Ad Hoc Networks 11 (2013) 1539–1555

Contents lists available at SciVerse ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Energy monitoring in residential spaces with audio sensor nodes: TinyEARS ☆

M. Amac Guvensan [a], Z. Cihan Taysi [a], Tommaso Melodia [b,*]

[a] Department of Electrical Engineering University at Buffalo, The State University of New York, 230 Davis Hall, Buffalo, NY 14260, United States
[b] Department of Electrical Engineering University at Buffalo, The State University of New York, 332, Bonner Hall, Buffalo, NY 14260, United States

ARTICLE INFO

Article history:
Received 5 January 2012
Received in revised form 30 July 2012
Accepted 8 October 2012
Available online 26 October 2012

Keywords:
Energy monitoring
House appliances
Audio data classification
Wireless audio sensor networks

ABSTRACT

Awareness on how and where energy is consumed is being increasingly recognized as the key to prevent waste in next-generation smart buildings. However, while several solutions exist to monitor energy consumption patterns for commercial and industrial users, energy reporting systems currently available to *residential* users require time-consuming and intrusive installation procedures, or are otherwise unable to provide device-level reports on energy consumption. To fill this gap, this paper discusses the design and performance evaluation of the Tiny Energy Accounting and Reporting System (TinyEARS), an energy monitoring system that generates device-level power consumption reports primarily based on the acoustic signatures of household appliances detected by wireless sensors. Experiments demonstrate that TinyEARS is able to report the power consumption of individual household appliances within a 10% error margin.

1. Introduction

Residential spaces account for approximately 21% of the total energy consumption in the United States [2], with raising figures worldwide. This motivates the growing interest in smart building technology to automate and otherwise promote energy conservation in residential and commercial spaces. Several techniques, including insulation and energy harvesting from solar panels, or integration with pervasive sensing and actuation technologies, are being proposed and discussed to reduce the carbon footprint of buildings by preventing energy waste.

However, while businesses rely increasingly on procedures and best practices to save energy and reduce costs, most home users do not have any means to control their energy usage patterns. The main resources in a typical household are electricity, water, natural gas, and heating oil. Saving a small portion of each in each residential space could have a significant impact on reducing costs, energy consumption, and impact on the environment. Several studies have shown the necessity of fine-grained energy monitoring to encourage conservation. In [3], Darby explored the effectiveness of different forms of feedback on energy consumption, such as self-meter-reading, interactive feedback via a PC, frequent bills based on readings plus historical feedback, among others, and emphasized the necessity and the benefits of direct feedback mechanisms. He concluded that immediate direct feedback can be extremely valuable in influencing behavior with savings in the range of 5–15%.

Stern [4] states that awareness not only encourages people to actively participate in doing the right thing in preventing waste of energy, but also provides indirect monetary benefits because of the reduced cost of resources. Therefore, a great promise lies in monitoring systems designed to reduce energy wastage in residential

☆ A preliminary version of this paper was accepted for publication in Proc. of ACM Workshop on Embedded Sensing Systems for Energy Efficiency in Buildings (BuildSys), Zurich, Switzerland, November 2010 [1].

* Corresponding author. Tel.: +1 716 645 3115x1147; fax: +1 716 645 1027.

E-mail addresses: amac@ce.yildiz.edu.tr (M.A. Guvensan), cihan@ce.yildiz.edu.tr (Z.C. Taysi), tmelodia@buffalo.edu (T. Melodia).

1570-8705/$ - see front matter
http://dx.doi.org/10.1016/j.adhoc.2012.10.002

spaces by reporting device-level energy usage information to the home user, with potential reductions in costs and emissions of harmful gases like CO_2.

Motivated by this, in this paper we discuss the design and implementation of a wireless monitoring system for residential spaces based on a multi-layer decision architecture and called Tiny Energy Accounting and Reporting System (TinyEARS). TinyEARS is based on wireless audio sensor nodes and a real-time power meter deployed to monitor device-level energy consumption in the household. The objective of TinyEARS is to detect and classify "on" devices based on their acoustic signatures and report device-level energy consumption by correlating node decisions with time and power information obtained from a real-time power meter. Although the multi-layer decision architecture of TinyEARS is currently implemented by using audio sensors only, clearly the coordinated use of multiple types of sensors, such as light and magnetic, would increase the system performance. There are also silent devices, such as modem, laptop, and mobile phones, which cannot be detected by their acoustic signatures. The energy consumption of these devices can be measured by using an additional node equipped with magnetic sensors [5].

However, the main focus of this work is to show how acoustic signatures of house appliances can be used to enable device-level monitoring of energy consumption in residential spaces, with (surprisingly) excellent performance. The key component of TinyEARS is what we refer to as the house appliance sound recognition system (HASRS), which we implemented on Imote2 [6] sensors to obtain node-level decisions in real-time. The HASRS employs the mel-frequency cepstral coefficients (MFCC) as a feature extraction algorithm and minimum distance classifier (MDC) as a classification algorithm.

