MotionSync: Personal Energy Analytics through Motion Tags and Wearable Sensing

Josiah Jee[1], Aveek K. Das[1], Parth H. Pathak[2] and Prasant Mohapatra[1]
[1]Computer Science Department, University of California, Davis, CA, USA,
[2]Computer Science Department, George Mason University, Fairfax, VA, USA
Email: {jjee, akdas}@ucdavis.edu, phpathak@gmu.edu, pmohapatra@ucdavis.edu

ABSTRACT

Individuals with the knowledge of electrical energy consumed by them can take steps to reduce their energy footprint, which can lead to energy conservation. Current energy apportionment schemes either require prohibitively large amounts of user-appliance training or performs poorly in detecting user-appliance interaction when there are multiple users and appliances in close proximity to each other. In this paper, we build MotionSync, a privacy-aware, scalable and robust personal energy analytics system. The system exploits the similarity between the motions of user's arm/hand (captured through wrist-worn wearable) and appliance interface (captured through a motion tag) to determine user-appliance interaction. We show that commonly used plugload devices can be classified in five categories based on their interfaces - button, door, free-floating, knob and switch. Based on this, it is possible to train a generic machine learning model for each category to detect user-appliance interaction with a significant lower training overhead. MotionSync is privacy-aware and allows users to measure their own energy consumption without sharing any private information with building infrastructure. MotionSync is also robust to crowded scenarios since it does not depend on user's location. We implement and evaluate our system on a real testbed, and find that it can determine user-appliance interaction with an average accuracy of 92.5% and has low average false positive rate of 8.6%. We also show that pretrained models of five interface types provide very high accuracy even for new and untrained appliances and users, eliminating per-appliance and per-user training overhead.

CCS Concepts

•**Information systems** → *Information systems applications;*

Keywords

internet of things, motion matching, personal energy apportionment, smartphones, smart meters, wearable sensors

1. INTRODUCTION

Determining the amount of energy consumed by each individual user in a home or an office (also referred as energy apportionment) is crucial in energy conservation and reducing carbon footprint. Accurate energy apportionment and eco-feedback have shown to reduce the per-user energy footprint by a considerable amount (7.6% - 19.4% for home energy usage [1]).

Unfortunately, energy apportionment is a challenging problem encompassing many complex aspects. It requires accurate determination of usage of different appliances by users. Many of the current solutions [2, 3] rely on tracking a user's location in a building to estimate her proximity from different appliances. The location based approaches perform poorly in the case of multiple appliances and users close to each other due to insufficient accuracy of current localization techniques. Additionally, they can reveal user's location and activity traces to the building, reducing user's motivation to subscribe to energy-feedback systems. Other approaches [4] based on user-object interaction can provide location independence. However, they require a large amount of training for each appliance and user, making them difficult to deploy for everyday usage in homes and offices.

With the increasing popularity of wearable mobile devices, it has become possible to sense and infer a wide variety of user activities. Similarly, due to advances in ubiquitous computing and Internet-of-Things (IoT), more and more everyday devices/objects are becoming smarter. In this paper, we present a novel energy apportionment technique that leverages user's wrist-worn wearables and appliances equipped with motion sensors. We show that there is a strong similarity between the motion of user's hand/arm and the motion of appliance's interface when the user operates the appliance. The motion can be captured using low-power, low-cost IMU sensors (accelerometer and gyroscope) in the motion tags which can be deployed on the interface (e.g. switch, knob etc.) of the appliances. Most current wrist-worn devices such as smartwatch or fitness trackers are already equipped with the IMU sensors. The motion matching can be used to detect user-appliance interaction, which can then be integrated with the information available through the smartmeter for accurate energy apportionment.

In this work, we design and implement MotionSync, a personal energy analytics system that leverages motion matching for detecting user-appliance interaction. It has three salient features - (1) MotionSync does not require user's location to detect interaction with an appliance. This independence from location allows accurate determination of user-appliance interaction events even when there are multiple users and appliances in a close proximity (for example, users in an office kitchen during lunch time). (2) MotionSync is private, which means that it does not reveal any user information such as her location traces, history of appliance usage etc. to any entity in the building. Instead, with the help of building sensors (motion tags on appliances and building smartmeter) and personal devices (wearable and smartphone), it enables a personal energy auditing. (3) MotionSync significantly reduces the required

BuildSys '16, November 16-17, 2016, Palo Alto, CA, USA
 ISBN 978-1-4503-4264-3/16/11. . . $15.00
DOI: http://dx.doi.org/10.1145/2993422.2993572

training by eliminating per-appliance and per-user training. It relies on generic training that is only required for the category of appliance interface.

Our work is the first step towards a far-reaching research objective of creating an accurate and personal energy analytics system where a user can track her own energy consumption on a daily basis across multiple buildings (home, office etc.) in a privacy-preserving manner. In this paper, we approach the energy apportionment problem through the point-of-view of user-appliance interaction. Although this approach is effective in cases where an appliance is serving only one user at a time (for example, a user heating up food in a microwave), it cannot properly apportion the energy when the appliance is serving multiple users (for example, a user switches on lights in a conference room with many other users) without integrating additional information. To the best of our knowledge, this limitation exists in majority of current apportionment schemes including the ones which depend on user location to detect appliance interaction. In this work, our objective is to make the user-appliance interaction-based apportionment more robust (multiple users, appliances in close range), private and scalable (significantly reducing training overhead).

The contributions of the work can be summarized as follows -

1. We propose categorization of commonly used plug-load appliances into five classes based on the type of their interface - *button, knob, switch, door and free-floating*. Because users interact with different appliances of the same class in a similar way, this categorization enables us to develop one generic machine learning model for each category. It also eliminates the need of separate training for each new appliance and user.

2. We show that when a user interacts with an appliance, for a very short time duration, the motion of user's arm/hand exhibits strong similarity with the motion of appliance interface (e.g. switch, door). This observation can be used for detecting which user interacted with which appliance. We provide a motion matching framework where low-power motion tags can be deployed on appliance interfaces to measure their motion and compare it with the motion of user's wrist-worn wearable.

3. We implement MotionSync as a practical system using compact embedded boards as motion tag, wearables and smartphone, and evaluate it using 24 appliances and over 1400 user-appliance interactions. We find that MotionSync can determine user-appliance interaction with an average accuracy of 92.5% across all appliances, users and interaction events with average false positive rate of 8.58%. Using generic models, the average accuracy is observed to be 90.1% and 90.5% for untrained appliances and untrained users respectively.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 provides an overview of our MotionSync with its design goals. Section 4 outlines the motion matching algorithm and its integration with smartmeter data for energy apportionment. Implementation details and evaluation of MotionSync are discussed in Section 5. We discuss the relevant insights, limitations and scope for future work in Section 6.

