Frank Englert, Irina Diaconita, Andreas Reinhardt, Alaa Alhamoud, Richard Meister, Lucas Backert, Ralf Steinmetz

Multimedia Communications Lab
Technische Universität Darmstadt
Darmstadt, Germany
{FirstName.LastName}@kom.tu-darmstadt.de

School of Computer Science and Engineering
University of New South Wales
Sydney, Australia
andreasr@cse.unsw.edu.au

Abstract

As a consequence of rising energy prices, manifold solutions to create user awareness for the unnecessary operation of electric appliances have emerged, e.g., real-time consumption displays or timer-based switchable wall outlets. A common attribute of these solutions is the need to buy and install additional hardware, although their acquisition and operation costs often diminish the attainable savings. Furthermore these solutions only permit to retrieve accumulated figures of the energy consumption. Especially in households or office spaces with multiple persons, however, attributing electricity consumption to individuals provides enormous potential to determine possible savings.

We therefore propose a system that allows to identify the energy demand incurred by a user's action based on audio recordings using smartphones. More precisely, we capture the user's ambient sounds and applying suitable filtering steps in order to determine the user's current activity with the help of a machine learning component. Our results indicate that our system is capable of detecting 16 typical household activities at an accuracy of 92%. By annotating the detectable household activities with information about typical energy consumptions, extracted from 950 real-world power consumption traces, a good estimate of the energy intensity of the users' lifestyles can be made. Our novel personalized energy monitoring system shows people their personal energy consumption, while maintaining their user comfort and relinquishing the need for additional hardware.

Categories and Subject Descriptors

C.m [**Computer Systems Organization**]: Miscellaneous—*Mobile Sensing Systems*

5th ACM Workshop On Embedded Systems For Energy-Efficient Buildings '13 Rome, Italy
ACM

Keywords

Mobile Sensing, Energy Consumption Monitoring, Activity Detection, Audio Analysis, Energy Consumption Estimation

1 Introduction

Globally rising energy prices have led to an increased user awareness for a conscious use of electric energy. Motivated by the widely used slogan "The cheapest energy is the one we do not consume," people take a great interest in their energy footprint and the potential for savings. Indeed, we agree on this opinion that the awareness of a user's electricity consumption is a key factor in supporting a sustainable lifestyle and avoiding the wastage of electric energy through the unnecessary operation of devices.

In this paper, we thus present an approach to estimate a user's energy footprint which is based on the collection of sound samples from the environment and extracting the underlying operating devices from these recordings. We have deliberately chosen not to install special energy sensors in buildings due to the additionally incurred cost that may quickly exceed the achievable energy savings. Instead, we propose to rely on sound samples that have been recorded using a mobile device, e.g. a smartphone. Having become lifestyle items that are carried close to their owners at all times, smartphones allow for unprecedented acoustic insights into the user's current environment and thus permit to attribute device operation to the user.

Our research is motivated by the fact that the largest optimization potential exists for appliances that are actively controlled by the user. This class of devices has a high chance of operating unnecessarily due to the user's failure to switch them off when not needed any longer. At the same time, however, they are usually specifically activated by the user when their operation is deemed necessary; in other words, the user is generally in close physical proximity to such appliances when turning them on. We thus investigate the efficacy of our approach to use smartphones for recording and identifying the sounds recorded from the user's surroundings.

We make the following contributions:

- We present our system design, which performs the tasks of collecting sound samples from the user's environment and extracting characteristic features from these data.

- We conduct a comprehensive evaluation of the situation recognition efficacy when smartphone audio recordings are used. For this purpose, we have sampled the operational sounds of 16 different devices and situations. Our activity classification algorithms could determine these situations based on their sound with an accuracy of 92%.
- We map the detected appliance operations to real situations and enrich the knowledge of their operation by the energy consumption of the situation in which they are typically used. This energy augmentation is based on situations extrapolated from a real-world data set.

This paper is structured as follows. First a broad overview of the related work is given in section 2. In the following section 3 the concept of our energy estimation system SensiMate is presented and the design decisions are discussed. Then in section 4 we give an exhaustive explanation of its implementation. In section 5 we extensively evaluate the key parts of our system, namely the activity classification and the energy estimator with real-world data. Last but not least we conclude this paper and summarize our findings.

2 Related Work

Building an indirect energy consumption estimator involves two major steps, namely the activity classification and the energy modeling. Therefore we begin with an overview of the state of the art in activity detection. Then we discuss selected works which build models to estimate the energy consumption.