The paper discusses the architecture, communication subsystem, signal acquisition and processing, and algorithmic details of TinyEARS. The key contributions of our work are as follows:

- We propose a multi-layer architecture that enables energy monitoring at the device level by leveraging the acoustic signatures of house appliances. This information is correlated with the overall power usage information of the house obtained from a real-time power meter.
- While most existing solutions for device-level monitoring require one sensor node per appliance, in TinyEARS a single sensor node monitors house appliance in a room based on their acoustic signatures. Deploying one sensor node per room reduces both the overall cost and communication burden of the wireless sensor network.
- TinyEARS, being based on a limited number of sensor nodes, is an easily deployable and maintainable system.
- We show that house appliances can be recognized with an overall success rate of 94% by their acoustic signatures with relatively simple processing algorithms implemented on the motes.
- Finally, we discuss the main system design challenges and their solutions in implementing an audio classification process on an Imote2 sensor node.

The rest of this paper is organized as follows: In Section 2, we review existing solutions for energy monitoring in residential spaces. In Section 3, we introduce the system architecture of TinyEARS. Section 4 describes the details of the house appliance sound recognition system. In Section 5, we describe the data acquisition and correlation process of TinyEARS. In the same section, we also discuss the issues and challenges of implementing the HASRS on Imote2. Section 6 describes our experimental environment and presents the test results obtained using Imote2 and the TED5000 power meter.

2. Related work

Previous work on monitoring energy consumption in residential spaces, shown in Table 1 can be broadly classified into two categories. The first group is concerned with systems designed to measure the overall energy consumption with a single sensor, usually located in a power box. The second approach is to monitor each household appliance individually, with fine-grained consumption feedback.

There exists several commercially available products to measure the overall energy consumption of a household, including Power Cost Monitor [7], Wattson [8], Onzo [9], EM-2500 [10] and TED-xxx [11]. These products generally consist of two units, a central unit connected to a fuse box and a display unit. Even though it is moderately difficult to install these devices, they do not require any maintenance once deployed. They have simple user interfaces reporting power usage in scales of seconds or minutes. However, while these products are able to present the overall energy consumption and detect anomalies in energy usage, they cannot provide per-appliance energy measurement. The Google Power Meter [12] is another option to visualize power meter readings. It is an opt-in software tool that allows users to visualize detailed home energy information. A secure Google gadget displays data on home energy consumption received from either a smart meter or another electricity monitoring device.

The Non-Intrusive Load Monitoring (NILM) [13], uses a single whole-house energy meter and attempts to identify and disaggregate the energy usage of individual devices by identifying changes in steady-state levels where the power is constant. The success rate of the NILM system varies between 75% and 90% in detecting the on/off status of working household appliances [14]. The key limitations of NILM are that loads are indistinguishable, and data is batch-processed [13]. To simplify the training process of NILM systems, in [15], the authors present an inexpensive contactless electromagnetic field event-detector that can detect appliance state changes within close proximity, based on magnetic and electric field fluctuations.

Device-level monitoring has recently received attention in the literature because of its fine-grained feedback on energy consumption. To monitor each device individually, two approaches have been considered in the literature, i.e., (i) installing electrical current sensors inline with each appliance, and (ii) deploying multiple sensors throughout the household. Some commercial products, based on

Table 1
List of most remarkable studies about monitoring energy consumption of residential spaces.

Study	Method	Advantages	Disadvantages	Data sensing type	Accuracy (%)	
[13,14]	NILM	Easy to deploy/inexpensive	Loads are indistinguishable Loads are steady-sate	Power signals	Direct*	75–90
[16,18,20]	Whole-energy monitoring	Easy to deploy/inexpensive	Coarse-grained Energy monitoring	Power signals	Direct	100
[15]	Device-level monitoring	Easy to deploy/low-cost	Complex devices (HVAC) not traceable	Electric/Magnetic signals	Indirect	70
[5]	Device-level monitoring with wireless sensor nodes	No inline sensor	Too many sensor nodes	Ambient signals (Magnetic, Light, Acoustic)	Indirect	90
		Fine-grained energy monitoring/ Inexpensive sensors				
[17,19]	Installing electrical current sensors inline with each appliance	Fine-grained energy monitoring	Very expensive	Electrical current signals	Direct	100
			Hard to deploy			
[26]	NILM with	Easy to deploy/inexpensive	High Computational Cost	Power signals	Direct*	96
	Kernel-based subspace Classification					
[28]	Single plug-in sensor with ML techniques	Easy to deploy/inexpensive	Loads are steady-state	Electrical Noise	Direct*	85–90%
		Not scalable				
[24]	Circuit-level	Better to monitor devices with complex states or continuously power usage	Hard to deploy/ costly	Power signals	Direct	95
	Energy measurement					

NILM: Non-intrusive Load Monitoring.
ML: Machine Learning.
Direct: Direct sensing with Multiple-Point Installation.
Direct*: Direct sensing with Single-Point Installation.
Indirect: Indirect sensing using wireless sensor nodes.

electrical current sensors, are already available, e.g., Plogg [16], Kill-a-Watt [17], Tweet-a-Watt [18], Watts Up [19] and Smart Insteon Central Controller [20]. Although these products provide fine-grained energy monitoring, they require inline installation between a standard AC plug and the outlet. Therefore, appliances such as heating and ventilation systems (HVAC), electric boilers and ceiling lights cannot be easily instrumented, since they lack AC plugs and have wired connections.