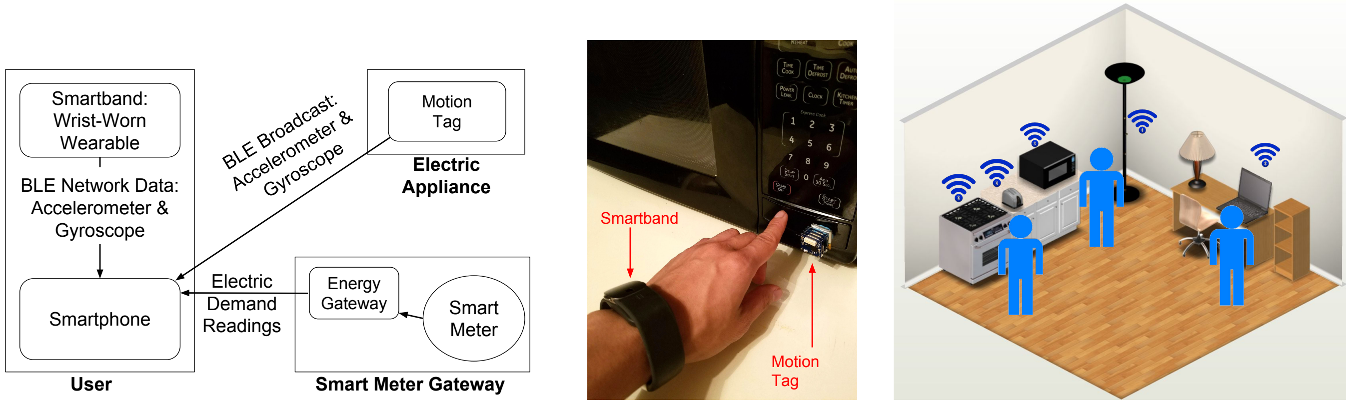
2. RELATED WORK

Energy Disaggregation: Disaggregation is an important problem within the field of energy conservation and analytics. Energy disaggregation methods can be separated into two main categories of Intrusive Load Monitoring (ILM) techniques [5] and Non-Intrusive Load Monitoring (NILM) techniques [6]. These methods are helpful in Appliance Load Monitoring (ALM) which is essential in energy management solutions. ILM methods rely on *intrusive* (located within the living environment) appliances that use a low-end metering device, while NILM methods generally rely on using machine learning to disaggregate the energy usage from a centralized meter panel. Since NILM requires a pre-training phase, it is difficult for any NILM method to perform well in discerning all types of different appliances having differences in size, make or manufacturer. Most energy disaggregation approaches attempt to detect the appliance states or state transitions to aid in energy disaggregation [7–12].

One approach uses smartphone acoustic sensors in conjunction with NILM methods to determine the ON/OFF states of appliances to aid in energy disaggregation [7]. However, the appliances covered by acoustic sensors are limited to noise generating appliance such as fans, heaters, dryers and washers and also compromises the privacy of the user by using an always-listening microphone sensor. Some other approaches use sensors that measure electromagnetic field changes to determine state transitions and the user of appliances [8, 9]. In [13], the authors have used deep neural networks for disaggregation, whereas [14] has used unsupervised clustering to improve electric appliance-based event detection. Some research works have used information regarding users' activities of daily living (inferred from sensors on users' smartphones) and users' indoor location (inferred from Wi-Fi signal strength data) [15, 16] to reduce the search space of electrical appliances considered for disaggregation. In [10], authors make use of smart plug sensors, perform feature extraction and apply machine learning to estimate appliances states for detecting user interaction events. Real-time measurement of power consumed by a plug load has been proposed in [17] which attaches a small unobtrusive sensor directly to the plug and wirelessly transmits the power data to nearby smartphones.

Energy Apportionment: MotionSync's primary objective is the *apportionment* [18] of energy usage to users; this is different from energy disaggregation's primary objective of mapping energy usage to appliances. Our system seeks to determine a user's *personal energy usage* for plug load devices within the home, which provides a personalized feedback of energy usage. A number of research efforts have explored the challenge of apportioning energy usage to individual users in home or office settings. The main challenge of apportioning energy usage to users is determining the match between a user and an appliance usage event. The general approach to matching a user and an appliance is done by tracking a user's location within a building or office [2, 3, 19–21]. The authors in [22] use wrist worn magnetic radiation sensors to detect the user interaction with devices based on the change in the surrounding magnetic field.

The location-based systems match an appliance to a user based on their proximity and rely on tracking a user's location within a home or office environment either by RFID sensors [19, 20], tracking sensors within the home [3], or by WiFi localization [2, 21]. Many of these localization methods depend on building sensors to track user's location, creating a privacy leakage issue where the user's location, activity trace, and interaction with appliances can be monitored by the building. Additionally, state-of-the-art WiFi-based localization accuracy is not always sufficient to distinguish between two or more appliances or users within the range of inaccuracy. MotionSync attempts to provide personal energy usage feedback to the user while preserving the *privacy* of the user without the need of the user's location. In [3], the challenge of multi-



(a) Overview of MotionSync

(b) User with a wrist-worn smartband, interacting with an appliance

(c) Multiple users in the room receives broadcast from appliance motion tags, however, motion matching ensure correct detection of user-appliance interactions

Figure 1: MotionSync overview, sample user-appliance interaction and multi-user scenario

ple users present in the same room is handled by the utilizing the knowledge of the proportion of time for which the user is present in the area to the total operating time of the device. This implies that users that do not cause an usage event will be penalized with energy usage for being in the vicinity of an appliance during its use. In [19] and [21], a study set of two people were used in the experiments. In the presence of increasing number of non-users, our system will be able to determine the correct user because of the uniqueness of the matching motion between the true user's wrist-worn wearable and appliance's motion tag.

Wrist-worn Motion Sensors: Our system relies on the matching of accelerometer and gyroscope data in order to determine user-appliance interaction. Many research projects have focused on utilizing motion data from a wearable device to provide new and interesting insights. In [23, 24], wrist-worn motion data is used to provide finger-hand gesture recognition and remote interaction through arm tracking, whereas [25] explores how motion sensors from users' smartwatches can reveal what an user is typing on a keyboard. In [26] the authors build a model to recognize smartphone hotwords (like, Okay Google) by utilizing the sensitivity of accelerometer to users' voice. Chang et. al uses accelerometers embedded in television remote controls or mobile devices with machine learning in order to identify specific household members [27] and Ranjan et. al proposes the use of wrist worn sensors to identify object-user identification [4]. These method rely on training a model for numerous features for every unique user. MotionSync seeks to create a method that ensures accurate user-appliance identification but is not reliant on training per user or per appliance.

3. SYSTEM OVERVIEW

In this section, we describe the design goals, discuss the main idea behind MotionSync, and provide its overview.

3.1 Design Goals

Our personal energy analytics system is designed to meet the following goals. These goals are related to the limitations of previous approaches as discussed in Section 2.

(1) Limited Training: Current models of user-appliance interaction identification can be used for energy apportionment. However, they scale poorly due to their dependence on training required for each new appliance and user. Our goal is to develop a system that can substantially reduce the required training by eliminating the per-appliance and per-user training.

(2) Private Analytics: Previous research on energy apportionment heavily relies on tracking user's location and actions through sensors deployed in the buildings. This can reveal private user information such as location, which appliance a user used, activity traces etc. in homes and offices. Our goal is to design a private energy analytics system where a user can determine her own energy footprint through the help of sensors deployed in the building, without revealing any private information beyond her own devices.

