WattShare: Detailed Energy Apportionment in Shared Living Spaces within Commercial Buildings

Shailja Thakur[†], Manaswi Saha[†], Amarjeet Singh[†], Yuvraj Agarwal[‡]

[†]Indraprastha Institute of Information Technology, Delhi [‡]Carnegie Mellon University

[†]{shailja1275, manaswis, amarjeet}@iiitd.ac.in, [‡]yuvraj.agarwal@cs.cmu.edu

Abstract

Increasing energy consumption of commercial buildings has motivated numerous energy tracking and monitoring systems in the recent years. A particular area that is less explored in this domain is that of energy apportionment whereby total energy usage of a shared space such as a building is disaggregated to attribute it to an individual occupant. This particular scenario of individual apportionment is important for increased transparency in the actual energy consumption of shared living spaces in commercial buildings e.g. hotels, student dormitories and hospitals amongst others. Accurate energy accounting is a difficult problem to solve using only a single smart meter. In this paper, we present a novel, scalable and a low cost energy apportionment system called WattShare that builds upon our EnergyLens architecture, where data from a common electricity meter and smartphones (carried by the occupants) is fused, and then used for detailed energy disaggregation. This information is then used to measure the room-level energy consumption. We evaluate WattShare using a week long deployment conducted in a student dormitory in a campus in India. We show that WattShare is able to disaggregate the total energy usage from a single smart meter to individual rooms with an average precision of 96.42% and average recall of 94.96%. WattShare achieves 86.42% energy apportionment accuracy which increases to 94.57% when an outlier room is removed.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Design, Experimentation

Keywords

personal energy apportionment, smartphones, smart meters, energy disaggregation

BuildSys'14, November 5–6, 2014, Memphis, TN, USA.
Copyright 2014 ACM 978-1-4503-3144-9/14/11 ...$15.00
http://dx.doi.org/10.1145/2674061.2674069

1 Introduction

Increasing electricity consumption has been an ever-growing concern for the past several decades. Buildings, specifically commercial buildings, account for a significant proportion of the overall energy use globally. Within commercial buildings, shared living spaces e.g. dormitories, hotels and hospitals have a peculiar feature. Occupants in these shared living spaces typically occupy their own room but are not billed for their actual energy consumption, resulting in higher energy wastage. Even if the property owner would want to do separate billing for each room, complex electrical infrastructure together with high metering costs makes it prohibitive.

Techniques for disaggregating meter level data into appliance level consumption, also referred to as Non-Intrusive Load Monitoring (NILM), have been studied [3, 9, 12, 16]. However, any such approach by itself, is likely to fail in the scenario of shared living spaces due to multiple appliances, typically of the same type, being simultaneously operated across different rooms. Recently, *personal energy apportionment* i.e. moving from disaggregation at the appliance level to the user level, has gained some traction [5, 10, 13, 17]. In this problem, the primary goal is to distribute the total energy consumption to the individuals based on their personal usage.

Measuring the energy usage of an individual, within a multi-occupant home, is non-trivial as it requires monitoring at a much fine grained level e.g. having the information about what the user is doing, when is he doing it, where is he doing it and so on. Shared living spaces are a special case, whereby the problem of *personal energy apportionment* is relaxed to an extent that the personnel typically occupy different rooms and hence room-level energy apportionment is sufficient to achieve the desired results. Such room level energy apportionment will also then allow for billing each occupant for their own energy consumption, thus motivating energy conservation behavior.

The usual layout in these shared living spaces are a cluster of rooms (collectively referred to as a wing in this paper) laid out in a sequential order. Total power consumption at the wing level can be easily monitored as it is typically fed through a separate electrical panel. Smart electricity meters, allowing for sampling up to 1 Hz, are now becoming common and affordable. Concurrently, smartphones have also become hugely popular over the past decade. We take the advantage of wide availability and affordability of smart meters and smartphones to propose WattShare - a system that apportions the aggregate power consumption, measured at the wing level using a smart meter, to individual rooms of the wing, taking advantage of the sensory information provided by the smartphones carried by the room occupants.

WattShare utilizes signal strength values from WiFi scans and audio signals from the microphone as input data sources from the smartphone, per phase power consumption from the 3–phase smart meter and some metadata that can be easily collected (e.g. type of appliances in each room and distribution of the three electrical phases across different rooms) to achieve room level energy apportionment. We use WiFi signal strength to estimate the room occupancy while the audio data is used to differentiate between the events occurring across different rooms.

WattShare system design, uses a similar system architecture as EnergyLens [17] that was proposed for personal energy apportionment in residential settings. However, the set of inference algorithms used to process and combine the multi-modal sensory information from both the phone and the meter are largely different. By combining the different sensor inputs, each being processed using simple techniques, WattShare is able to accurately measure per room energy usage, accounting for even the events caused by low power consuming appliances such as lights and fans which are otherwise hard to detect with other NILM techniques.

In summary, the primary contributions of this work are outlined below:

- We introduce WattShare - an energy disaggregation and apportionment system that provides room-level energy usage together with the disaggregated appliance usage within the room i.e. identifying and measuring the contribution of each appliance that contributed towards the room's energy usage.
- We demonstrate our system's effectiveness with a week long deployment of our prototype system in a student dormitory building in IIIT-Delhi, India.

2 Related work

Researchers have proposed several techniques for energy disaggregation for monitoring energy usage, analyzing consumption patterns and, motivating users to reduce their consumption through regular feedback. In this section, we specifically focus only on the studies done in the area of energy apportionment and discuss systems that have been designed to target this problem since it is the closest problem to the one we are trying to solve with WattShare.

Apportionment of energy to individuals is a challenging task. Hay [10] investigated the problem of apportionment in a shared office environment, illustrating policies that might work and the role of sensor systems for apportionment. A recent paper [8] that conducted two studies showed that providing information to users about their personal energy consumption does help raise people's awareness, change their perception about energy consumption and eventually has an impact on their usage behavior. They specify that factors such as lack of appropriate information and low cost of utility bills, leads to careless attitude towards energy usage. Even though the study was done at a small scale (12 participants in the first lab study and 4 shared households with a total 21 participants in the second two week field study), the results are encouraging and warrant the need to develop a better understanding of this problem.