Researchers have recently proposed deploying sensor networks in buildings to achieve device-level monitoring of energy consumption. Shah et al. [21] discuss issues and challenges in monitoring energy consumption with a sensor network. In [5], the authors developed Viridiscope, a system designed to provide device-level power consumption feedback by using magnetic, acoustic and light sensors. Ambient signals from sensors placed near appliances are used to estimate power consumption. Viridiscope provides a very detailed picture of household energy consumption, at the expense of significant additional hardware cost - at least one sensor or meter per home device. Jiang [22] et al. deployed a wireless sensor network for high-fidelity monitoring of electrical usage in buildings. This network consists of 38-mote-class AC meters, six light sensors, and one vibration sensor which are placed at carefully chosen sampling points. The authors have explored several techniques for modeling, estimating, and disaggregating energy usage across functional, spatial, user and signal domains.

Schoofs et al. [23] propose ANNOT (Automated Electricity Data Annotation), a system to automate electricity data annotation leveraging cheap wireless sensor nodes such as temperature, sound, light and accelerometer. ANNOT automatically acquires appliance signatures, collects training data, validates the monitoring output without human supervision, and is integrated within the RECAP (Recognition and Profiling of Appliances) appliance load monitoring system.

Marchiori and Han [24] explore an alternative approach to monitor household energy usage, including small devices. They propose using circuit-level energy measurements as a compromise between the two aforementioned approaches. With this objective, they identify and separate the energy usage of individual devices by applying analysis inspired by non-intrusive load monitoring. The drawback of this approach is that as many load controlling devices as the number of monitored house appliances have to be installed on the circuits.

Another possible approach is to try to infer device-level consumption using only power meter readings. In [25], the authors interpret power meter readings to estimate the power consumption of individual electric appliances. They use machine learning algorithms including multi-layered perceptron, radial basis function network, and support vector regressors to find the power consumption of inverter-type appliances. In [26,27], the same authors also study the characteristics of electric appliances that show a constant power consumption. Existing studies also analyze the signature of devices based on electrical noise on the residential power line. For example, Patel et al. [28] developed a system that tries to predict active devices at any given time. Finally, in [29], the authors exploit current and voltage waveforms to classify working devices. The system can determine the status of an appliance, but not its actual power consumption.

Finally, several studies use wireless sensor networks for energy efficiency without directly addressing energy monitoring in residential spaces. For example, in [30] the authors create user profiles through a prediction algorithm using physical parameters like light and temperature. Thus, the system has the ability to automatically set system parameters to minimize the energy consumption while guaranteeing a desired comfort level. In [31], Delaney et al. propose a wireless sensor network based tool that evaluates lighting control systems in existing office buildings. The tool is used to determine points in the control system that exhibit energy wastage and to highlight areas that can be optimized. Gao and Whitehouse [32] deployed a wireless sensor network to monitor house occupancy and consequently regulate the house temperature automatically. They show that by optimizing setback schedules based on home occupancy patterns with the help of a self-programming thermostat energy savings up to 15% can be achieved. In [33], the authors propose a system that estimates occupancy with an accuracy of 80%. They construct multivariate Gaussian and agent based models for predicting user mobility patterns in building. Their simulations indicate a 14% reduction in HVAC energy usage by having an optimal control strategy based on occupancy estimates and usage patterns. Marchiori and Han present a prototype distributed control system [34] for building energy management that uses wireless sensor network-class nodes. This prototype system achieved an energy savings of up to 15% by implementing a relatively simple control policy.

TinyEARS aims at monitoring house appliances and their energy usage with the help of a wireless sensor network deployed at home and a wireless power meter connected to the electric panel. The system differs from available solutions by combining the information from acoustic signatures of house appliances and power meter readings through a multi-layer architecture. TinyEARS allows deploying one single sensor node per room. The system can potentially combine the individual features of different types of sensors such as light, temperature and magnetic to improve the accuracy of the estimated energy consumption in a residential space. For instance, silent devices, such as modems, charging devices, lights, could not be tracked via audio sensors, since they do not produce any sound or produce unrecognizable sound during their operation. Thus, to detect the on/off status of these devices and to measure their power consumption, we suggest to use extra sensor nodes equipped with sensors such as light, temperature and magnetic. Viridiscope [5] is a very good example of such use of magnetic sensors. These sensors could be easily adapted to our system to improve the accuracy of the estimated energy consumption in a residential space. However, this paper is focused on the detection of household appliances emitting acoustic signals.

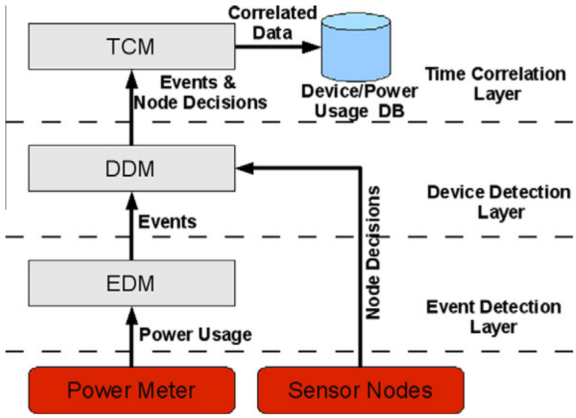


Fig. 1. Multi-layer decision architecture of TinyEARS.

3. System design

TinyEARS is composed of a multi-layer decision architecture that includes (i) an event detection layer, (ii) a device detection layer and (iii) a time correlation layer, as shown in Fig. 1. Operations within each layer are handled by individual software modules running on the Data Fusion Center (DFC). The Event Detection Module (EDM) communicates with the real-time power meter and uses a basic filtering mechanism on the real-time power consumption readings to detect possible changes in the use of house appliances. Details of the EDM module is given in Section 5.2. When an event is detected, the EDM alerts the Device Detection Module (DDM).