(3) Multi-user Scenarios: Current approaches of energy apportionment often rely on user's location to determine user-appliance interaction. These approaches are not only limited to stationary appliances, but also to a single user within the localization range of accuracy. With state-of-the-art localization accuracy, they also cannot accurately apportion energy in the cases with multiple users close to each other in a room (e.g. two users in a kitchen). Since such multi-user scenarios can be very common in homes and offices, our goal is to design an accurate energy apportionment technique that does not rely on location estimates.

3.2 Approach

To meet the design goals mentioned above, we propose a novel *personal energy analytics* scheme through matching the motion of a user's arm/hand and the appliance the user interacted with. We show that appliances can be equipped with *motion tags*, and the motion captured by the tag when a user interacts with the appliance can be matched with the motion of the user's arm/hand captured by her wrist-worn wearable. With increasing adaptation of Internet-of-Things (IoT), it has become possible to embed sensors in everyday objects such as appliances. Fig. 1a shows an overview of MotionSync. Here, the motion data captured by user's wrist-worn wearable (referred as smartband here onward) can be transmitted to the user's smartphone. Similarly, when a user interacts with an appliance, as seen in Fig. 1b, the motion tag on the appliance *broadcasts* the observed motion data. The broadcast can be performed using any low-power, short-range wireless communication standard such as Bluetooth Low Energy (BLE). This data is also received by user's smartphone, which in turn compares the motion tag data with the smartband data to determine if the owner user interacted with an appliance or not. This user-appliance interaction can then be combined with the building's smartmeter data to apportion the appliance's energy usage to the user (more details about apportionment in Section 4.4).

One main advantage of MotionSync's motion matching approach is that it does not require per appliance or per user training. Instead, the training in MotionSync can be generalized to per *category of appliance interface*. As we will discuss in Section 4, motion matching is accomplished through training for five appliance interface

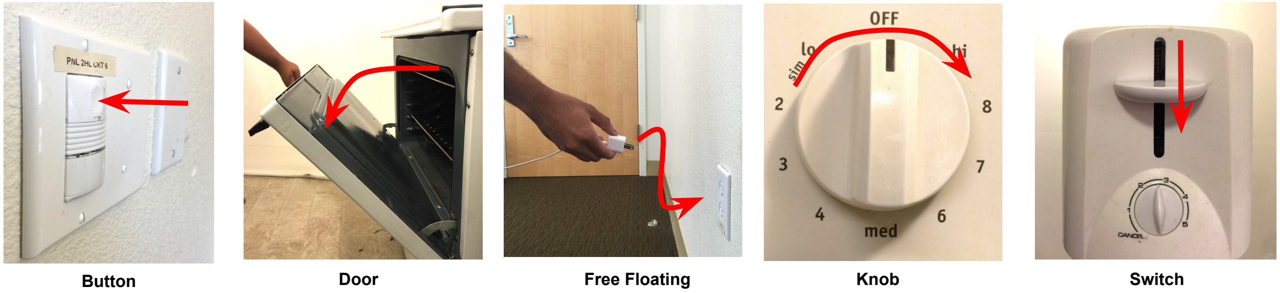


Figure 2: Appliances are categorized in five categories based on their interface type. The ways in which users typically interact with an appliance of each category is depicted using a red arrow

types - button, door, free-floating, knob and switch. By separating training on different types of appliances, the presented model substantially reduces the amount of training required from per user, per appliance to simply per appliance type.

MotionSync is *privacy-aware* because it does not reveal any information about a user's interaction with an appliance to any entity in the building. Specifically, when the appliance tag transmits its motion data, it does so through broadcast. This ensures that the user's smartphone can receive the data without establishing a connection with the motion tag (which could lead to the tag knowing/remembering user/phone identity). Although this broadcast can be received by other users also in the vicinity, it will not result in a match with the motion of their smartband. Once a user's smartphone has detected that the user interacted with an appliance, it can lookup the smartmeter data to determine the energy usage of her appliance interaction (more details in Section 4.4). Also, our system is different from other approaches [3,19,21] where the building measures the energy consumption of users by tracking their location and/or proximity to appliances. MotionSync is *personal* energy analytics system where a user can choose to install the smartphone app and subscribe to measuring her personal energy consumption locally on her smartphone.

Additionally, the system is effective in scenarios with many users in the vicinity of an appliance. For example, location-based approaches [3, 19, 21] perform poorly in cases like home or office kitchen where there can be many appliances and possibly multiple users depending on time of the day. An example of multiple users and appliances is shown in Fig. 1c. It can be observed that even when multiple users are receiving motion data from all appliances, the system matches only the specific user who is interacting with the appliance because of motion matching with user's smartband. As a result, accurate apportionment can be performed.

We note that MotionSync can allow a user to accurately estimate personal energy usage *anywhere* and not limited to home environments. We envision a scenario where a user can estimate personal energy usage not only at home but also at work and leisure environments, given that appliances are equipped with motion tags and smartmeter data is available (possibly through a cloud service). However, our evaluation is primarily concerned with homes and small office scenario with a focus on plugload devices (appliances, electronics, lighting etc. [28]). In its current form, the system equally apportions the energy consumption of other types of systems including air conditioning, heating etc. and any other systems that are not directly controllable by user's interaction with a device or an appliance.

4. MATCHING ALGORITHM

MotionSync relies on low-cost, low-power accelerometer and gyroscope sensors commonly available in today's wrist-worn wearables such as smartwatches or fitness trackers. The motion tags also includes the two IMU sensors for capturing the appliance motion. The objective of the motion matching process is to accurately match the accelerometer and gyroscope data available from appliance motion tag and user's smartband.

Button	Light and Fan Button, Microwave Coffee Maker, Desktop Button
Door	Oven, Washer, Dryer
Free-Floating	Phone and Laptop Charger, TV Remote, Hair Dryer
Knob	Stove Knob, Espresso Machine, Dishwasher, Convectional Oven
Switch	Light and Fan Switch, Toaster, Rice Cooker, Water Heater

Table 1: Electrical appliances under each of the five interface types

The motion matching is complex because even though the appliance tag observes motion only when it is interacted with, the user's wrist IMU sensors constantly register motion due to a user's arm/hand movements. We address this using an observation that for a short duration of interaction (i.e. user pressing a light switch), the motions of appliance tag and user's arm are similar. When carefully matched, they can accurately reveal a user-appliance interaction event. The magnetometer sensor is not used in our system, as we observe that the use of electric appliances result in significant change the magnetic field readings.

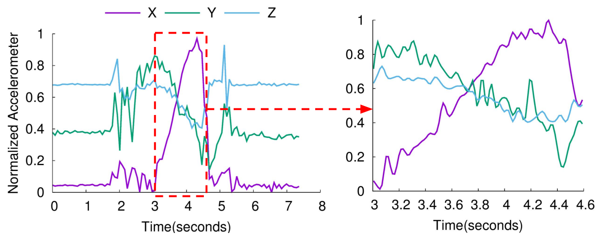
To eliminate per-user and per-appliance training, we categorize the appliances based on their interface type as discussed next.