In the past few years many models for user activity detection were proposed, which vary with respect to both the types and combinations of sensors used, as well as approaches and use cases. For instance, accelerometers have been successful to detect the users' physical activity [24, 10, 13, 9, 11, 27], while GPS and WiFi/Bluetooth traces help track users [19] and build and predict user mobility patterns [25, 8, 26], detect their most important places and estimate when and for how long they would come to a specific place [22]. More complex activities, like telling the type of location the user's in, say a bookstore, restaurant or disco, would need to combine the WiFi traces, sound, accelerometer, camera and light sensor data, as SurroundSense [1] shows.

Still, to the best of the authors' knowledge, there are no approaches to specifically detect a user's energy consuming activities and to build a consumption profile based on that. The problem of recognizing the time period as well as the type when somebody is using an electrical device, has to focus more on detecting the activity of a device than of the user himself. We decided to use the acoustic emissions for this purpose and discuss the reasons for this in section 3.3. Therefore we will present related approaches from the field of acoustic event detection in the next few paragraphs.

There are approaches to use microphone recorded sounds to detect the health state of the user, for instance a cough detector [12] or a heartbeat counter [17]. Schweizer et al. [23] create a participatory noise mapping system and Chen et al. [4] detect and analyze activities taking place in the bathroom. Mesaros et al. [15] detect a wide array of activities related to the user's environment (namely, 60 activities), while AmbientSense [21] takes this one step further by executing the recognition in real time, for a varied array of sounds that could characterize the user's environment, from a dishwasher running to birds chirping and restaurant or forest noises. Peltonen et al. [18] detect various contexts of the user, which are grouped in six main categories: outdoors, vehicles, public/social places, offices/quiet places, home and reverberant places.

While all the above mentioned approaches create applications for specific sets of actions and sounds, there are frameworks like Auditeur [16] which attempt to label and classify all environment sounds and employ a user participatory approach for that, allowing users to add and label their own data.

When knowing the activity and its duration, we can estimate the energy consumption for the given activity. Chen et al. [3] propose a framework for energy data collection, analysis and prediction of future consumption in smart homes. It analyzes the relationship between behavior patterns and energy consumption and aims to support users in saving energy. The autors model the energy consumption of a whole household for different activities based on non-linear curve fitting.

Many researchers from the field of WSN and Mobile Sensing built similar models to estimate the energy consumption of certain hardware parts or computer programs based on different input parameters: Zhang et al. [28] have developed a system to estimate the battery consumption of certain computer programs based on the state of display, the CPU speed and the network state. Dunkels et al. [7] developed a program to aproximate the energy consumption of WSN motes based on the current running program, and the communication strategies. The goal is to increase the lifetime and thus the maintenance time of sensor networks by developing energy aware routing algorithms. We use the experiences from the aforementioned works to create different models for estimating the energy consumption of different activities.

3 Concept

As mentioned in the introduction, state of the art energy monitoring systems often require the installation of special hardware. Such a solution has many disadvantages: firstly the user has to buy specific energy meters. Secondly, the user has to manually install them in his household or at work. Thirdly, the measurements conducted by these energy meters are non personalized which hinders the analysis of the metering data in multi-person environments.

To get over these limitations, we will develop an energy consumption estimation system named SensiMate, which relies on detection of emissions caused by electricity consuming activities or appliances. These measurements could be carried out by customary smartphones which measure the acoustic noise emissions, derive the activity which caused these emissions and then map an energy consumption to these activities.

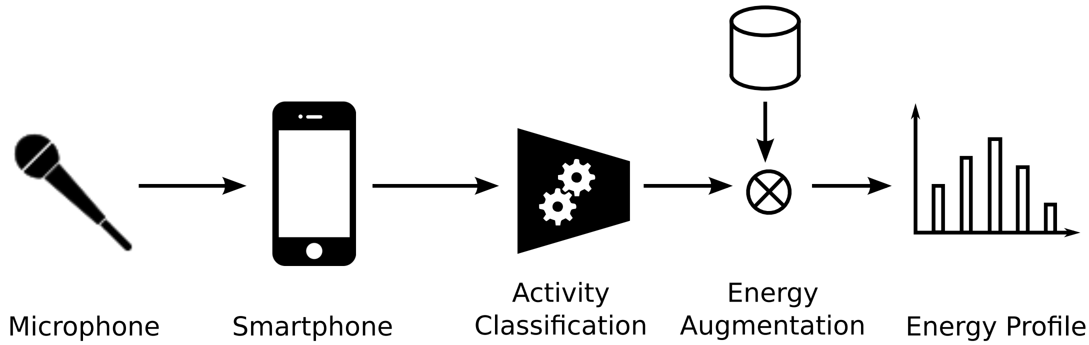

Figure 1. Overview over SensiMate

In this section, we'll first discuss the concept of indirect measurements, describe our concept to model the energy consumption and explain the design space of such a solution in detail.