The DDM controls communication with the sensor nodes through control packets that trigger sensor nodes to start collecting audio samples. Based on the collected samples, each individual sensor decides which (if any) house appliances are active. Each sensor node reports its decision back to the DDM along with a confidence value. The DDM sends event information and node decisions to the Time Correlation Module (TCM). The TCM may decide to rely on the node decision or override it with a better match. The TCM creates alternative decisions by correlating time, power usage and node decisions. A set of rules have been defined for the correlation operation, whose details will be discussed in Section 5.5.

TinyEARS employs a real-time wireless power meter and wireless sensor nodes to detect the energy consumption of individual house appliances. The deployment of TinyEARS consists of a few basic steps: (i) installation of a real-time power meter, (ii) deployment of one sensor node in each room, and (iii) setup of a PC or laptop as DFC. A real-time power meter must be connected to the main power line of the house. Based on our experience, this is a very simple operation that takes less than 10 min. The deployment of the sensor nodes does not have any specific requirements: sensors can be placed in any position within the room. In addition, mobility of house appliances does not effect the system performance. For example, moving of a vacuum cleaner within a room has a negligible impact on the success rate of the audio classification procedure performed on the sensor node. The setup of the DFC includes installation of several software modules and drivers. A sample deployment of TinyEARS is shown in Fig. 3.

3.1. System elements

TinyEARS consists of three main components: a real-time power meter, a DFC, and sensor nodes. The real-time power meter is used to monitor changes in power usage in the household. It must be able to report power usage every second. There are several off-the-shelf power meters with different capabilities and prices. We have used the TED 5000 produced by Energy Inc., because of its simple installation procedure, very accurate readings, and open development API. The TED 5000 system consists of three units: a measuring transmitting unit (MTU), a gateway, and a user display unit. The MTU is mounted on the main electric panel of the house as shown in Fig. 2. It measures and transmits energy, power, and voltage information to the gateway, which can be plugged to any outlet in the house. The gateway relays data to a user display unit via an IEEE 802.15.4 connection. In addition, real-time and historical data can be accessed by utilizing the gateway's Ethernet interface.

The DFC is a typical PC that runs EDM, DDM, TCM modules, and a configuration utility to manage the initial training of TinyEARS. The DFC reads power usage data from the power meter every second. If the DFC decides that a house appliance has been turned on or off based on these readings, it issues a command instruction to the sensor nodes to sample audio. Then, it waits until node decisions have been received.

In TinyEARS, sensor nodes are required to perform signal processing operations including Fast Fourier Transform (FFT), feature extraction and classification. Therefore, we have used the Imote2 sensor node platform that can satisfy the processing and memory requirements of these algorithms. The Imote2 is built around the low-power PXA271 XScale processor and integrates an IEEE 802.15.4 radio (CC2420) with a built-in 2.4 GHz antenna. The Imote2 is a modular stackable platform and can be expanded with extension boards to customize the system to a specific application.

The Imote2 sensor platform is distributed with TinyOS or.Net Micro Framework. The TinyOS system, libraries, and applications are written in nesC, a programming language designed for structured component-based applications. The nesC language is primarily intended for embedded systems such as sensor networks. Although nesC has a C-like syntax, porting existing C libraries to a particular mote platform requires manually re-writing all parts of the code that allocate buffers due to the static nature of nesC [35]. Conversely, the.Net Micro Framework is a modular platform that allows reusability of existing C/C++ libraries with minor modifications. Our processing-intensive application required porting a math library and a FFT method, which made the.Net Micro Framework a natural choice. In addition, the.Net Micro Framework provides a simple installation process and a visual development environment that reduces the complexity of programming and debugging.

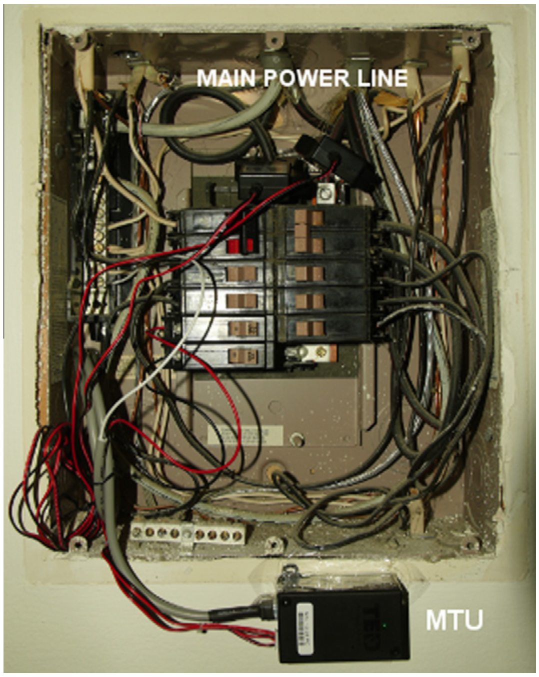


Fig. 2. MTU connection.

3.2. Configuration of the TinyEARS

During the initial setup of TinyEARS, a training phase is required. After deploying one sensor node in each room, house appliances are introduced to the HASRS. This process is performed through a user interface on the DFC. For each device, the user basically selects the room that the device will be placed into and enters a label for that device. Then, that appliance should be turned on for the sensor node in that room to collect audio samples from the device. The sensor node extracts audio features from the audio samples and adds these features to its classification database. Whenever a new house appliance with an acoustic signature is added, it needs to be introduced into TinyEARS.