Very few systems exist in the field of personal energy monitoring. Personalized Energy Auditor [13] is one such system that estimates personal energy usage by tracking the user's movements (using Wifi scans from his/her smartphone and the doorway sensors installed in the house) and correlating it with appliance usage that is monitored by the home electricity meter. Another similar system is our prior work on EnergyLens [17] that combines data from smartphone sensors (WiFi signal strength and audio) and the smart electricity meters, to provide fine grained apportionment information for each occupant in a home. It identifies four main pieces of information required for energy apportionment i.e. *"who"* did the activity, *"what"* was the activity, *"where"* was it done and *"when"* did it occur. Both of these systems were designed for residential settings. WattShare extends the EnergyLens system architecture to a commercial setting and provides room level energy usage from the common electricity meter. To the best of our knowledge, WattShare is the first energy apportionment system designed for commercial buildings, specifically the shared living spaces.

3 System Architecture

We now describe the design of our proposed WattShare system that fuses sensory information from smartphones, appliance metadata, with energy usage from a central smart meter for energy apportionment to individual rooms and occupants. We first describe the individual components of WattShare (Figure 1), how they connect to each other, as well as how they translate to different stages in our apportionment algorithm.

3.1 WattShare Input Sources

3.1.1 Electricity Meter Data

We start by describing the different sensor sources that we use for WattShare. The first sensor source is the total power consumption data of the entire shared space using standard networked three phase energy meters, collected at the rate of 1Hz. Our algorithm uses an edge detection strategy developed in our previous work [17] to detect rising and falling

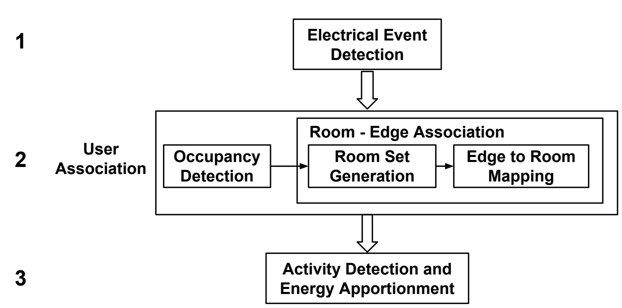

Figure 1: WattShare Algorithm Stages

Phase	Room1	Room2	Room3	Room4	Room5	Room6
Phase 1	Light, Fan, Plug	Light, Fan, Plug	-	-	-	-
Phase 2	-	-	Fan, Plug	Fan, Plug	Fan, Plug	Fan, Plug
Phase 3	-	-	-	-	-	-
Phase 1*	-	-	-	-	-	-
Phase 2*	Plug, AC	Plug, AC	Light	Light	Plug, AC	Plug, AC
Phase 3*	-	-	Plug, AC	Plug, AC	Light	Light

Table 1: *MetaData - I* : Appliance–Phase–Location Mapping. Two 3–phase meters, separately monitoring the light and the power loads, were installed at the wing level. Phase 1–3 belong to Meter-1 and Phase 1*–3* belong to Meter-2.

edges, which correspond to changes in power levels as appliances turn ON/OFF.

3.1.2 *Smartphone Sensors*

Our WattShare smartphone application, running on an occupant's smartphone, collects two sensor data streams periodically and sends them to our analysis server. The first stream is a periodic scan of visible WiFi APs (BSSIDs) and their Received Signal Strength (RSS) alongwith a time stamp. This data is used by our localization algorithm to detect the occupants' location within the building.

The second sensor stream captured from the occupant's smartphone is the raw Pulse Code Modulated (PCM) signal (sampled at 8 KHz) from the microphone which is processed by an audio processing pipeline. It involves 1) preprocessing the raw signals and 2) generating consequent audio features from them. For the first step, audio signal is sampled with a duty cycle of 50% and divided into 500 ms frames. Next, a Hamming window function is applied onto each frame. Finally, 13 MFCC (Mel Frequency Cepstral Coefficient) [6] features are calculated from the processed audio samples. The raw PCM data is discarded (ensuring the privacy of the user) and only the extracted features are sent to the server at regular intervals. We use these features to differentiate between electrical events in our WattShare algorithm.

3.1.3 *Metadata*

In addition to the sensory input from smart phones and electricity meter, the third input to the system is a set of static information about the shared environment called *Metadata*. This information is collected at the training phase and is currently done manually. As we envision our system to be used in commercial buildings, the facilities department and the occupants would ideally collect this data using our app itself. We collect two kinds of static information about the appliances present in the shared area. They are:

- *Appliance–Phase–Location Mapping*: It contains the list of appliances tagged with their corresponding room. They are tagged with the electrical phase they are on using the building electrical layouts.
- *Appliance–Power Mapping*: It contains the mapping of each appliance with its power consumption.

Illustration of Appliance–Phase–Location Mapping (*Metadata - I*) and Appliance–Power Mapping(*Metadata - II*) are shown in Table 1 and Table 2. In modern buildings, multiple electrical phases are pretty typical, which are often uniformly distributed to provide some load balancing. In fact at both UCSD and IIIT-Delhi, the rooms had multiple phases, in case one of the phases goes down (e.g. in India) or to balance loads (e.g. in the US, each phase is on a 15A/20A circuit breaker). Such a distribution makes the overall room level disaggregation complex. Appliance–Phase mapping, even though complex, is appropriately used by the WattShare algorithm, as explained in Section 3.3.

Appliance	Magnitude (Watts)
Light	35
Fan	35
Laptop	60
AC	630

Table 2: *MetaData - II* : Appliance–Power Mapping. Appliances across all the rooms are same in terms of number as well as their make and model.

3.2 Training Stage

During the training stage, we collect the *Metadata* (as explained in the previous section) and calculate a set of thresholds required by the WattShare algorithm, from the WiFi and audio data streams. We use WiFi signal strength for localizing the user within his/her room. For each room, we observe the range of signal strength values received from the visible APs for 5 minutes (see Figure 2). Therefore, if there are k visible APs in a room, then there would be k sets of signal strength values associated with the room. For each of these AP range sets, the values that lie within the 25th and 75th percentile of the range set is defined as a *threshold range*. The k threshold ranges associated with the room are then used for localizing a user to his/her room (usage described in *Occupancy Detection* step of Section 3.3.2). Note that other WiFi based algorithms [2, 11] and more advanced indoor localization techniques [14, 18] can also be used in case of more complicated building layouts, although we found our simple technique to work well in practice for our testbed.