3.1 Indirect Measurement

Instead of directly measuring the desired variable, indirect measurements relies on metering effects caused by this variable together with a known correlation between the metered unit and the desired variable. These indirect measurements are often easier to achieve while being less accurate in the general case. This is also true in our scenario: Directly measuring the energy consumption on device granularity would require either one distributed electricity meter per appliance, energy meters with load disaggregating capabilities or electrical appliances which are aware of their current energy consumption.

In contrast to that, an indirect energy measurement solution only requires sensors which can detect the emissions caused by the energy consumption. Some of these emissions, like audio signals, can even be recognized by customary smartphones. Therefore no special equipment is required to capture the energy consumption of certain activities or appliances. This simplifies the setup overhead dramatically - instead of installing energy meters at home or in the office, the user only has to install ours application on his smartphone.

When analyzing the energy reports, it would be helpful to know which user and which activity caused the energy consumption. Without this information, deriving energy saving recommendations for specific persons is a hard task. Creating a relation between persons and device level energy consumption is difficult to achieve with classical energy meters, because the energy consumption carries not much information about the current user of an electrical appliance.

As an advantage of our solution, the indirectly estimated consumption profiles are automatically personalized because such a solution can only detect activities which happen next to the user.

3.2 Activity based Energy Modeling

Knowing the activity carried out by the user, the energy consumption for this activity must be estimated to conduct energy consumption reports. There are many possibilities to build such an estimator ranging from Constant Values over Linear Regression Models up to Neuronal Network Estimators or Hidden Markov Models [2]. Selecting the right model directly influences the accuracy of such estimations. Thus carefully determining the parameters with respect to the input of the model is crucial. In our specific scenario, there are mainly two parameter variations:

1. Activities with a determined duration, started by the user.
2. Activities with a variable duration, carried out by the user.

In the first case, the user only starts the activity and therefore the energy consumption depends on the tool to carry out this certain activity. Using a dishwasher or a toaster are examples for the first type of activities as these appliances are started by the user and continue running unattended. In the second case, the energy consumption directly depends on the tools used to carry out the specific activity as well as on the time frame, the user performs this activity. An example of such an activity is working or watching television.

To model activities started by the user, our solution estimates these activities with a constant energy consumption without considering the activity duration. Hence such a model even fits well when the detected activity continues unattended.

For activities directly carried out by the user, we assume that it is possible to determine the duration of this activity. This assumption may hold most of the time because the user must be present and therefore the acoustic emissions of the current activity are present most of the time. Having the activity and its duration, we use a linear regression model to estimate the energy consumption based on the duration.

3.3 System overview

As visible in Figure 1 the SensiMate system mainly consists of five components arranged in a processing pipeline. The first stage of this pipeline is obviously the sensing device which captures the required sensor data and outputs a raw data stream. The second stage consumes this raw data stream, windows the data, applies an activity classification to each window and outputs the resulting activity stream. Building upon that, the third stage augments the activity stream with a predefined energy consumption profile and outputs an integrated activity and energy consumption stream. The last stage then aggregates this stream to sum up the energy for each activity and stores these reports for visualization purposes.

Basically such a solution could be based on different kinds of sensor data like audio, video, the gyroscope, GPS or a mixture of them, which are available in most state of the art smartphones. However, practical considerations narrow down the choice to the usage of microphones because these sensors work under nearly all environmental conditions without further user interaction. Only activities happening in noisy environments are not detectable by these sensors. Additionally, our system could rely on gyroscope measurements, but energy consuming activities are hard to differentiate based on the movements and are few activities which do not involve acoustic emissions but movements. For instance a user bending down could either use the washing machine, the dishwasher, the dryer, a refrigerator or simply be picking up a bag.

Except of the data recording, each stage of SensiMate can either run locally on the phone or on a remote server. Each of both deployment options has certain advantages and disadvantages. A detailed of these two options is given by Rossi [21]. We decided to deploy all stages except of the recording of SensiMate remotely in the cloud because this setup simplifies the update process of our machine learning model. Nevertheless our solution could be adopted to run on the smartphone without major changes to the underlying design principles.

4 SensiMate Implementation

As described in section 3, the components of SensiMate are arranged as a data processing pipeline. The input of this pipeline is an audio sample stream from the smartphone's microphone and the output is an energy augmented activity event stream. The pipeline concept has many advantages, namely conceptual simplicity, parallelizability, flow-decoupling of the pipeline stages and the capability to process infinitely long data streams.

These advantages are simply based on the fact that each pipeline stage only depends on the interfaces to the predecessor and to the successor. There are no direct dependencies between single pipeline stages. This allows an injection of middleware components which provide certain nonfunctional properties like parallelization, queuing or deploying one or more stages at remote hosts. But it also allows to replace certain stages with other implementations satisfying the same interface.