4.1 Appliance Categories

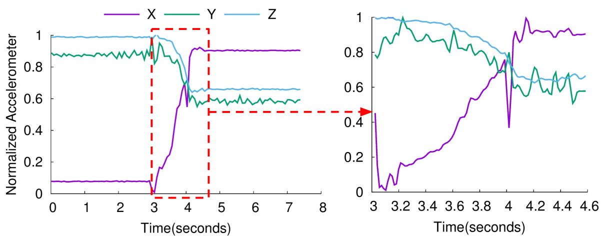
In MotionSync, we group the appliances into five different types of interfaces: (1) button, (2) door, (3) free-floating, (4) knob and (5) switch. We find that most common appliances found in homes can be categorized in one of these interface types as shown in Table 1. The intuition behind this interface type based categorization is that it captures the underlying nature of motion when users interact with appliances, allowing us to reduce the training to five different interface types.

The motion tag can be attached on the interface (i.e. knob of a dishwasher or door of a dryer) to make sure that there is sufficient similarity between the motion of the tag and user's arm/hand. Note that the motion tag is only used to detect user-appliance interaction. It is not used for detecting the switching ON/OFF of the appliance. MotionSync relies on the smartmeter data for determining when the appliance of switched ON and OFF (Section 4.4). If an appliance possesses multiple different types of interfaces (e.g. switch and knob on a toaster), the motion tag is affixed to any of the interfaces that the user interacts with. In some cases, using the interface which triggers the ON/OFF state change of the appliance can be advantageous (such as the pull down switch of a toaster instead of the duration knob seen in Fig. 2) for easier integration with smartmeter readings.

We now define the five interface types and the characteristics of

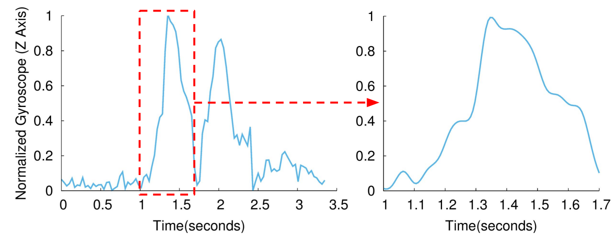


(a) User smartband motion

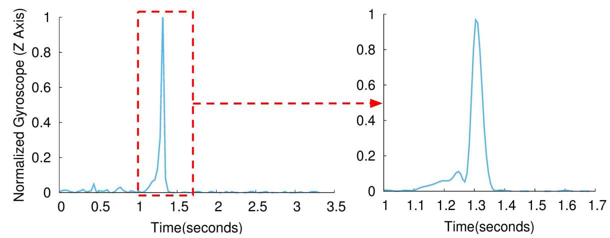


(b) Motion of tag on appliance

Figure 3: Motions of user's smartband and tag on appliance when user interacts with the **knob interface** of a gas stove. The figures on the right show accelerometer motion of both sensors for exact period of interaction, whereas the figures on left show motion for a longer time period before and after the interaction.

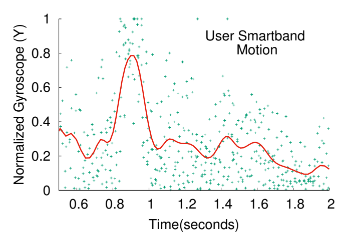


(a) Motion of user's smartband

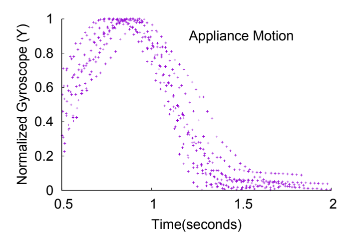


(b) Motion of appliance tag

Figure 4: Z-axis gyroscope motion of user's smartband and appliance motion tag when user switches on a light **button**. The figures on the right show motion of both sensors for exact period of interaction whereas, the figures on left show motion for a longer time period before and after the interaction.



(a) Users' smartband motion



(b) Appliance tag motion

Figure 5: Smartband and motion tag gyroscope readings for 10 user interactions with the **door** of an appliance

resultant motion. Fig. 2 shows corresponding example of appliance of each type.

(1) Button: A button type interface is characterized by a small in and out motion. An example of this is seen in a microwave button or small tactile buttons on a coffee maker.

(2) Door: A door type interface is characterized by a large smooth rotary motion. An example of this is seen in an oven or a door of a washing machine.

(3) Free-Floating: Free floating is a class of interfaces that does not rely on a consistent pattern of motion between the user and appliance. The way the user interacts can vary and is characterized by unconstrained motion. An example of this can be a hair dryer, laptop charger or a TV remote.

(4) Knob: The knob type interface is a class of interfaces characterized by a quick, small rotary motion. An example of this type of motion is seen in most stove tops.

(5) Switch: A switch type interface is characterized by a small up and down motion. An example of this is seen in a standard light switch or on a toaster.

It is worth noting that the same appliance or device (for example a light) can have button as its interface in one building and switch in another. However, because MotionSync simply relies on the type of interface (not type of appliance), user-appliance interaction can still be detected simply by choosing appropriate trained model of the given interface type.

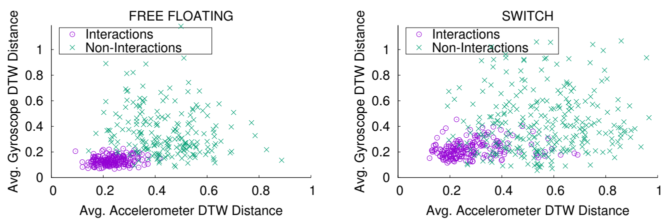
4.2 Exploiting Similarity in Motion

In this section, we show that the motion observed on a user's smartband and the appliance tag are similar within the short duration of user-appliance interaction. Accelerometer data is oriented under the earth's axes for the user and the appliance tag. This is done to keep gravity from polluting accelerometer data when the user and appliance experience different orientations.

Fig. 3 shows the accelerometer data for the motion tag and user's wrist-worn smartband when a user rotates the knob of a gas stove. It can be seen that the observed acceleration in 3 axis for the motion tag remains unchanged before and after the motion. On the other hand, user's arm/hand movements before and after the interaction can result in large changes in acceleration for user's smartband. However, we observe that for the short time duration of interaction (from 3s - 4.6s), both accelerometer sensors exhibit noticeable similarity. Similarly, Fig. 4 plots Z-axis gyroscope data for tag and smartband when a user pushes a button (shown in Fig. 2) to switch on a light. Due to the relative location of the tag and the button, pushing of the button results in a rotary motion, which however small, is captured in a gyroscope axis. In Fig. 4, it is also observed that even for a short period of interaction (e.g. pushing a button), significant similarity is observed between tag and wristband motions.

Motion Tag Broadcast: As discussed in Section 3.2, the appliance motion tags broadcast their accelerometer and gyroscope data, which is then received by smartphones of users in close proximity. To save energy, the tags can only broadcast the data when an interaction event happens and refrain from broadcasting when they are idle. From Figs. 3b and 4b, it can be observed that acceleration and speed of rotation remain unchanged when the appliance is idle. MotionSync utilizes a simple threshold to detect when the appliance is being interacted with. When the threshold (for accelerometer and/or gyroscope) is crossed, the tag starts broadcasting the sensor data. The broadcast stops when the sensor data is observed to be lower than the threshold for a certain amount of time.