Audio information is used to differentiate between events occurring across different rooms. In order to do so, we assign each room a set of threshold values corresponding to events such as switching ON/OFF an appliance (e.g. fan, light, plug loads and AC) and locking/unlocking the room door while turning ON/OFF the same appliances. We calculate the threshold values by taking two windows of 60 seconds each, one before (w_{pre}) and one after (w_{post}) the event time. For all the frames in each window, we extract MFCC feature vectors and take the Euclidean distance between the two [15]. The calculated distance is termed as the *event*

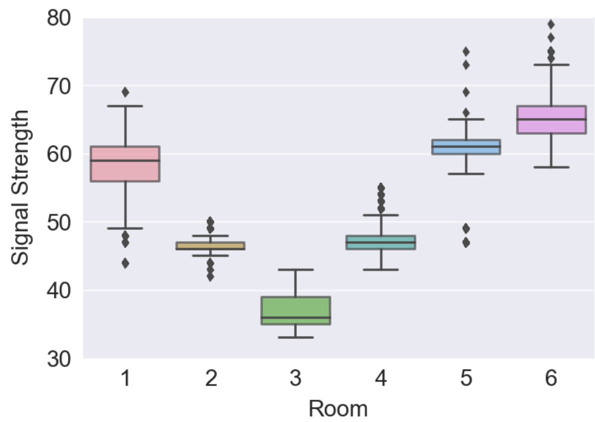

Figure 2: Range of signal strength values observed in each room from one access point. This figure shows the plot of 5 minutes worth of WiFi data collected in each room.

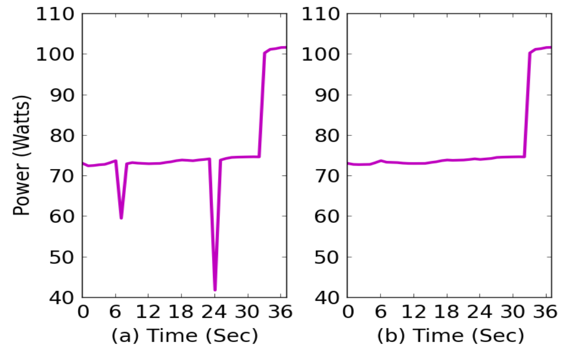

Figure 4: (1) Power Trace with noise (Left) (2) Filtered Power Trace(Right)

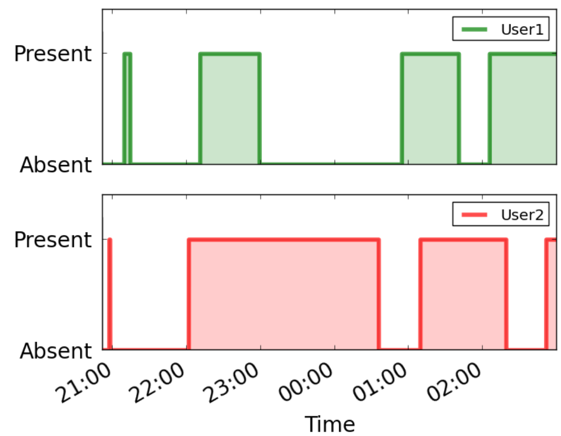

Figure 5: Occupancy Detection. This figure illustrates the outcome of this step. It shows the presence/absence in the room for each user.

threshold. This is repeated for each event and for each room. At the end of the process, we have a threshold set containing values corresponding to the events specified above for each room. We now explain the various stages involved in the WattShare algorithm.

3.3 WattShare Algorithm

The central component to our WattShare system is the apportionment algorithm, which comprises of three stages as shown in Figure 1 at a high level: (1) Electrical Event Detection (2) User Association (3) Activity Detection and Energy Apportionment. Figure 3 describes each of the algorithm stages in further detail.

3.3.1 Electrical Event Detection

The first stage, *Electrical Event Detection*, consists of detecting all electrical events or edges (rising and falling) from the raw power trace using the basic edge detection algorithm. Before performing the edge detection process, we remove noise that is observed on each of the electrical phases. Periodic spikes with the magnitude similar to that of the appliances on that phase (as shown in Figure 4) were observed and filtered out. Once all such noise spikes are removed, we run our edge detection algorithm. Each detected edge is marked as a tuple $e_i = (t_i, m_i, p_i)$ where t_i is the time at which the event occurred, m_i is the power magnitude of the edge (in Watts) and p_i is the electrical phase on which the event is observed. For more details on the edge detection algorithm, please refer to [17].

3.3.2 User Association

In the second stage, *User Association*, we associate the detected electrical events with the respective rooms. This is the most important stage of the WattShare algorithm that involves three steps, namely, *Occupancy Detection* followed by *Room Set Generation* and *Edge to Room Mapping*. The last two steps are together termed as '*Room–Edge Association*'.

Occupancy Detection In this step, the WiFi data stream from the phone is first summarized by taking the mean of signal strength samples (received from visible access points) for a window of 20 seconds. Each record in the WiFi data stream is of the form $< time, rss_1, rss_2, ..., rss_k >$ where $rss_1 \cdots rss_k$ represent the signal strength values received from each of the k access points.

Room level occupancy is detected by comparing the observed signal strength from the visible access points with corresponding threshold ranges for each AP. If the RSS value lies within a room's threshold range, occupancy status is set to "present" or else is marked as "absent". Figure 5 illustrates the outcome of this process. With this location information, we now have the time intervals when the occupants were present in their respective rooms. It is important to note that other systems, e.g. a keycard based access control system in a hotel, or a hybrid room occupancy system [1], can also provide this occupancy status in a shared setting. In such scenarios, WiFi RSSI can be used as an additional sensor input to validate that the occupant is present.