Having explained the overall structure of our SensiMate, we will now explain its pipeline stages in detail.

4.1 Stage 1: Recording

This stage records the sensor data obtained from the smartphone's internal microphone, creating an infinitely long stream of sensor data. Thus, we use an Android background service, which collects audio sample stream. We use a plugin kind of architecture for the sensors we collect the data from, so the app could be easily extended in the future if other sensors incorporated in the phone are deemed to bring useful information related to the energy consumption.

The recordings can be either stored on the smartphone's internal memory or fed into the processing pipeline by transferring them to a central server. Having both possibilities is particularly important for the user, who might have no internet connectivity at some point, or might be on a limited mobile plan, so the app has the option of saving the time-stamped samples and transferring them only when the user is connected to a WiFi network. The app also allows the user to label specific readings/situations, which is mainly intended for gathering the training instances for the classifier and less for the final user who should be able to leave the application run in the background, without having to intervene or give any input.

Also, depending on the manufacturer, the phone might not have a unique identification number and the MAC addresses cannot be read if the WiFi is disabled, so we had to come up with a different solution to identify the user whose consumption profile we create. Thus, we do not identify the device, but the app installation, by generating a unique number for each installation instance.

4.2 Stage 2: Activity Classification

As described in section 3.3, the main purpose of this stage is the deduction of the current activity based on the incoming audio stream. To achieve this task, the stage is partitioned in five sub-stages. In the subsequent section, we will describe them in detail.

4.2.1 Windowing

The first sub-stage of the activity classification is responsible for windowing the infinitely long audio stream from the recording stage in smaller pieces of data which are easier to process. To window the data, we use a simple, rectangular windowing function with a fixed length window size and a configurable overlap between the last window and the current window. As preliminary tests have shown, the side lobes generated by this window function do not affect the classification accuracy. Currently, our windowing function creates windows with a size of 4,096 samples. Together with the sample rate of 44.1 kHz, this window size corresponds to an audio length of 10.8 ms per window.

Nevertheless, this windowing function may have a further improvement potential: currently, each window starts at a randomized point in time. Each activity also starts at a randomized point in time. The chances are good, that the starting point of the nearest time window and the starting point of the activity have a high difference which may cause a window which is nearly filled with background noise. This fact may be a problem, because the turn on transient of some activities are very characteristical. To solve this problem, we configured the windowing in a way, that each window overlaps the last window by a factor of 50%. More sophisticated solutions could try to detect the starting point of events and synchronize this with the starting point of a new window.

4.2.2 Silence Removal

The main goal of the silence removal stage is to filter out windows containing only evironmental noise and no information about the current activity. Removing them is important to achieve a high classification accuracy in the next sub-stage. If these segments were not removed before the classification, the classifier will assign them an arbitrary class which reduces the accuracy of the overall system. Therefore having a robust silence removal algorithm is crucial to the overall system performance.

Detecting silent parts in audio signals is important in many fields. For example the well-known speex audio codec uses such an algorithm for voice activity detection [5]. Typically these algorithms consist of three consecutive steps: first, a noise reduction schema is applied, then some features like the spectral shape or the signal energy are calculated and in the last step a classification rule is applied to decide whether the signal is noise.

According to this concept, our silence removal sub-stage works straight forward. First we calculate the signal energy of the current window. If this energy is below a certain threshold, the window is filtered out. If the signal energy is above this threshold, the window is forwarded to the next sub-stage.

4.2.3 *Feature extraction*

The extraction of significant features is the one of the most important tasks when solving separation problems. Hence, a modular feature extraction sub-stage was developed. When designing the feature extractor, we had the extensibility in mind. Thus, new feature extractor functions can be added on demand. So far we implemented the following features:

- Zero Crossing Rate (ZCR)
- Mel Frequency Cepstral Coefficient (MFCC)
- Delta MFCC (DMFCC)
- Band Energy (BE)
- Power Spectrum (PS)

The Zero Crossing Rate was used in some older speech recognition algorithms. It is defined as the number of sign changes in a defined period. Thus it is an easy to calculate feature. Newer speech recognition algorithms and music information retrieval systems rely on MFCC features. These features have been well proven for speech recognition [6, 14], and were adopted by the authors of [21, 15, 16] for audio event classification due to their accurate recognition results. Therefore we use MFCC features with 13 logarithmically distributed, triangular shaped filter banks to calculate the coefficients. As the MFCC features are only calculated in one particular window, they do not contain information about the long term time domain. To also obtain features for the variation in time, we also calculate Delta MFCC features which are the first order derivative of the MFCC-features with respect to time. Last but not least we also calculate the band energy of 13 logarithmically distributed frequency bands as a feature for the classification. The filter function for each frequency band has a triangular shape.