4. On-board real-time device classification

Different house appliances are characterized by different levels of current absorption. However, it is not easy to discriminate devices based only on power meter readings or on electrical noise on power lines. Therefore, the key idea behind TinyEARS is to infer the energy consumption of house appliances by primarily relying on their acoustic signature.

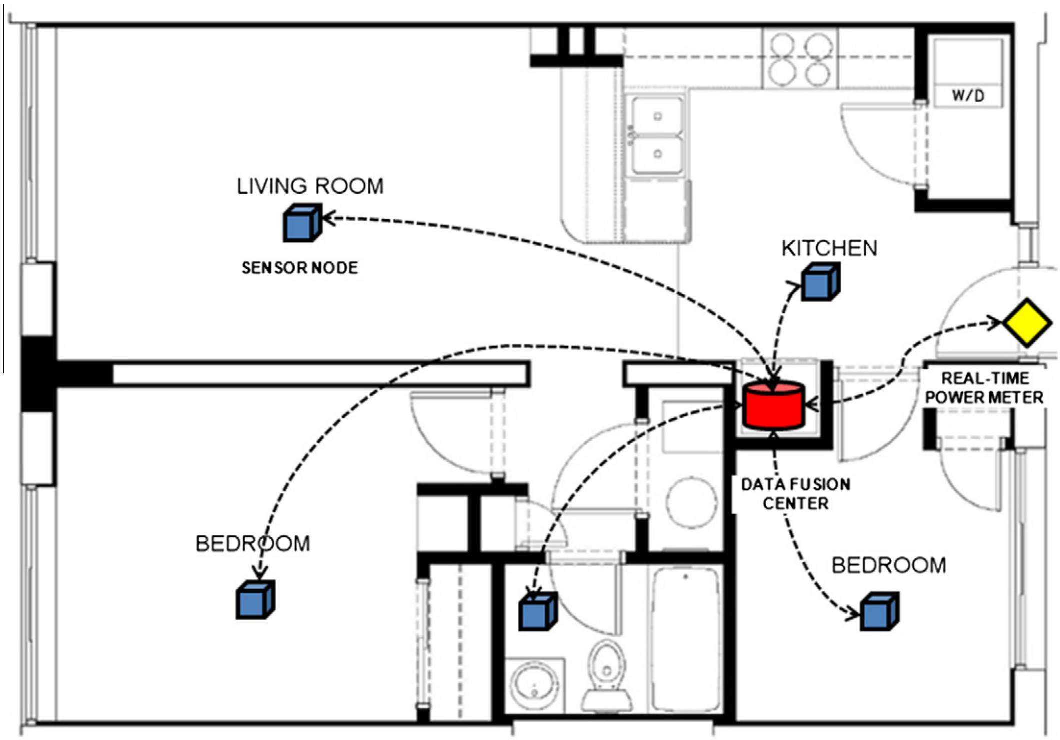


Fig. 3. A sample deployment of the TinyEARS system.

In previous studies, researchers have explored sound classification problems especially for speech, speaker, and musical instrument recognition. In recent years, the problem of classifying environmental sounds has also attracted significant interest. For example, Cowling [36] has explored the environmental sound classification problem in detail. Environmental sounds can be broadly classified into two groups. The first group consists of sounds that start abruptly and have short duration. This includes animal sounds, the sound of a breaking glass, or a door creaking, among others [37]. The second group consists of environmental sounds that appear in a specific frequency band and last longer compared to the first group. This includes the engine of a car or plane, refrigerators, and hair dryers, among others. Sounds in the latter group are generally produced by the motor of a machine and can be categorized as mechanical sounds. The fact that most house appliances (e.g., refrigerator, dishwasher, exhauster, blender, vacuum cleaner, washer, and hair dryer) contain a motor suggests that it is possible to classify them based on their acoustic signature. To the best of our knowledge, the sound characteristics of house appliances were only explored in [38]. Thus, we exploit the feature extraction and classification algorithms proposed in that study.

4.1. Audio feature extraction

Feature extraction from an audio signal is one of the most important issues in the field of audio data classification. Audio data generally includes two types of features, i.e., *physical and perceptual features*. Physical features are low-level signal parameters that capture aspects of the temporal and spectral properties of the signal, whereas perceptual features refer to loudness,pitch, and timbre that can be differentiated easily by human ears.In this study, seven different audio features are primarily considered, including zero-crossing rate (ZCR), short-time energy (STE), band-level energy (BLE), spectral-centroid (SC), spectral roll-off (SRO), spectral flux (SF), and mel-frequency cepstral coefficients (MFCCs) [36]. To obtain a high classification accuracy, we have assessed the effectiveness of each of these features. Based on this, we have selected effective and robust feature combinations for discriminating the acoustic signatures of household appliances. First, we have analyzed the power spectral density of the acoustic signature of each house appliance via AudaCity, an audio signal processing tool. This preliminary analysis shows that the discriminative frequency band is between 0 and 1 kHz. Before implementing our algorithms on the sensor nodes, we ran extensive MATLAB tests to validate our feature set. In Fig. 4, we report a summary of our results that clearly demonstrate how the MFCC features perform better than any other group of physical features.

