The case for efficient renewable energy management for smart homes. In *BuildSys*, 2011.