To sum up, this sub-stage takes a window with audio samples as input, calculates the aforementioned features and then outputs these features to the next stage.

4.2.4 *Classification*

Having extracted the features from the audio signals, the corresponding activities can be deduced. Responsible for this task is the classification sub-stage. Our implementation is modular in a way, that allows plugging in many classifier implementations. Currently our solution supports the following classifiers: Decision Tree (DT), Random Forest (RF), Gaussian Naive Bayes (GNB), K Nearest Neighbors (KNN), Support Vector Machine Classifier (SVC), and Gaussian Mixture Model (GMM). We expect the best classification results with state of the art ensemble classifier Random Forest. A detailed comparison of different classification methods is available in section 5.

4.2.5 *Window reduction*

There are multiple windows per second - this means that there are multiple classification results per second. Having so many activities causes a high computation overhead in the next stages and make the whole process prone to errors caused by single windows which are incorrectly classified.

To reduce the number of windows we use a flattening algorithm which pre-filters the data and aggregates the generated activity stream over time to reduce the amount of data to process in further stages.

4.3 Stage 3: Energy augmentation

Having an activity stream from the previous SensiMate pipeline stage, now the activity must be augmented with its energy consumption. To do so, this pipeline stage accumulates sequential activities of the same kind to measure the activity's duration. Once the activity is finished, this stage uses the estimator to approximate its energy consumption. As described in section 3.2, depending on the activity, the estimator either uses a Constant (C) or a linear regression model for approximating the energy consumption. To build the underlying models, we used energy profiles from the publicly available Tracebase [20]. For the Constant model, we calculate the average energy consumption E_{est} for the energy consumption e_n of each carried out activity a:

$$E_{est}(a) = \sum_{n=0}^{N} \frac{e_n(a)}{N} \tag{1}$$

The output of the estimator for these models is $E_{est}(a)$. Obviously the output does not depend on the duration of the activity.

For the Linear model, we calculated the average power consumption P_{est} for the power p_n and the duration t_n over all carried out activities a.

$$P_{est}(a) = \frac{1}{N} \sum_{n=0}^{N} \frac{p_n(a)}{t_n} \tag{2}$$

In this case, the output of the estimator is the linear function

$$E_{est}(a,t) = P_{est}(a) * t \tag{3}$$

depending on the activity a and its duration t.

With these approximation models, the estimator can guess the energy consumption of the given activity. Finally this pipeline stage outputs this activity together with its energy wastage E_{est}.

4.4 Stage 4: Aggregation

This last stage of the pipeline reads the energy augmented activity stream from the input and aggregates the energy wastage for each activity. Then it outputs the activity, its start

Table 1. Activities recorded to evaluate the activity classification stage.

Code	Activity	# Windows	Precision
bre	Cutting bread	964	97%
cof	Making coffee	4213	93%
com	Working at laptop	205	81%
dis	Dishwasher	12,224	99%
doo	Opening door	656	68%
ket	Using kettle	14,617	99%
lcd	Watching television	6149	98%
men	Lunch in the cafeteria	654	96%
mic	Using microwave	2217	99%
tap	Using water tap	3033	92%
toa	Toasting bread	197	86%
too	Brushing Teeth	1374	98%
unl	Unlock door	726	78%
uac	Air Conditioner	1046	98%
was	Washing machine	12,416	99%
wc	Flushing toilet	3240	92%
	Total	64,960	92%

time and duration as well as the estimated energy consumption. So this pipeline stage generates a personalized energy consumption profile. This information is quite valuable for further processing. Therefore various applications for giving user feedback or deriving energy saving recommendations could use these reports to give user feedback.

5 Evaluation

The accuracy of SensiMate depends on precise activity detection as well as on exact energy estimations for the detected activities. Thus we first evaluate the classification precision of our solution by categorizing a set of audio samples. In the next step, we evaluate the accuracy of our energy model by comparing the estimations of our model with the real energy consumption of different activities.

5.1 Evaluation Setup

We used a Samsung Galaxy S2 as well as a Google Nexus S for recording various audio samples of 16 different appliances and activities. All audio samples were recorded sequentially in a quiet office environment with the microphone exposed to the open air. As this setup does not reflect the usage patterns of many smartphone users, a follow up study is required to prove the usability of SensiMate under real world conditions. The audio sample rate is 44.1 kHz which gives us, together with the Nyquist criterion a usable frequency range from zero to 22.05 kHz. The length of these recordings ranges from 10 to 240 seconds. We've built up a training set consisting of 108 recordings from these samples to train our activity classifier. A listing of these samples is shown in Table 1. The column Windows of this table shows the number of non-silent windows extracted from the sound files for each activity whereas the column Precision shows the accuracy of classifying the activity as described in section 5.2. The activities used in the evaluation were selected to determine the capabilities of our activity classification with a broad set of different sounds.