Robustness of Motion Similarity: We further evaluate the similarity between tag and wristband motion by studying its robustness



(a) Free-Floating (b) Switch

Figure 6: Comparison of computed DTW distances for interaction and non-interaction events in appliances with **free floating** and **switch** based interfaces

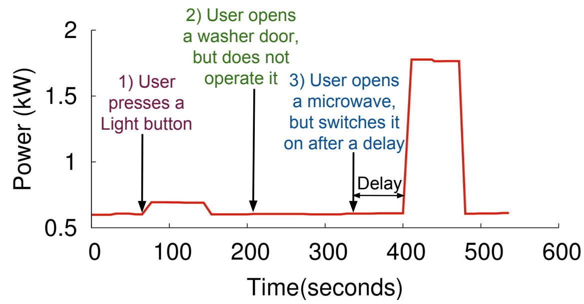


Figure 7: Smartmeter power data variation based on three controlled user experiments - 1) pressing a light button, 2) opening washer door but not switching it on and 3) opening a microwave and switching it on after a minute's delay.

over multiple user-appliance interaction events. Fig. 5 shows the gyroscope data for 10 different events of user's interaction with the door of a washing machine. Note that in all experiments presented throughout the paper, users interact with the appliance in their natural and unconstrained manner. It can be observed from Fig. 5 that even though there are more similarities between different tag events than there are between different user events, the general motion of the tag and the user show similar patterns. Higher variance within a user's hand motion is expected as hand motion is more volatile and random than a static tag. Despite the higher variance in user data, the trend of the hand motion is consistent over multiple instances of interaction. This means that the similarity between tag and wristband motion can be exploited for robust detection of user-appliance interaction.

4.3 Dynamic Time Warping and Learning

Once the accelerometer and gyroscope data from user's smartband and appliance motion tag is available on user's smartphone, the motion matching procedure is initiated. In order to quantify the similarity in motion, MotionSync uses Dynamic Time Warping (DTW). Specifically, it calculates six DTW distances (for 3-axis accelerometer and 3-axis gyroscope) for the sensor data available from smartband and appliance tag. DTW is especially suitable for motion matching in our system because the data from smartband and motion tag can be loosely synchronized and collected at different sampling rates. DTW's ability to compare two time series data even when they are not perfectly synchronized allows us to perform motion matching.

Fig. 6 shows the effectiveness of DTW distances in motion matching. It plots DTW distances (averaged for 3 axis of accelerometer and gyroscope) for over 400 user-appliance interaction events and 500 non-interaction events. For the two appliance categories (free-floating and switch), these interaction events were tested for multiple appliances and users. We define a non-interaction event as an event where when a user is not using the appliance but is in the proximity that the user's smartphone receives the BLE broadcast from the appliance tag. The non-interaction events are created by collecting accelerometer and gyroscope data from the smartband of different users over multiple days, and uniform randomly choosing time periods when a user was not interacting with any appliance (i.e. performing any other activity). It can be observed that the average DTW distances between interaction events are consistently lower than the average DTW distances of non-interaction events.

MotionSync utilizes the accelerometer and gyroscope DTW distances (a total of six features) to learn a machine learning model of interaction for each appliance category. As we show in Section 5, the model can be trained using data of user interactions with sample appliances of a category and can be used for detecting user-appliance interaction events for other appliances of the same category. This eliminates the requirement of training for every new appliance and user, substantially decreasing the barriers of widespread system deployment.

4.4 Energy Apportionment through Smart Meter Integration

Once a user's smartphone matches the motion of user's wristband and the appliance's motion tag, the smartphone will use electric power consumption information available from the smartmeter of the building to determine the amount of energy to apportion to the user. The appliance motion tag broadcasts, in addition to its IMU sensor readings, the average power rating (kW) of it's corresponding appliance. This rating information can be hardcoded in the tag by the device manufacturer. When an interaction event is detected, MotionSync will use smartmeter energy readings after the time of the detected interaction. Each event has different gains and drops in electric demand depending on the power draw of the appliance being used. With the power draw of the appliance known through the broadcast, MotionSync can calculate the total energy used in that particular interaction.

In Fig. 7, we show the variation of smartmeter power readings for a 10 minute duration of controlled experiments. We utilize the Rainforest Eagle [29] to collect and stream smartmeter data in a home. The Rainforest Eagle is an Ethernet gateway that connects to a smartmeter and provides energy usage in real time (one sample every eight seconds). We discuss three important cases in terms of smartmeter data after a user-appliances interaction is detected as follows -

(1) Immediate increase: The power demand can increase immediately after a user-appliance interaction event is detected. This is shown in Fig. 7 where a user pushes a light button to turn on the light. This results in immediate increase of 100 Watts in the smartmeter readings. Note that user's smartphone, which successfully detected a user-appliance interaction, also received a power rating of approximately 100 Watts broadcast from the appliance tag. The smartphone can then use the motion matching, power ratings and smartmeter readings to apportion the energy to the user.

(2) No increase: It is possible that user's smartphone detects a user-appliance event, receives the power rating broadcast from the appliance tag, but there are no corresponding changes observed in the smartmeter. This is possible when a user interacts with an appliance but does not necessarily switches it on (no change in electrical state). As shown in Fig. 7, a user can open a washer door, but does not switch it on. In this case, the smartphone can monitor the smartmeter for a certain amount of time and if the increase of expected power usage does not occur, no energy usage is apportioned to the user.

(3) Delayed increase: In many cases, a user can interact with an appliance and switch it on after a certain delay. Fig. 7 shows an example where a user opens a microwave to heat up the food, however, the actual switching on of the microwave (increase in power usage) happens after a certain delay. The broadcast power ratings can be used for apportionment as long as the interaction and switching on events occur within a certain amount of time.

A user-appliance interaction is also detected when user switches off the appliance (e.g. turning off lights, removing food from microwave etc.). However, there will be no corresponding increase in the smartmeter data due to this action. Hence, such events are not used in the apportionment. The energy consumed for an interaction is calculated from the duration of active time followed by an immediate or delayed increase in demand.

We note that the above mentioned apportionment technique works even when there are multiple user-appliance events occurring simultaneously. As an example, consider two users Alice and Bob interacting with microwave and coffee-maker respectively. Both the users receive motion data from both the appliances, however, due to motion matching with smartband and power rating information, correct user-appliance interaction can be accurately determined. It is worth noting that both, motion matching and smartmeter integration procedures, does not have to performed in real time. This means that user's smartphone can collect the appliance tag data in real time, but the smartband motion data and smartmeter readings can be available at a later time for motion matching and apportionment. The procedures can be carried out once a day during low activity periods (e.g. charging of phone at night).

5. EVALUATION AND RESULTS

In this section, we study the effectiveness of MotionSync from the point of view of the design goals summarized in Section 3. Overall, we study how accurately the system detects the user interaction with the appliance - without new training for every additional appliance and user.

5.1 Experimental Setup

In this section, we describe the experimental setup for the evaluation of our system. As observed in Fig. 1a, MotionSync requires a motion tag on the electric appliances, a smartphone that receives broadcast data from the tag, a wrist-worn smartband to record the user's motion, and smartmeter readings from the building where the system is evaluated.