4.1.1. Mel frequency cepstral coefficients

The MFCC, introduced in [36] as a candidate feature extraction method for stationary audio signals, perform better than other feature extraction methods such as Fourier transform, human factor cepstral coefficients, fast wavelet transform and short-time Fourier transform. Since

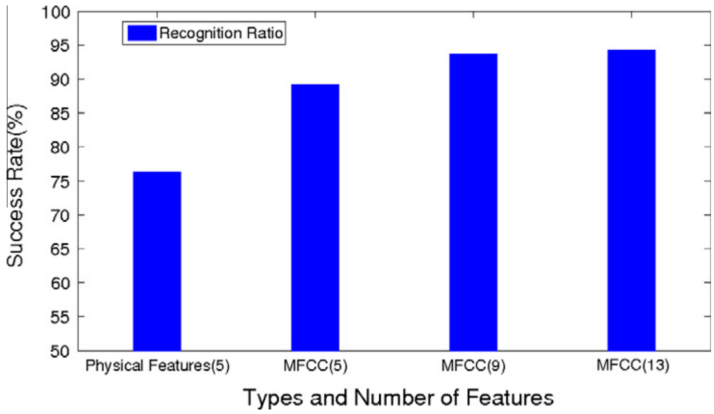


Fig. 4. Comparison of different types and number of features.

the sounds of house appliances can be considered to be stationary, we applied the MFCCs for our audio classification problem. There are five main processing steps for obtaining MFCC features of an audio signal, as illustrated in the diagram reported in Fig. 5. First, the audio signal is fragmented into frames of predefined length. Then, each frame is multiplied with a Hamming window to maintain the continuity between the first and the last points in the frame. Then, the FFT is applied to the signal and smoothened by a series of triangular filters, including 13 linear filters below 1 kHz and 27 logarithmic filters within 1–6.4 kHz. The MFCCs are finally calculated.

The complete coefficient extraction procedure [39,40] can be described step-by-step as follows:

Step 1: The digital audio signal $s[n]$ is fragmented into frames of length F.

Step 2: Each individual frame is windowed to minimize signal discontinuities at the beginning and end of each frame. The idea here is to minimize the spectral distortion by using the Hamming window to taper the signal to zero at the beginning and end of each frame. The Hamming window $hw[n]$ is given by

$$hw[n] = \left(0.54 - 0.46\cos\frac{2\pi n}{F-1}\right), n = 0, \cdots, F-1. \tag{1}$$

where F is the frame size.

Step 3: All frames are transformed to the frequency domain by applying the FFT:

$$Y(t) = \frac{1}{F}\sum_{n=0}^{F-1} s[n]hw[n]e^{-jwnt}, \qquad w = \frac{2\pi}{F}, \quad t = 0, \ldots, F-1. \tag{2}$$

Step 4: The energy spectrum is calculated as

$$X[l] = |Y[l]|^2 \tag{3}$$

Step 5: The energy in each frequency channel is calculated as

$$S[k] = \sum_{j=0}^{F/2-1} W_k[j]X[j], \qquad W_k = e^{-j2\pi k/F}, \quad k = 0, \ldots, M-1. \tag{4}$$

In (4), M represents the number of mel windows in the mel scale. The mel scale, proposed by Stevens, Volkman and

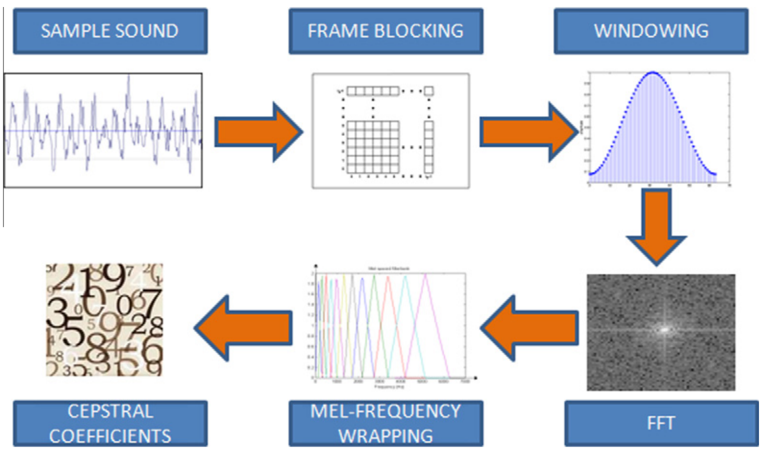


Fig. 5. A flowchart of MFCC.

Newman in 1937, is a perceptual scale of pitches judged by listeners to be equal in distance from one another. We can convert f Hz into m mel through

$$m = 2595 \log 10 \frac{f}{700} + 1. \tag{5}$$

$W_k[j]$ is the triangular filter associated with the kth channel in the mel scale. $W_k[j]$ satisfies the following constraint:

$$\sum_{j=0}^{F/2-1} W_k[j] = 1, \quad \forall k. \tag{6}$$

Step 6: Logarithm and cosine transforms are applied to the audio signal. Here, $c[n]$ represent the amplitude coefficients of the resulting spectrum, i.e., the MFCCs of the given audio signal.

$$c[n] = \sum_{k=0}^{M-1} \log S[k] \cos\left[n(k+0.5)\frac{\pi}{M}\right], \tag{7}$$

where $n = 1, \ldots, L$, and where L is the desired order of the MFCC.