To evaluate the energy augmentation stage, we used the energy consumption of various devices from the publicly available Tracebase [20]. This data set consists of approximately 1.800 power traces for different electrical appliances. To record these traces, Plugwise electricity meters with a sample rate of 1 Hz were used. As the traces are recorded over 24 hours, we used a simple segmentation algorithm to extract only segments when the appliance was consuming energy from the power traces. This segmentation works as follows: a segment starts when the consumption is above 0 and ends when the consumption is 0 for more than 10 seconds. We extracted 950 segments for 10 different activities from the tracebase to evaluate our energy model. The detailed evaluation process is described in section 5.3.

Table 2. Precision/Recall results for each feature and classifier combination. All values are precentages.

Results	DT	RF	GNB	KNN	SVC	GMM
ZCR	46/43	46/37	46/39	43/40	46/44	47/37
PS	82/83	88/88	41/67	89/89	62/67	42/68
BE	85/84	88/88	31/61	78/77	58/61	32/58
MFCC	88/88	92/92	82/84	91/91	87/86	82/84
DMFCC	92/93	92/93	81/82	90/90	86/85	80/82

5.2 Activity classification

The SensiMate proposed in this work highly depends correctly classifying the current activity. Thus we will carefully evaluate the precision of our audio based activity classification. To come to a conclusion which classification algorithms together with which features to use, we compare the classification performance for all of these combinations and select the best combination for the further evaluation of the whole SensiMate pipeline.

As shown in Table 2, we evaluated the classification algorithms described in section 4.2.4 together with the features described in section 4.2.3. DMFCC features and MFCC features achieve comparable results where, depending on the classifier the DMFCCs are slightly more accurate. As the DMFCC also contains information about historic signal changes, this behavior is expectable. The best classification results were achieved with a Random Forest Classifier but the KNN-Classifier achieved comparable results for all feature classes except the Band Energy. According to these results, we configured the SensiMate to extract DMFCC-features and to use Random Forest for classification.

The results of a classification run with DMFCC features and a Random Forest Classifier are shown in Figure 2. It can be noticed that these results are highly accurate for most of the classes. Even very similar sounds like flowing tap water or flushing the toilet can be separated. An exception to this are the classes open-door and unlock-door. This happens due to the fact, that both audio sequences have very similar sequences. Therefore distinguishing between these classes fails more often than not. While showing the limits of the audio-based-activity classification, however, this is no major problem for the SensiMate because these classes do not cause an electrical energy consumption.

The results of the audio based activity classification look really promising. In a future work, we plan to benchmark

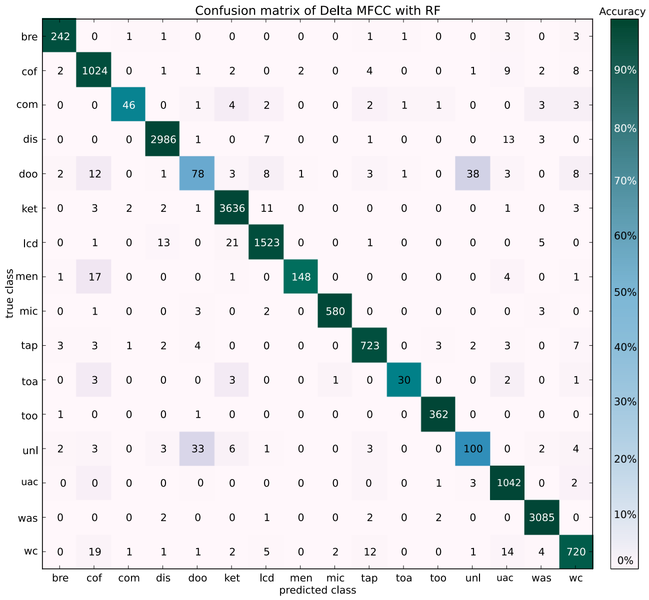

Figure 2. Confusion matrix showing the 16 classes we used to check the precision of our classification pipeline. To generate this plot, MFCC features together with a random forest classifier was used.

our SensiMate with the algorithms of other researchers [21, 15, 18] in the field.

5.3 Energy Augmentation Accuracy

After determining the current activity of the user, the SensiMate uses an estimator to approximate the energy consumption of an activity. This raises the question of how accurate this estimation is. To find an answer, we evaluated it with ten different activities. Therefore, we built different Linear (L) and Constant (C) energy models for the activities given in Table 3. This model building procedure is described in section 3.2. In the next step, we extracted 950 triples from the Tracebase. These triples consist of the energy consumption, duration and activity and act as ground truth to validate the energy consumption models. For each triple, the activity as well as the duration was given to the estimator to get an estimated energy usage. Finally we compared the approximation with the real energy consumption to determine the accuracy of the energy estimator. The mean relative error for each activity is also shown in Table 3.