Room Set Generation After determining room level occupancy, in this step, we identify the overlapping and the non-

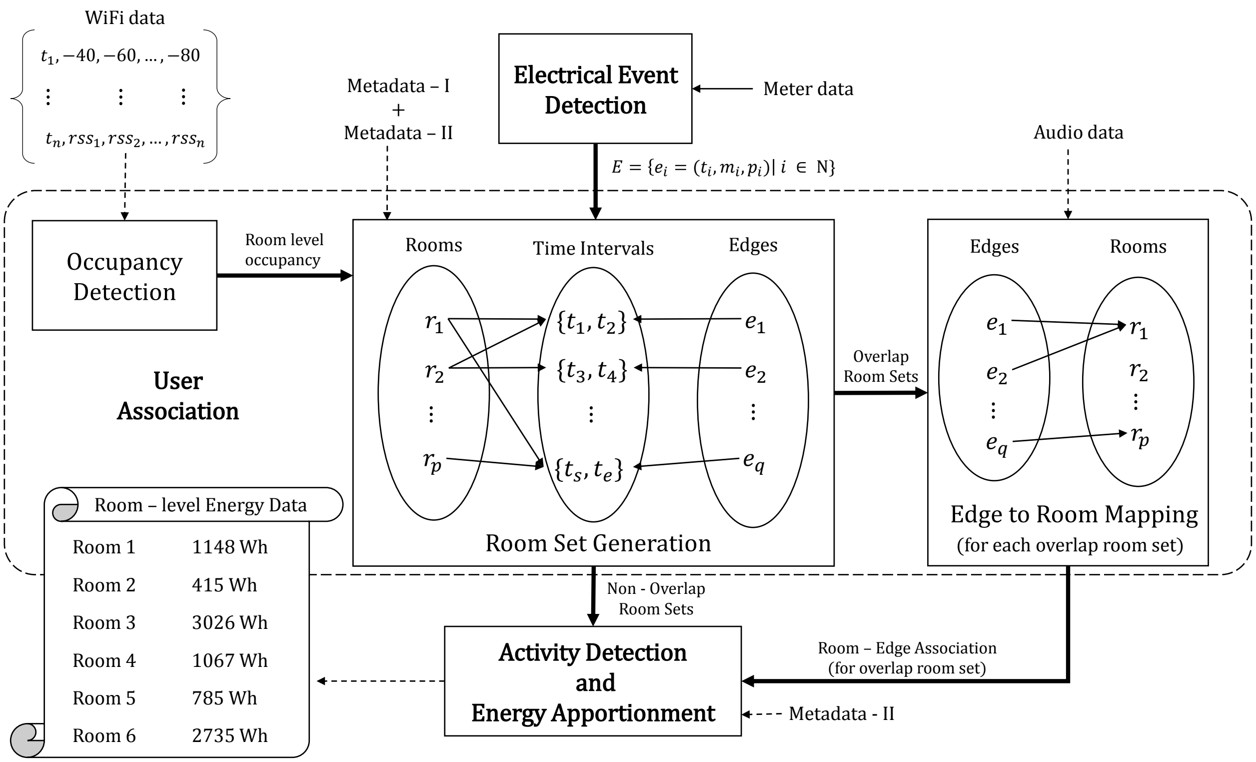

Figure 3: WattShare Algorithm Flow. In this illustration, the three main stages are (1) Electrical Event Detection (2) User Association (3) Activity Detection and Energy Apportionment. The inputs for the algorithm are real power trace from the electricity meter, WiFi and audio data from the smartphone and two types of *Metadata* (explained in Section 3.1.3). The output of the algorithm is the room-level energy data.

overlapping time intervals and associate them with a set of rooms and edges. Here, an overlapping time interval refers to the time period when multiple occupants are present in their respective rooms simultaneously. Thus, the corresponding set of rooms in this interval is called as a '*overlap room set*'. Conversely, the interval when only one occupant is present in the wing is termed as a non-overlapping time interval and the corresponding set of rooms in this interval is called as a '*non-overlap room set*'.

We identify these room sets based on the occupancy information we obtain from the previous step. We then associate all the detected edges that lie within each of these intervals to these room sets by comparing the event time t_e that is associated with each edge. The intuition behind this step is that room sets, when separated into the two categories, make room to edge association easier as multiple edges (contained in *non-overlap room sets*) are automatically associated with the corresponding rooms. This idea is further elaborated below.

Figure 6 shows an illustration of this step. The room sets are shown in curly braces over the overlapping/non-overlapping time intervals. The overlap room set can be formalized as $< (t_{start}, t_{end}) \rightarrow (\{room_i, room_j, room_k\}, \{e_1, \dots, e_n\}) >$ where, (t_{start}, t_{end}) is the overlapping time interval when rooms $\{room_i$, $room_j$, $room_k\}$ are occupied and the events (their associated edges) that occurred within this interval are $\{e_1, \dots, e_n\}$.

Similarly, the non-overlap room set can be represented as $< (t_{start}, t_{end}) \rightarrow (\{room_a\}, \{e_1, \dots, e_m\} >$. Note, here only a single room is occupied during (t_{start}, t_{end}). Thus, all edges are automatically associated to $room_a$.

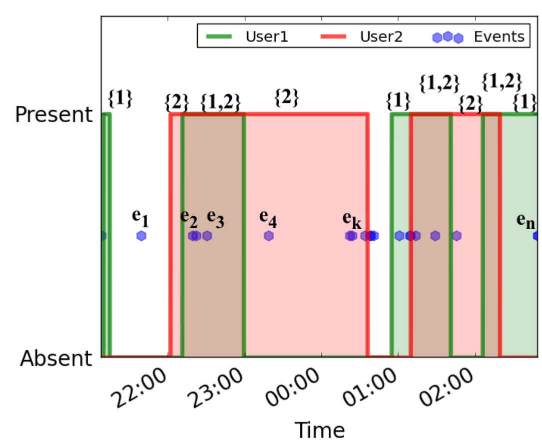

Figure 6: Room Set Generation. Identified overlap and non-overlap room sets are marked within curly braces over the time intervals. Additionally, the edges associated with those time intervals are shown with hexagon shaped dots.