TinyDuino Motion Tag for Appliances: For the motion tag deployed on the interfaces of the electric appliances, we use the TinyDuino platform [30]. TinyDuino is a miniature open-source modular electronics platform that is compromised of a processor board and multiple TinyShields (modules) which add specific functionality. MotionSync's motion tag utilizes a 9-axis IMU TinyShield for motion data and a BLE TinyShield Nordic chipset to broadcast the motion data over BLE. We modify the BLE advertisement packets (broadcast) on the tinyshield platform to include the accelerometer and gyroscope data in each packet. In addition, we use a module to connect a rechargeable battery to the tag. In our experiments, we affix the motion tag to the appliances' interface in order to capture motion data of the appliance interface as seen in Fig. 8. We program the TinyDuino tag to sample and broadcast accelerometer and gyroscope data at 40 Hz. We fix the motion tag on appliances in such as way that its orientation matches the orientation of smartband when user is standing and her arm is pointing towards the floor (in rest position).

User Wearable and Bluetooth Sniffer: We use a Google Nexus 5 smartphone, affixed to the user's wrist to collect the sensor data for users' motion. We collect the accelerometer and gyroscope data using the AndroSensor app [31] at 40 Hz. To simulate the smart-

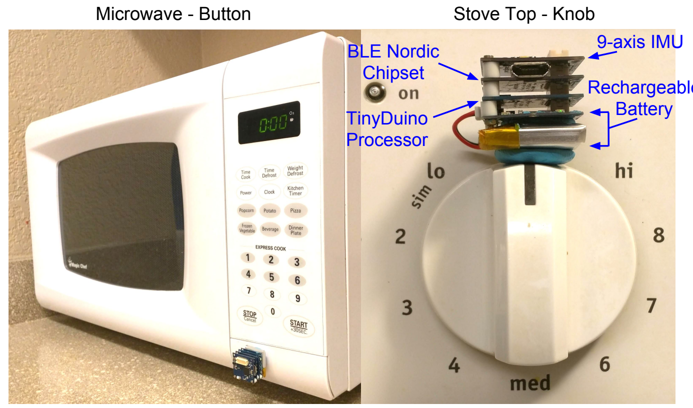


Figure 8: Motion tags: We affix a TinyDuino motion tag to the appliance at the point of user interaction with the appliance - the microwave open button and stove knob. The motion tag on the stove shows the 4 modules of the TinyDuino platform.

phone sniffing Bluetooth packets, we utilize a ComProbe BPA 600 Dual Mode Bluetooth Protocol Analyzer [32]. The ComProbe, in BLE mode, can capture BLE advertising or broadcast packets on all three advertising channels. We use a laptop that collects the tag broadcast as well as the smartband data, and run MotionSync to detect user-appliance interaction events. Energy is apportioned after integrating the smartmeter readings.

5.2 User Experiments

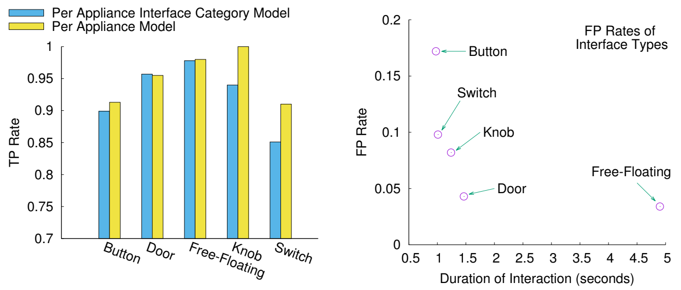
We test the efficiency of our model by conducting experiments with at least four different appliances for each appliance interface category. A list of all the appliances used in our experiments is shown in Table 1. For every appliance, we test with 3 users where each user interacts with an appliance at least 20 times. Overall, for each appliance, we have 60 user-appliance interaction events. Non-interaction events are simulated by collecting gyroscope and accelerometer data from smartband while the user is carrying out normal everyday tasks. We collect this data for three different users and for about three hours each over multiple days. For every unique user interaction event, we have a corresponding non-interaction event created from this data (chosen uniform randomly).

The main objective of MotionSync is to detect user-appliance interaction events correctly, in order to determine personal energy usage. We evaluate the system based on the True Positive (TP) and False Positive (FP) rates of detection of correct and incorrect user-appliance interactions. A true positive is defined as a detection of a user-appliance interaction event when the user actually interacts with the appliance. A false positive is defined as a detection of a user-appliance interaction when the user *does not* interact with any appliance. In such as case, the user was in the vicinity of an appliance tag to receive its broadcast, however, she did not interact with the appliance, and the system incorrectly detected that the user interacted with the appliance.

5.3 Numerical Results

In our system, we train a machine learning model for each *appliance interface category* using logistic regression. For each appliance interface category, we compute the DTW distances between the sensor data collected for user motion (smartband) and the motion tags (appliances). We calculate the afore-mentioned distances for the three axes of gyroscope and three axes accelerometer data for both, interaction and non-interaction events. Overall, our machine learning model is built using six features.

Overall System Performance: We start with evaluating the TP



(a) TP rate for models built based on each appliance interface category and each specific appliance. The category based models achieve comparable TP rate with much lower training overhead

(b) Variation of FP rate with duration of user-appliance interaction - shorter interaction duration result in random matches with non-user activity and higher FP rate

Figure 9: TP and FP rates for each appliance category

Interface Class	Appliance	TP Rate	FP Rate
Button	Light Button	0.873	0.190
	Microwave	0.842	0.088
	Coffee Maker	0.914	0.190
	Desktop	0.897	0.034
	Fan Button	0.968	0.190
Door	Oven 1	0.931	0.026
	Oven 2	1.00	0.051
	Washer	0.968	0.051
	Dryer	0.933	0.00
Free-Floating	Phone Charger	0.966	0.026
	Laptop Charger	0.969	0.00
	Remote 1	1.00	0.051
	Remote 2	0.952	0.051
	Hair Dryer	1.00	0.00
	Hair Dryer 2	0.900	0.00
Knob	Espresso Maker	1.00	0.050
	Stove Top 1	0.981	0.110
	Stove Top 2	0.900	0.017
	Toaster Oven	0.948	0.052
Switch	Toaster 1	1.00	0.067
	Toaster 2	1.00	0.00
	Rice Cooker	1.00	0.25
	Light Switch	0.828	0.276
	Water Heater	1.00	0.032

Table 2: TP and FP rate for individual appliances under each appliance interface category

rate for each of the models build for the five appliance interface categories - Button, Door, Free-floating, Knob and Switch. Here, 10-fold cross validation is used for training and testing.

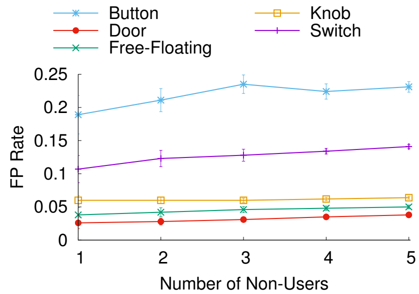
(1) TP rate: The results of TP rate are presented in Fig. 9a. For every appliance category, we observe 85% or higher accuracy in determining user-appliance interaction events. The relatively high TP rate across all interface appliance categories shows that motion matching between the tag and the smartband is a clear indicator of user-appliance interaction. Additionally, we train a separate model for each and every appliance (listed in Table 2) to evaluate how well our per-appliance category models perform compared to per-appliance models. The TP rates of these per-appliance models (averaged over all appliances in a category) are presented in Fig 9a. We observe that the "per appliance" models achieve only slightly better accuracy than the models built per appliance category, with an average improvement of 2.66%. Thus, the models built per appliance category not only reduce the training substantially, but also achieve comparable detection accuracy, and the appliance-specific models do not provide significant accuracy gain.