As mentioned before, an analysis of the power spectral densities of each house appliance shows that the discriminative frequencies are concentrated below 1 kHz (Fig. 8). Thus, we omitted logarithmic filters that are applied only to the frequencies above 1 kHz and evaluated our audio classification process for different number of cepstral coefficients to compare the recognition success rate.

4.2. Audio classification

In the classification step, the objective is to recognize "on" devices based on MFCC features. Several classification algorithms, including support vector machines (SVMs) and k-nearest neighbor (k-NN), have been previously proposed for environmental sound classification [36].

SVM [41] is a binary classifier widely used for solving two-class problems. SVMs find the hyperplane that maximizes the margin, described as the distance from the hyperplane to the instances closest to it on either side, between the two classes. Conversely, our problem consists of n classes, i.e., each room may have n different house appliances. Moreover, there may be multiple devices running simultaneously. TinyEARS uses a similarity check function to differentiate the sound of multiple house appliances from the sound of individual devices. The similarity check function estimates how close a given audio sample is to the audio classes that constitute a training set. The distance between a given sample and the existing classes, as expressed by the similarity check function, is then used as an estimate of the reliability of the decision of a sensor node. We have therefore considered two computationally inexpensive classification algorithms, i.e., minimum distance classifier (MDC) and k-NN, which are able to associate a distance function to each decision. This distance is used as the similarity check function to quantify the level of confidence in the decision. The main advantage of MDC against k-NN is that MDC performs the similarity test only with the centroid of available acoustic signatures of household appliances, whereas the k-NN [41] algorithm tests a given sample against all individual audio samples in the database. Given n classes with m samples for each class, the computational complexity of k-NN and MDC are nm and n, respectively.

During our preliminary design phase, we compared the household appliance recognition success rates of MDC, k-NN and SVM through Matlab simulations. Our findings, reported in Table 2, show that the success rates do not vary significantly for the considered application. These results are obtained using nine MFCC features for seven house appliances and their different running levels with 120 audio samples for each. The performance of k-NN and SVM is only slightly better than MDC. Therefore, even though Imote2s have the capability to run complex classification algorithms, we decided to use a simple classifier as MDC, characterized by a relatively low computational cost, to speed up the decision process on each sensor node and reduce the energy consumption.

4.2.1. Minimum distance classifier

The minimum distance classifier algorithm aims at finding the closest class centroid to the given test sample. A centroid for each class is calculated based on the available features. To compute the centroid of a class, the algorithm calculates the mean of those features individually. Let **x** be an unknown sample to be classified, and $\mathbf{y}_i$, $i = 1, \ldots, n$ be a prototype for class $(\rfloor_i)$. Both **x** and $\mathbf{y}_i$ are m-dimensional vectors in the feature space, n is the number of classes, and m is the number of dimensions of the feature space.

The minimum distance classifier is defined as

$$x \in (\rfloor_i) \iff d(\mathbf{x}, \mathbf{y}_i) = \min_{j \in 1,\ldots,n} d(\mathbf{x}, \mathbf{y}_j),$$

where $d(\cdot)$ is the Euclidean distance function

$$d(\mathbf{x}, \mathbf{y}_i) = \sqrt{\sum_{k=1}^{m} (x_k - y_{ik})^2}.$$

4.3. House appliance sound recognition system

HASRS uses a supervised learning method, MDC, where the system consists of two main phases, *training* and *test* processes.

4.3.1. Training process

The system needs to be trained using the sound of each house appliance to generate their acoustic signatures. The training process is done only once after deploying the sensor nodes. This process takes place on the DFC. The acoustic signatures of each house appliance are generated at the end of this process.

There may be different house appliances in each room. Some of them, like a vacuum cleaner, may be used in

Table 2
Comparison of k-NN, SVM and MDC with nine MFCC features.