For some of these activities, the estimator could accurately predict the energy consumption with a relative error below 6%. For other activities, like using the microwave oven, the error is high. These high errors have multiple reasons like the constant model assuming a constant energy consumption for a certain activity or the linear model assuming a constant power consumption over the whole run time. This model may fit well for a specific appliance which always operate within the same operation mode but it may be inaccurate for the general case. In our scenario, the 149 microwave activities were recorded with 8 different microwave ovens used in multiple different operating modes. Due to this high variance, making accurate energy estimations is hard to

Table 3. Relative error of the energy model compared to the real consumption is given here.

Model	Activity	Count	rel Error.
C	Dishwasher	45	5.72%
L	Watching TV (CRT)	22	5.49%
L	Working at Laptop	14	14.62%
C	Toasting Bread	46	22.49%
L	Watching TV (LCD)	117	35.82%
C	Making Tea	420	40.73%
L	Cutting Bread	39	58.79%
C	Making Coffee	93	58.93%
C	Washing Machine	5	82.18%
L	Using the Microwave	149	192.47%
Total		950	51.72%

achieve without having further information about the operating mode or the specific appliance series. Nevertheless a more sophisticated model may reduce the estimation error significantly.

5.4 Energy Consumption

As SensiMate runs partially on a ressource constrained devices with a bounded battery capacity, it is important to know the influence of SensiMate on the battery lifetime of a smartphone. To find an answer to this question, we measured the battery drainage over time with SensiMate running on a Samsung Galaxy S2 phone.

6 Future Work

In the previous sections we presented SensiMate, our personalized energy consumption estimator. During the development as well as the paper writing process, this work showed us a great potential for extensions and future research. Therefore we will present the most promising directions in this section.

Most importantly, we have to increase the accuracy, as well as the robustness of the energy estimation model. To achieve these two goals, we will carefully investigate the estimations with a high error to find improvement potentials. Having such a refined model, we will evaluate the estimation accuracy of the overall system in a real world deployment, together with a distributed smart meter installation as ground truth.

Especially our activity classification algorithms have shown a great potential for recognizing various activities. Together with more and better estimators, we could not only approximate the energy consumption of a user but also his water consumption as well as his carbon dioxide footprint. Using participatory sensing for collecting many activity and energy consumption tuples for multiple users, we may even be able to extend this system for providing crowd sourced energy saving recommendations.

7 Conclusion

In this paper we have presented the SensiMate system which creates user awareness their current energy footprint. Instead of relying on power sensors, whose installation may be costly and require technical support by an electrician, our approach is based on the use of audio data which is collected by the users' smartphones. SensiMate extracts several feature sets from the samples, which are being subsequently being used to train a machine learning component. Our evaluation of different types of classifiers in combination with different feature sets has shown that high precision and recall values of up to 92% and 93%, respectively, can be attained. The resulting possibility to detect appliance operation solely based on audio data enables novel ways to inform users about the energy footprint without the need for external hardware.

In the future, we will extend SensiMate by similar functionality for water and natural gas consumption monitoring, as we expect many of these consumers (e.g., shower, gas stove) to also emit characteristic audio signals when in use. The final step on the way to a holistic energy awareness system is the development of a smartphone app that suggests possibilities for energy saving to the user based on the detected consumption patterns.

8 References

[1] M. Azizyan, I. Constandache, and R. Roy Choudhury. SurroundSense: Mobile Phone Localization via Ambience Fingerprinting. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, 2009.

[2] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[3] C. Chen, D. J. Cook, and A. S. Crandall. The User Side of Sustainability: Modeling Behavior and Energy Usage in the Home. *Pervasive and Mobile Computing*, 2013.

[4] J. Chen, A. H. Kam, J. Zhang, N. Liu, and L. Shue. Bathroom Activity Monitoring Based on Sound. In *Pervasive Computing*. 2005.

[5] I. Cohen. Noise Spectrum Estimation in Adverse Environments: Improved Minima Controlled Recursive Averaging. *IEEE Transactions on Speech and Audio Processing*, 2003.

[6] M. Cooke, P. Green, L. Josifovski, and A. Vizinho. Robust Automatic Speech Recognition with Missing and Unreliable Acoustic Data. *Speech Communication*, 2001.

[7] A. Dunkels, F. Osterlind, N. Tsiftes, and Z. He. Software-based Online Energy Estimation for Sensor Nodes. In *Proceedings of the 4th Workshop on Embedded Networked Sensors*, 2007.

[8] T. Higuchi, S. Fujii, H. Yamaguchi, and T. Higashinoz. An Efficient Localization Algorithm Focusing on Stop-and-go Behavior of Mobile Nodes. In *International Conference on Pervasive Computing and Communications*, 2011.