(2) FP rate: We also calculate the FP rate for each appliance interface category in order to study, how well MotionSync is in discarding non-interaction events. FP rates for each appliance category is shown in Fig. 9b. With the exception of one interface category (i.e. Button), all other classes of appliances have FP rate less than 10%. From Fig. 9b, we can also observe that shorter duration of interaction between user and the appliance interface leads to higher FP rates. Specifically, we see that appliance categories with longer duration of interaction between user and appliance (door, free-floating and knob) has lower FP rates than appliance classes with shorter interactions (button and switch). Combined and averaged together, instances with duration of interaction greater than 1 second has 96% and 5.3% TP and FP rate respectively, while the instances where the motion is less than one second (mostly buttons and switches) has an average of 87.5% and 13.5% TP and FP rate. This is because the IMU data of a short user-appliance interaction is more likely to match to the IMU data of a non-interaction (chosen randomly in our case) than a longer one. This is particularly observed in the button class which has the highest false positive rate 17%. Based on this, it is clear that as the duration of the user's interaction with the appliance increases, the TP rate also increases and the FP rate decreases.

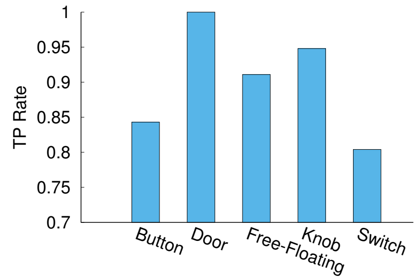
(3) Performance of Individual Appliances: We further evaluate how MotionSync performs for individual appliances within each specific interface category. In Table 2, we show the TP and FP rates of every appliance tested in our system, based on the model created above. It is observed that across all appliances within the same class, there is a similar pattern of TP and FP rates. This shows that the system successfully exploits the similarity of the interface type for an appliance category to eliminate the per-appliance training.

Multiple Users in the Vicinity: MotionSync is designed with a goal of accuracy in environments where multiple users are present (Section 3). It is important that, in these environments, our system is able to correctly apportion energy usage to the correct user and to ignore the non-users. Fig. 10a represents the performance of our system in this aspect, by showing how the FP rate varies with the change in the number of non-users present in the vicinity - all of whom receive the BLE broadcast from the motion tags. For example, if a user operates the stove in the presence of N other users in the kitchen, all users receive the BLE broadcast of the appliance motion tag, but only the user who operated the stove should detect an interaction event based on the motion matching. We are interested in understanding how the FP rate increases when the number of non-users (N) increase. The results are presented for all five appliance interface categories. We observe only a small increase in FP rate as the number of users increase. Despite the addition of more users, MotionSync is able to maintain considerably similar FP rate as compared to when only one additional user is present. Because the FP rate is directly impacted by the kind of actions non-users are doing, the presented FP rate is an average ($\pm$standard deviation) of the false positives achieved with multiple actions performed by 5 additional users. Fig. 10a shows that as the number of users in proximity increase, FP rate also increases but only by a small marginal amount (in average, an increase of only 2.08% after four additional users).

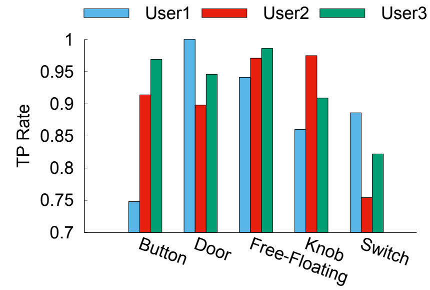
We emphasize that in real world settings, the FP rate will be lower than observed in our experiments. False positives are only introduced when a non-user receives BLE broadcast from the appliance motion tag. However, in real settings, the number of non-users within the vicinity of BLE broadcast range ($\approx$ 15 meters) are



(a) FP rate variation (average±standard deviation) with multiple non-users present in the vicinity who will receive BLE broadcast from motion tag



(b) TP rate for new and untrained appliances



(c) TP rate for new and untrained users

Figure 10: Performance of MotionSync from the point of view of multiple non-users, new and untrained appliances and users.

likely to be very small, reducing the observed FP rate. Additionally, it is possible to reduce the BLE transmission power on the tags to further reduce the FP rate. We leave this exploration to future work.

Untrained Users and Appliances: In order for the system to be easily adapted, it must be able to determine user-appliance interaction for untrained users and untrained appliances. We envision a scenario where a developer/manufacturer can create an user-appliance interaction model trained with their own set of users and appliances. However, in presence of an appliance with motion tags on them, when a new user installs this model on her smartphone, the system should be able to determine user-appliance interactions even for new users and new appliances. Thus, the appliances with the motion tag can be deployed on any appliance interface without users having to train on their own appliances or on themselves. Limiting the training of the model to generic users and appliance interface classes will greatly improve the adaptation of MotionSync.

Fig. 10b shows the TP rates of the five different interface category classes on new and untrained appliances. We test our model on two untrained appliances for each interface category with 50 interaction events each. The TP rates of untrained appliances highly resemble those in the trained model in Fig. 9a. We find that our system can detect a user-appliance interaction event even on untrained appliances as long as the interface category of the appliance matches one the five types listed in Table 1. This is primarily due to the fact that the system relies on generic characteristics of interface type while performing the motion matching which is common across trained or untrained appliances.

In addition to limit training to appliance interface class, the system also seeks to limit training to a limited number of users. We now show that training per user is not required for MotionSync to determine an user-appliance interaction event. We evaluate this by taking two of the three users originally used to create our training model and create a new training model. Similar to before, we train the new models by using logistic regression. We create these new models for every appliance interface category. We then test the third (excluded) user's interaction events on the newly trained model. We show the TP rates of the untrained user for each appliance interface category in Fig. 10c. The results show a similar results of TP rates to that shown in Fig. 9a. MotionSync's performance (TP rate) on untrained users does not degrade, allowing the system to limit it's training on generic users and not require training for every unique user in order to determine user-appliance interaction. Multiple users might interact with the appliance in different ways, but still our model can correctly identify user-appliance interaction. This is because, our model seeks to capture the difference in the motion of users' hand/wrist and the motion of the appliance interface. Even though users interact differently, the motion of wearable and the appliance tags remain similar. Thus, any variance introduced by the way a user interacts with an appliance does not significantly affect the performance of our system.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we presented MotionSync, a personal energy analytics system that exploits motion matching to detect user-appliance interaction and energy apportionment. Our evaluation showed that motion matching can be performed using generic models trained per category of appliances, significantly reducing the training overhead. This paper focuses primarily on a MotionSync's ability to accurately detect user-appliance interaction events. The evaluation of MotionSync's energy apportionment of real-time smart meter data is reserved for the scope of future work. The following aspects of MotionSync design can be improved in the future work. First, the system is designed to only work with plugload appliances. This excludes other energy consuming systems such as the HVAC system. It also excludes devices that users do not directly interact with such as network printers or WiFi routers, all of which consume different amount of energy based on user's actions. Second, our system needs a systematic framework to account for energy consumed by the shared appliances (for example, TV being watched by many users or light in an office corridor). Currently, MotionSync assigns the consumed energy to the user who interacted with the appliance instead of enabling a way to properly share it. Existing location based approaches can be integrated with our solution in a privacy-preserving manner, which is an important direction of future work.