Filtering room sets using *Metadata*: Before moving to the next stage, we filter the room sets to contain only those edges/rooms that match with the stored *Metadata* (see Section 3.1.3). In case of edges in *non-overlap room sets*, the edge attributes i.e. its power magnitude and the phase it is on, are matched against the stored room metadata (using *Metadata - I* and *Metadata - II*). If the metadata matches, the edge-room association is retained or else this association is discarded. Similarly, for edges in *overlap room sets*, metadata for each room in the *overlap room set* is matched against the attributes of the edges contained in that set. In case of a match, the room is retained in the *overlap room set*, else it is discarded. This is repeated for all the rooms in each overlap set. This helps in converting some *overlap room sets* into *non-overlap room sets* wherein the number of rooms is reduced to 1 and consequently, room–edge association is done at this step. The intuition behind this filtering stage is that since the RSS based occupancy detection may not be very accurate, some rooms may get incorrectly associated with certain time intervals. This filtering step will remove some of these incorrect room associations.

Edge to Room Mapping From the previous step, we have already associated some edges (from the *non-overlap room sets*) with the respective rooms that generated them. In this step, we match the remaining edges that are in the *overlap room sets* with the corresponding rooms using MFCC features obtained from the occupant's smartphones (see Section 3.1.2).

For the process of room–edge association in *overlap room sets*, we use the MFCC features to differentiate between the rooms that generated those edges by calculating the Euclidean distance for the event time using the same technique used for calculating *event threshold* (see Section 3.2). The calculated value is then matched with the threshold values generated for all the events for every room. If there is a unique match with one of the *event threshold*, then the corresponding room is associated with the edge. If there is a non-unique match, then the edge is associated with all the matched rooms. If no match is found, then the edge is discarded.

The intuition behind this procedure is that a considerable change in the audio signals is expected when these events, that generate some noise, occur. Figure 7 clearly shows, for some of the events mentioned above, a notable change observed in sound frequency. This change is reflected when the Eucledian distance is calculated between the pre- and the post-event windows.

At the end of this stage, the detected edges are either associated with unique rooms or have been discarded. The outcome can be represented as $< room_i \rightarrow \{e_1, \ldots, e_n\} >$ where i represents the rooms from $(1, \ldots, k)$. With the room associated edges, we can now generate event time slices for the process of activity detection.

3.3.3 Activity Detection and Energy Apportionment

In this stage, we generate event time slices for the edges associated with each room using a simple power magnitude based edge matching algorithm. For each of these time slices, we then identify the appliance that generated those events in the room. A time slice annotated with the corresponding appliance label is referred to as an *activity*. Finally, we calculate the energy consumed by each of these activities to determine the total energy consumption of the room.

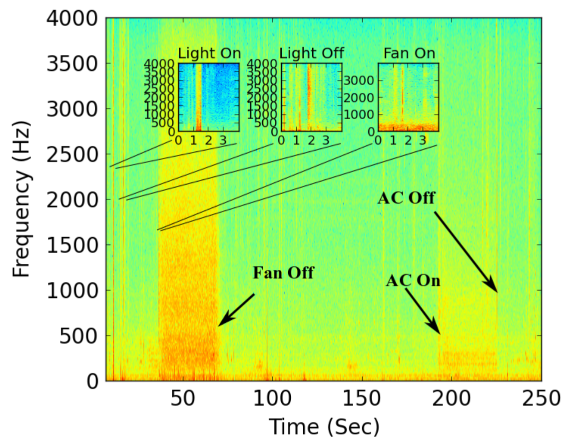

Figure 7: Change in sound frequency during the events of Light ON/OFF, Fan ON/OFF and AC ON/OFF

Before generating time slices, we identify all the edges corresponding to potential fan related events in the edge set for each room. This reduces the error during the edge matching process for similar events (in terms of their power magnitude) like those generated by lights and fans. We use the Euclidean distance based technique (described previously) to identify fan events. The same approach can be extended to other common electrical appliances that have a specific audio signature. After filtering out the fan edges, we then run the edge matching algorithm (described in [17]) on the remaining edges. The algorithm matches rising and falling edges based on similar power magnitude range. For each rising/falling edge pair, a time slice $t_s = (t_r, t_f, mag_t, room_i, phase_j)$ is generated where, t_r is the start time , t_f is the end time, mag_t is the power consumption of the event, $room_i$ is the associated room and $phase_j$ is the phase on which the event occurred.

For all the generated time slices, we annotate them with the appliance that generated the corresponding event. For appliance identification, we match power magnitude of each time slice with the stored values in *Metadata-II*. Note, we require appliance metadata to be accurate in order to perform energy apportionment. As a result, if an occupant brings in a new appliance and does not update the metadata our algorithm will be inaccurate. One potential solution to this is to detect when events are caused by an unknown appliance, localize it to a room, and use that to notify the occupant to update their appliance inventory. We associate the appliance with a time slice if the power magnitude falls within p% of the value stored in the metadata for that appliance, where the value p is an empirically calculated. The value p depends on the appliance in question. For example, we observed that lights varied by 5% of the stored value of 35 Watts (in *Metadata -II*). Similarly, we determine the value of p for other

Model	OS	CPU	RAM
Nexus S	Android OS, 4.1.2 (Jelly Bean)	1 GHz Cortex-A8	512MB
Galaxy Chat B5330	Android OS, v4.0 (Ice Cream Sandwich)	850 MHz	512MB
Galaxy Star S5280	Android OS, v4.1.2 (Jelly Bean)	1 GHz Cortex-A5	512 MB

Table 3: Smartphones used by the participants in the one week deployment period

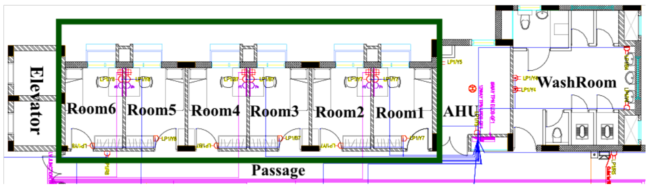

Figure 8: Wing layout in the student dormitory building

appliances. At the end of this step, we have the time slices annotated with the usage duration, identity of the appliance and identified room, thus, generating the list of activities (and hence the energy consumption) apportioned to each room.