[9] S. Hoseinitabatabaei, A. Gluhak, and R. Tafazolli. uDirect: A Novel Approach for Pervasive Observation of User Direction with Mobile Phones. In *International Conference on Pervasive Computing and Communications*, 2011.

[10] N. C. Krishnan and S. Panchanathan. Analysis of Low Resolution Accelerometer Data for Continuous Human Activity Recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008.

[11] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Activity Recognition Using Cell Phone Accelerometers. *ACM SIGKDD Explorations Newsletter*, 2011.

[12] E. C. Larson, T. Lee, S. Liu, M. Rosenfeld, and S. N. Patel. Accurate and Privacy Preserving Cough Sensing Using a Low-cost Microphone. In *Proceedings of the 13th International Conference on Ubiquitous Computing*, 2011.

[13] J. Liu, Z. Wang, L. Zhong, J. Wickramasuriya, and V. Vasudevan. uWave: Accelerometer-based Personalized Gesture Recognition and its Applications. In *International Conference on Pervasive Computing and Communications*, 2009.

[14] H. Lu, A. J. Bernheim Brush, B. Priyantha, A. K. Karlson, and J. Liu. SpeakerSense: Energy Efficient Unobtrusive Speaker Identification on Mobile Phones. In *Pervasive Computing*. 2011.

[15] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen. Acoustic Event Detection in Real Life Recordings. In *18th European Signal Processing Conference*, 2010.

[16] S. Nirjon, R. F. Dickerson, P. Asare, Q. Li, D. Hong, J. A. Stankovic, P. Hu, G. Shen, and X. Jiang. Auditeur: A Mobile-Cloud Service Platform for Acoustic Event Detection on Smartphones. In *The 11th International Conference on Mobile Systems, Applications, and Services*, 2013.

[17] S. Nirjon, R. F. Dickerson, Q. Li, P. Asare, J. A. Stankovic, D. Hong, B. Zhang, X. Jiang, G. Shen, and F. Zhao. MusicalHeart: a *Hearty* Way of Listening to Music. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, 2012.

[18] A. Peltonen, J. Tuomi, A. Klapuri, S. Fragerlund, T. Sorsa, G. Lorho, and J. Huopaniemi. Audio-Based Context Recognition. *Transactions on Audio, Speech and Language Processing*, 2006.

[19] T. Pongthawornkamol, S. Ahmed, K. Nahrstedt, and A. Uchiyama. Zero-knowledge Real-time Indoor Tracking via Outdoor Wireless Directional Antennas. In *International Conference on Pervasive Computing and Communications*, 2010.

[20] A. Reinhardt, P. Baumann, D. Burgstahler, M. Hollick, H. Chonov, M. Werner, and R. Steinmetz. On the Accuracy of Appliance Identification Based on Distributed Load Metering Data. In *Proceedings of the 2nd IFIP Conference on Sustainable Internet and ICT for Sustainability (SustainIT)*, 2012.

[21] M. Rossi, S. Feese, O. Amft, N. Braune, S. Martis, and G. Troster. AmbientSense: A Real-Time Ambient Sound Recognition System for Smartphones. In *International Conference on Pervasive Computing and Communications*, 2013.

[22] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell. NextPlace: A Spatio-temporal Prediction Framework for Pervasive Systems. In *Pervasive Computing*. 2011.

[23] I. Schweizer, R. Bärtl, A. Schulz, F. Probst, and M. Mühläuser. NoiseMap-real-time Participatory Noise Maps. In *Proceedings of the 2nd International Workshop on Sensing Applications on Mobile Phones*, 2011.

[24] E. Tapia, S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman. Real-Time Recognition of Physical Activities and Their Intensities Using Wireless Accelerometers and a Heart Rate Monitor. In *11th IEEE International Symposium on Wearable Computers*, 2007.

[25] L. Vu, Q. Do, and K. Nahrstedt. Jyotish: A Novel Framework for Constructing Predictive Model of People Movement from Joint Wifi/Bluetooth Trace. In *International Conference on Pervasive Computing and Communications*, 2011.

[26] L. Vu, K. Nahrstedt, S. Retika, and I. Gupta. Joint Bluetooth/Wifi Scanning Framework for Characterizing and Leveraging People Movement in University Campus. In *Proceedings of the 13th ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*, 2010.

[27] H. Ye, T. Gu, X. Zhu, J. Xu, X. Tao, J. Lu, and N. Jin. FTrack: Infrastructure-free Floor Localization via Mobile Phone Sensing. In *International Conference on Pervasive Computing and Communications*, 2012.

[28] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang. Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones. In *Proceedings of the 8th IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, 2010.