7. REFERENCES

[1] Mohit Jain, Deepika Chhabra, and Amarjeet Singh. Short paper: Comparing energy feedback techniques for dormitory students in india. In *2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, BuildSys '15, pages 221–224, New York, NY, USA, 2015.

[2] Shailja Thakur, Manaswi Saha, Amarjeet Singh, and Yuvraj Agarwal. Wattshare: Detailed energy apportionment in shared living spaces within commercial buildings. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, BuildSys '14, pages 30–39, New York, NY, USA, 2014. ACM.

[3] Seungwoo Lee, Daye Ahn, Sukjun Lee, Rhan Ha, and Hojung Cha. Personalized energy auditor: Estimating personal electricity usage. In *Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on*, pages 44–49, March 2014.

[4] Juhi Ranjan and Kamin Whitehouse. Object hallmarks: Identifying object users using wearable wrist sensors. In

Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '15, pages 51–61, New York, NY, USA, 2015. ACM.

[5] Antonio Ridi, Christophe Gisler, and Jean Hennebert. A survey on intrusive load monitoring for appliance recognition. In *2014 22nd International Conference on Pattern Recognition (ICPR)*, pages 3702–3707. IEEE, 2014.

[6] Ahmed Zoha, Alexander Gluhak, Muhammad Ali Imran, and Sutharshan Rajasegarar. Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. *Sensors*, 12(12):16838–16866, 2012.

[7] N. Pathak, M. A. A. H. Khan, and N. Roy. Acoustic based appliance state identifications for fine-grained energy analytics. In *Pervasive Computing and Communications (PerCom), 2015 IEEE International Conference on*, pages 63–70, March 2015.

[8] Anthony Rowe, Mario Berges, and Raj Rajkumar. Contactless sensing of appliance state transitions through variations in electromagnetic fields. In *Proceedings of the 2Nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, BuildSys '10, pages 19–24, New York, NY, USA, 2010. ACM.

[9] K. Y. Chen, S. Gupta, E. C. Larson, and S. Patel. Dose: Detecting user-driven operating states of electronic devices from a single sensing point. In *Pervasive Computing and Communications (PerCom), 2015 IEEE International Conference on*, pages 46–54, March 2015.

[10] A. Ridi, C. Gisler, and J. Hennebert. User interaction event detection in the context of appliance monitoring. In *Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on*, pages 323–328, March 2015.

[11] Deokwoo Jung and Andreas Savvides. Theory and algorithm of estimating energy consumption breakdowns using on/off state sensing. *ACM Trans. Sen. Netw.*, 11(1):5:1–5:36, September 2014.

[12] N. Pathak, N. Roy, and A. Biswas. Iterative signal separation assisted energy disaggregation. In *Green Computing Conference and Sustainable Computing Conference (IGSC), 2015*, pages 1–8, Dec 2015.

[13] Jack Kelly and William Knottenbelt. Neural nilm: Deep neural networks applied to energy disaggregation. In *Proceedings of the 2Nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, BuildSys '15, pages 55–64, New York, NY, USA, 2015.

[14] Farrokh Jazizadeh. *Building Energy Monitoring Realization: Context-Aware Event Detection Algorithms for Non-Intrusive Electricity Disaggregation*, chapter 85, pages 839–848.

[15] Akshay S.N. Uttama Nambi, Antonio Reyes Lua, and Venkatesha R. Prasad. Loced: Location-aware energy disaggregation framework. In *Proceedings of the 2Nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, BuildSys '15, pages 45–54, New York, NY, USA, 2015. ACM.

[16] Nirmalya Roy, Nilavra Pathak, and Archan Misra. Fine-grained appliance usage and energy monitoring through mobile and power-line sensing. *Pervasive and Mobile Computing*, pages –, 2016.

[17] Samuel DeBruin, Branden Ghena, Ye-Sheng Kuo, and Prabal Dutta. Powerblade: A low-profile, true-power, plug-through energy meter. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, SenSys '15, pages 17–29, New York, NY, USA, 2015. ACM.

[18] Ekhiotz Jon Vergara, Simin Nadjm-Tehrani, and Mikael Asplund. Sharing the cost of lunch: Energy apportionment policies. In *Proceedings of the 11th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, Q2SWinet '15, pages 91–97, New York, NY, USA, 2015. ACM.

[19] Juhi Ranjan, Erin Griffiths, and Kamin Whitehouse. Discerning electrical and water usage by individuals in homes. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, BuildSys '14, pages 20–29, New York, NY, USA, 2014. ACM.

[20] Simon Hay and Andrew Rice. The case for apportionment. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, BuildSys '09, pages 13–18, New York, NY, USA, 2009. ACM.

[21] Manaswi Saha, Shailja Thakur, Amarjeet Singh, and Yuvraj Agarwal. Energylens: Combining smartphones with electricity meter for accurate activity detection and user annotation. In *5th International Conference on Future Energy Systems*, e-Energy '14, pages 289–300, New York, NY, USA, 2014. ACM.

[22] Edward J. Wang, Tien-Jui Lee, Alex Mariakakis, Mayank Goel, Sidhant Gupta, and Shwetak N. Patel. Magnifisense: Inferring device interaction using wrist-worn passive magneto-inductive sensors. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '15, pages 15–26, New York, NY, USA, 2015. ACM.

[23] Chao Xu, Parth H. Pathak, and Prasant Mohapatra. Finger-writing with smartwatch: A case for finger and hand gesture recognition using smartwatch. In *16th International Workshop on Mobile Computing Systems and Applications*, ACM HotMobile '15, pages 9–14, New York, NY, USA.

[24] Sheng Shen, He Wang, and Romit Roy Choudhury. I am a smartwatch and i can track my user's arm. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, ACM MobiSys '16.

[25] He Wang, Ted Tsung-Te Lai, and Romit Roy Choudhury. Mole: Motion leaks through smartwatch sensors. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, ACM MobiCom '15, pages 155–166, New York, NY, USA.

[26] Li Zhang, Parth H. Pathak, Muchen Wu, Yixin Zhao, and Prasant Mohapatra. Accelword: Energy efficient hotword detection through accelerometer. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '15, pages 301–315, New York, NY, USA, 2015. ACM.

[27] Keng-hao Chang, Jeffrey Hightower, and Branislav Kveton. Inferring identity using accelerometers in television remote controls. In *Pervasive Computing*, pages 151–167. 2009.

[28] Household Energy Usage - U.S. Energy Information Administration. http://www.eia.gov/consumption/residential/reports/2009/state_briefs/.

[29] Rainforest Eagle Energy Gateway. https://rainforestautomation.com/rfa-z109-eagle/.

[30] TinyDuino Platform. https://tinycircuits.com/pages/tinyduino-overview.

[31] Android AndroSensor. https://play.google.com/store/apps/details?id=com.fivasim.androsensor&hl=en.

[32] ComProbe BPA 600. http://www.fte.com/products/BPA600.aspx.