4 Experimental Setup

In this section, we describe the setup used to validate our system. The experiments were conducted in a student dormitory building in the IIIT Delhi campus in India. Each floor in the building is divided into 3 wings with six rooms each, that are monitored by two 3–phase electricity meters. The layout of the wing, where we conducted our week long study, is shown in Figure 8. Each room has a fan, light, two plug points and an AC. The 2 sets of 3 phases (from the two electricity meters) are distributed in such a way that each room is served by at least 2 phases, from across both the meters (see Table 1 for more details). Further, each wing has one wireless access point. The presence of multiple WiFi APs that it is typical in the US and other places will only improve the location accuracy given multiple visible APs at each location providing higher fidelity.

Sensors We use Schneider Electric's EM6400 3–phase electricity meters to monitor and collect energy usage information at 1Hz, and store it on a local installation of sMAP [7]. We used power consumption from each of the three phases as an input to the WattShare algorithm instead of the aggregate power consumption of all phases. Smartphones carried by the occupants include: Samsung Google Nexus S, Samsung Galaxy Chat and Samsung Galaxy Star. Hardware specifications are listed in Table 3. All the phones had WattShare data collection mobile app running in the background. The app sampled data from the Wifi radio and the microphone after every 20 seconds interval. Audio data was captured for 10 seconds in every interval and MFCC features were computed over the collected data (more details see Section 3.1.2). Both WiFi data and the audio features were periodically uploaded to the server every 5 minutes and the raw audio data was discarded from the phone.

Ground truth For collecting the ground truth on occupancy and appliance usage, we deployed a sensor mote in every room. Each mote had a PIR Sensor (for capturing motion), Light Sensor (for light events) and Temperature Sensor (for fan and AC events). We also asked the occupants to manually log the ON/OFF times for the activities such as laptop charging, use of fans, ACs and lights performed in their respective rooms. The logged activities together with the data from these sensors were accurate enough to be used as ground truth.

Data Collection Process The data collection was conducted for a week during the winter semester in the month of February. During this time interval, electrical activity that occurred in the dorm rooms were mostly from lights and plug loads. Very few AC and fan activities were observed. Users were asked to carry the phone throughout the experiment week. Plug events usually included charging laptops and phones. We have considered only the laptop charging events as charging a phone was a very low power consuming activity. For encouraging the occupants to carry their phones and to log events, they were offered food coupons at the end of the week as an incentive.

5 Evaluation

We evaluate our system's accuracy by analyzing two critical stages of the algorithm namely, *User Association* and *Activity Detection*. Finally, we report the energy apportionment accuracy by comparing the apportioned energy usage with the actual usage for each of the rooms.

For reporting accuracy, we use the standard measures of precision and recall. *Precision* is the ratio of correctly identified activities to the total number of detected activities. *Recall* is the ratio of correctly identified activities to the total number of activities performed by an occupant.

5.1 User Association Accuracy

In this section, we evaluate the accuracy of the *User Association* stage where the detected electrical events (from the first stage) are assigned to the room that generated them, and therefore, consequently associating it with its corresponding occupant. At the algorithm level, the user association accuracy depends on the performance of the *Room–Edge Association* steps (refer Section 3.3.2) where event edges lying in the *overlap* and *non-overlap room sets* are associated with the corresponding rooms.

Figure 9 shows the precision and recall values for associating all the detected events to the corresponding rooms during the experiment week. The events are classified into two types: events occuring in *overlap room sets* (shown as 'Overlap Events') and *non-overlap room sets* (shown as 'Non–Overlap Events'). We present the association accuracy values for each of these event types. The average precision

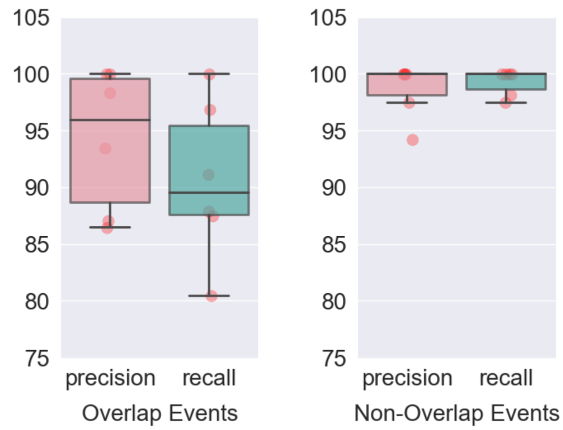

Figure 9: User Association Accuracy. The red bubbles represent the precision/recall values (associated with each room) for each day of the experiment week. The box plots show the maximum, minimum and the average accuracy values across all the days.

and recall of WattShare for the non-overlap events is found to be 98.62% and 99.27% respectively. The overlap events included same appliances being used simultaneously across one or more rooms that shared the same phase (e.g. lights on the same phase being used by both the occupants at the same time) or different appliances on the same phase with similar power magnitude (e.g. AC with the internal fan turned ON and laptop adapter having similar power consumption) being used simultaneously. Even with these complex set of activities, our algorithm is able to differentiate and associate the events with 94.22% average precision and 90.65% recall.

5.2 Activity Detection Accuracy

Figure 10 shows the accuracy for activity detection of each room. The red bubbles represent the accuracy obtained on each day of the experiment week. WattShare is able to accurately detect activities with 93.7% average precision and 91.3% average recall taken across all the rooms.

Activity detection accuracy depends on the performance of the user association component. The main factors that influence its performance are inaccurate differentiation between events using the audio based technique described in Section 3.3.2 and the inability of the current implementation of WattShare to handle complex multi-state behavior exhibited when charging laptops (see Section 6). Other causes of inaccuracy include the occupant leaving his/her phone behind and hence, events occurring during this interval by some other room occupant getting mismatched. The lowest accuracy is observed when most of the event edges lie in the *overlap room set*. This is due to inaccurate event edge association to a room when multiple occupants are present in the wing at the same time and events overlap with respect to time, phase and power consumption. Inspite of these inherent limitations of the algorithm components, we are able to achieve a minimum average (across all the rooms) of 85% precision and recall for detecting activities.

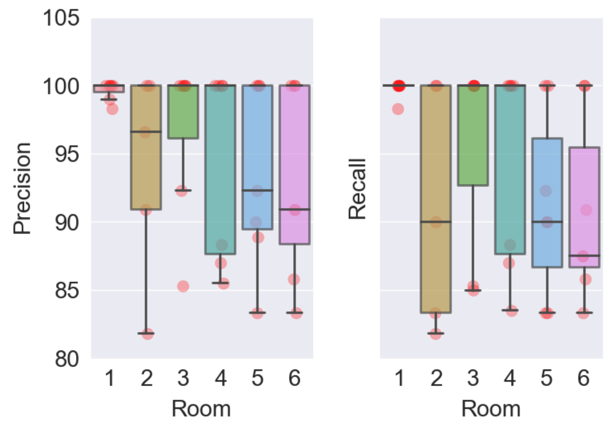

Figure 10: Activity Detection Accuracy. Here, the red bubbles represent the accuracy in terms of precision and recall observed for each day.

5.3 Energy Apportionment Accuracy

We now show the accuracy with which the energy is apportioned to each room. We compare the apportioned energy with the actual energy consumption (obtained from ground truth data) and calculate the estimation error percentage. We calculate the energy consumption by taking the product of the power associated with each activity and the usage duration of the activity. We then calculate the estimation error with the following formula:

$$\text{Error}(\%) = \frac{\text{Predicted Energy(Wh)} - \text{Actual Energy(Wh)}}{\text{Actual Energy(Wh)}} \times 100 \quad (1)$$

Table 4 shows the energy consumption and estimation error(%) for each room separately. We observe that the estimated error for Room 4 is the highest and is the outlier in this set. The reason for such high error percentage is due to laptop's variable power consuming behavior (see Figure 13) – the falling edge with the same power magnitude was not found for the corresponding rising edge when the laptop was put on charge. Due to this, necessary edges were missing in the edge set to perform the edge matching process. This resulted in missing many laptop charging events causing the accuracy to drop (see Section 6 for more details). Figure 11 also illustrates the drop in accuracy for Room 4. To further validate that the accuracy drop was due to the laptop events, we calculated Room 4's accuracy after removing the laptop events. We found that the error reduced from -54.338% to -2.941%, thus, confirming our hypothesis. When taking Room 4's actual accuracy into account, WattShare's energy apportionment component attributes energy usage with 86.42% accuracy on an average. The accuracy after removing the outlier becomes 94.57%.

Figure 11 shows the estimated and true power consumption for the heavily used appliances during the experiment week i.e. light and laptop adapters. ACs and fans weren't used extensively and consisted of less than 3% of all the events and therefore, has not been shown. We found that the

Room#	Predicted (Wh)	True (Wh)	Error (%)
1	1148.40	1150.95	-0.221
2	415.70	413.68	0.488
3	3026.35	2843.18	6.442
4	1067.03	2336.34	-54.328
5	785.16	888.49	-11.629
6	2735.68	2983.69	-8.312

Table 4: Comparison of the estimated and actual energy consumption for every room

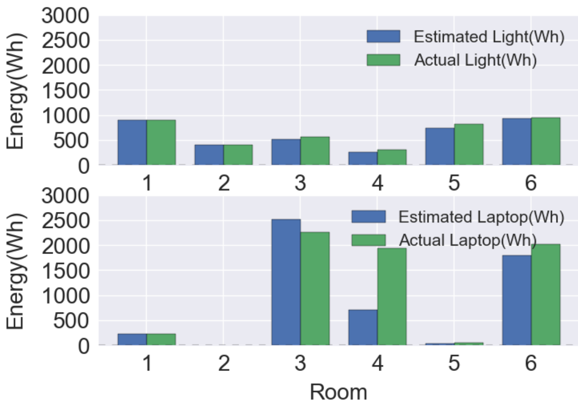

Figure 11: Energy Apportionment Accuracy (for lights and laptop charging events). Usage of fans and ACs were negligible during the experiment week.

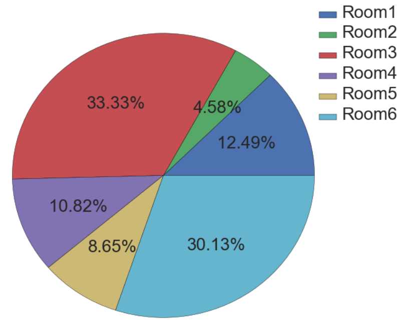

Figure 12: Energy usage distribution in the student wing

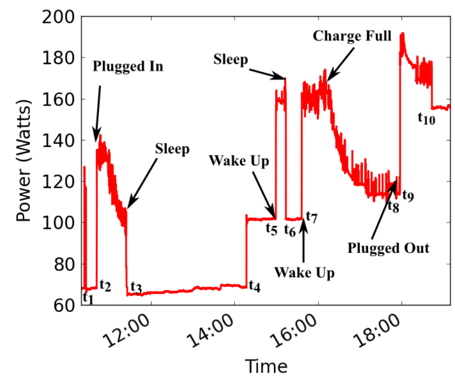

Figure 13: Power Consumption Profile of a Laptop. Most of the errors were caused due to the system's limitation in handling such complex multi-state behavior in appliances. Here, events at t_1, t_4, t_9 and t_{10} are not caused by the laptop.

energy consumption estimates for appliances like lights, fans and ACs is comparable with the actual consumption. However, in case of laptop charging, the estimated consumption varies with large error margin due to the reasons explained above and in the previous section.

Figure 12 shows the distribution of the total energy consumed amongst the wing occupants. From the pie chart, we find that occupant 3 consumed the maximum energy and occupant 2 consumed the least. WattShare allowed us to obtain some behavioral insights from the apportioned energy data such as which occupants stayed most in their rooms, how often did they use appliances such as fans or lights, did they keep their lights turned ON when they left their room and so on. Such insights can be useful in settings such as offices in commercial buildings where HVAC schedules can then be adjusted based on the energy usage behavior of the occupants in these buildings.

6 Discussion

In the previous sections, we saw the design of our WattShare algorithm and its potential in accurately disaggregating and attributing the total energy usage to individual rooms and consequently, to the occupants of those rooms. We now describe some limitations, along with suggestions on how to address them, in the current design of our system that might prohibit its wide scale deployment.

The proposed WattShare approach works well for all the appliances that have a stable power consumption profile such as lights, fans and ACs. However, it fails to determine the usage duration for events caused by appliances with multiple states and dynamic power signature such as charging of laptops. Figure 13 illustrates the laptop power profile as seen in our deployment. When the laptop is put on charge, it draws maximum power reaching its peak consumption of 60 ± 20 Watts until it is fully charged. Once charged, there is a gradual fall in power consumption (t_2 to t_3 and t_7 to t_8) till the laptop is either suspended (seen as a sharp drop at t_3), or the charger plugged out (observed at t_8). This results in edges (rising/falling) with unequal power magnitudes. Laptop usage events such as between $t_5 - t_6$ are detected accurately by our edge matching algorithm. However, since it is crucial to have edges with similar power magnitudes for generating event time slices, we miss some laptop events with variable power draw. Identifying and disaggregating multi-state appliances with variable power draw, including plasma televisions, is a known problem in the NILM community as well. Better NILM algorithms combined with smartphone sensors, using the proposed WattShare system can potentially address this limitation.

Another challenge that is of prime importance is dealing with complex real world scenarios when associating electrical events with rooms. Some examples of such scenarios include - an event taking place in a room when occupants from other rooms are present there; an occupant using an appliance in a different room while the room's occupant is absent and so on. With the current design of the *User Association* stage (responsible for associating events to rooms), some of these scenarios are not appropriately handled. Many such scenarios will arise in large scale real world deployments. We need additional sensors from the smartphone as input sources (or information from additional ambient sensors from the room such as motion or light sensors) to improve WattShare to handle such complex scenarios.

Lastly, the experiments were conducted in settings where only a limited number of appliances were present. We would like to extend and evaluate the efficiency of WattShare for identifying and disaggregating more appliances, such as TV, microwave, refrigerator and others, that are likely to be present in shared living spaces. In addition, we would also like to measure its performance in other complex commercial shared spaces such as offices, where assumptions made for disaggregation in residential settings, don't always hold true [4].

7 Conclusion

In this paper, we present a novel energy apportionment algorithm (WattShare) that leverages the most commonly available sensors on modern smartphones and the increasingly common smart meter. We demonstrate a low cost and scalable system that can be deployed for shared living spaces in commercial buildings such as hotels, dormitories, hospitals, offices and others, wherein a set of rooms are monitored by a single meter. WattShare fuses the context information such as location and audio from the smartphone with the aggregate power from the smart meter to identify electrical events and measure a room's energy consumption. We show that simple localization and audio based event differentiation techniques can achieve highly accurate disaggregation results. Even with the inherent limitations of the algorithm's individual components, WattShare achieves an accuracy of 86.42% for energy apportionment and increases to 94.57% when an outlier room is removed. WattShare attributes the total energy usage to individual rooms with an average precision of 96.42% and average recall of 94.96%.

8 Acknowledgments

Authors would like to acknowledge the support provided by ITRA project, funded by DEITy, Goverment of India, under grant with Ref. No. ITRA/15(57)/Mobile/HumanSense/01. This work was also supported in part by NSF grant SHF-1018632. We wish to thank our anonymous reviewers for their feedback to help improve the paper.

9 References

[1] Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei, and T. Weng. Occupancy-driven energy management for smart building automation. In *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, pages 1–6. ACM, 2010.

[2] P. Bahl and V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 775–784. IEEE, 2000.

[3] N. Batra, H. Dutta, and A. Singh. Indic: Improved non-intrusive load monitoring using load division and calibration. In *Proceedings of the 12th International Conference on Machine Learning and Applications*, volume 1, pages 79–84. IEEE, 2013.

[4] N. Batra, O. Parson, M. Berges, A. Singh, and A. Rogers. A comparison of non-intrusive load monitoring methods for commercial and residential buildings. *arXiv:1404.3878*, 2014.

[5] Y. Cheng, K. Chen, B. Zhang, C.-J. M. Liang, X. Jiang, and F. Zhao. Accurate real-time occupant energy-footprinting in commercial buildings. In *Proceedings of the 4th ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 115–122. ACM, 2012.

[6] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4):357–366, 1980.

[7] S. Dawson-Haggerty, X. Jiang, G. Tolle, J. Ortiz, and D. Culler. smap: a simple measurement and actuation profile for physical information. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 197–210. ACM, 2010.

[8] Y. Guo, M. Jones, B. Cowan, and R. Beale. Take it personally: personal accountability and energy consumption in domestic households. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pages 1467–1472. ACM, 2013.

[9] G. W. Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891, 1992.

[10] S. Hay and A. Rice. The case for apportionment. In *Proceedings of the 1st ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildingss*, pages 13–18. ACM, 2009.

[11] Y. Jiang, X. Pan, K. Li, Q. Lv, R. P. Dick, M. Hannigan, and L. Shang. Ariel: Automatic wi-fi based room fingerprinting for indoor localization. In *Proceedings of the 14th ACM Conference on Ubiquitous Computing*, pages 441–450. ACM, 2012.

[12] J. Z. Kolter, S. Batra, and A. Y. Ng. Energy disaggregation via discriminative sparse coding. In *Proceedings of Advances in Neural Information Processing Systems*, pages 1153–1161. Curran Associates, Inc., 2010.

[13] S. Lee, D. Ahn, S. Lee, R. Ha, and H. Cha. Personalized energy auditor: Estimating personal electricity usage. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications*, pages 44–49. IEEE, 2014.

[14] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao. A reliable and accurate indoor localization method using phone inertial sensors. In *Proceedings of the 14th International Conference on Ubiquitous Computing*, pages 421–430. ACM, 2012.

[15] T. Lim, K. Bae, C. Hwang, and H. Lee. Classification of underwater transient signals using mfcc feature vector. In *Proceedings of the 9th International Symposium on Signal Processing and Its Applications*, pages 1–4. IEEE, 2007.

[16] S. N. Patel, T. Robertson, J. A. Kientz, M. S. Reynolds, and G. D. Abowd. At the flick of a switch: Detecting and classifying unique electrical events on the residential power line. In *Proceedings of the 9th International Conference on Ubiquitous computing*, pages 271–288. Springer, 2007.

[17] M. Saha, S. Thakur, A. Singh, and Y. Agarwal. EnergyLens: Combining Smartphones with Electricity Meter for Accurate Activity Detection and User Annotation. In *Proceedings of 5th International Conference on Future Energy Systems*, pages 289–300. ACM, 2014.

[18] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No need to war-drive: unsupervised indoor localization. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 197–210. ACM, 2012.