	MDC	k-NN	SVM
Success rate (%)	93.69	97.14	97.74

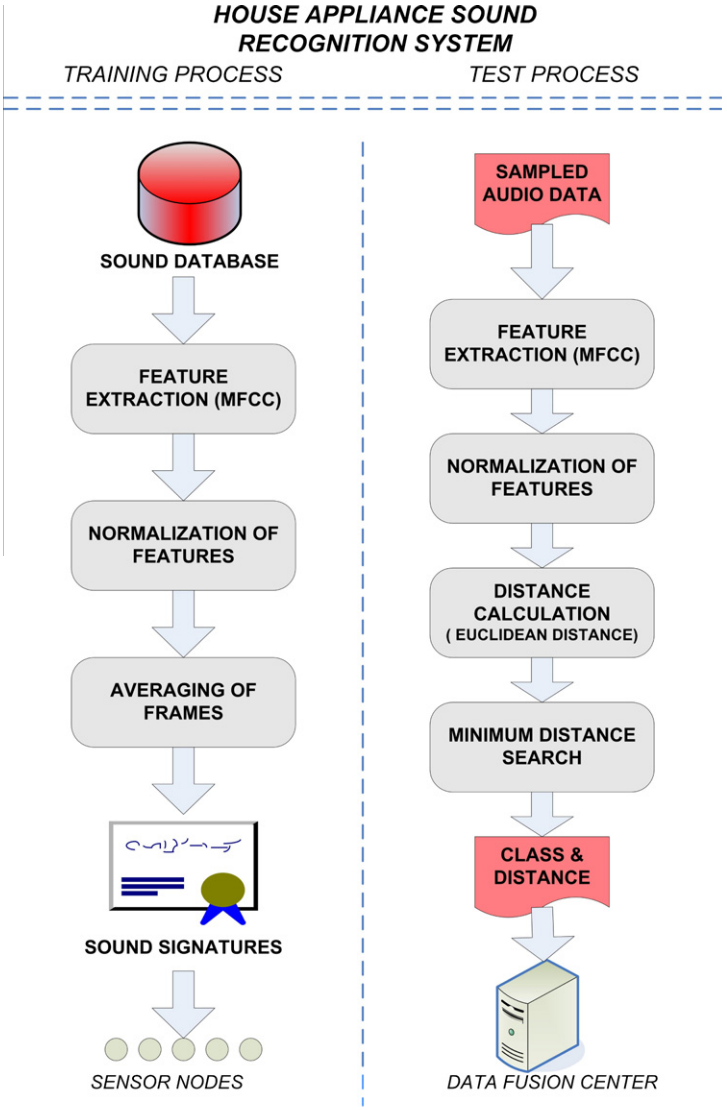


Fig. 6. Block diagram of the classification process.

different rooms. Therefore, this type of devices should be included in the training list of each room. For example, in addition to refrigerator, exhauster, blender, and dishwasher, the vacuum cleaner is counted as a new class independent of other rooms.

After training the system on the DFC, the mean values and scale coefficients of each class, which are used to normalize the feature set, are delivered to the corresponding sensor node in the house. Fig. 6 describes the main steps of the training process. First, the training software module (TSM) loads all audio samples of each class from the sound database into the memory. Each class has an equal number of samples. The TSM divides then each audio sample into frames of equal length. The frames are extracted into its MFCC features. Next, samples are normalized to report values of each feature on the same scale. Thus, each feature is weighted equally in the process of distance calculation. Then, the TSM generates the acoustic signature of each device and its run levels. In the last step, sound signatures are delivered to the corresponding sensor node in the house.

The training process occurs only once before the deployment of acoustic signatures. If a new device needs to be added, only the acoustic signature of the new device would be generated and then the new signature would be deployed to the nodes placed in each room through the DFC.

The acoustic characteristic of an house appliance do not change too much during its running period. Only different states of an house appliance could produce distinctive sounds. Therefore, in our tests we have sampled three times (at different time points) 30 s sound of each state of each house appliance. Each 30 s period has been split into 1 s chunks. Then, these samples have been labeled and loaded into the sound database.

After several test scenarios, we observe that increasing sampling time above 30 s do not improve the success rate of HASRS. Besides, since MDC aims at finding the mean of each feature, the HASRS do compare test data to only one value for each feature, contrary to k-NN. Thus, more training data would be unnecessary in our system, whereas much more training data in k-NN might increase the system performance.

4.3.2. Test process

In the test phase, after sampling audio data, the sensor node extracts the MFCC features of the signal. Then, it calculates the Euclidean distance to each class. The class with minimum distance and the distance are sent to the DFC. Fig. 6 shows a detailed diagram of the test process module (TPM). First, the sampled audio is divided into frames. The TPM normalizes each frame using scale coefficients. In the next step, the TPM invokes the similarity check function that calculates the distance of the sampled audio data to the defined classes, i.e., devices. Finally, the minimum distance function determines the most similar class and the sensor node transmits related information to the DFC.

5. Data acquisition and processing

TinyEARS requires real-time power usage information and node decisions to estimate the power consumption of individual house appliances. Several modules have been implemented to gather and process information.

5.1. Interfacing the power meter

TED5000 provides an XML-based development API to retrieve real-time and historical data from the gateway. Interfacing with the real-time power meter involves two operations. First, an HTTP request is sent to the TED5000 gateway. The gateway replies back with a XML file that contains information including a time stamp, voltage, and power from several MTUs. In the second step, this XML file is parsed to extract relevant information, i.e., time stamps and power usage of a particular MTU. These two operations are performed by an application running on the DFC. This application has been developed in the C# programming language, since C# provides built-in class libraries for handling HTTP requests and XML files.

5.2. Data/event filtering

The DFC monitors power changes in the household and decides whether a device has been turned on or off. In our experiments, we have noticed that the power consumption of house appliances might change due to two reasons. First, the power consumption of some devices in the household such as vacuum cleaner varies in time based on their usage. Second, some devices cause a peak in power consumption when they are turned on and then descend to a lower power consumption level.

Thus, basic approaches like using the difference between the last two readings fail to correctly detect changes in device states. In our system, we employ two filtering techniques to overcome this problem. First, when a change occurs between the last two readings, the difference in power usage is tested against a predefined threshold. The purpose of this threshold is to eliminate the effect of devices with varying power consumption. If the change is greater than the threshold, a second filter is applied. The DFC keeps monitoring the power usage for five additional seconds and calculates the average of these values. This low-pass filter is used to remove peaks and estimate the actual steady-state change in current absorption caused by the device. This value is also used by the TCM to validate and (possibly) override node decisions.

5.3. Audio sampling at the sensor node

The ITS400 [42] sensor board is designed to be interfaced with Imote2. It contains a three-axis accelerometer, an advanced temperature/humidity sensor, a light sensor and a 4-channel Analog/Digital Converter (ADC). The ADC on this board can be used to interface with different analog sensors. In our application, one of the four available ADC channels is used to interface an electret microphone [43]. The ADC on the ITS400 sensor board can provide 10- and 12-bit samples. To preserve the quality of the captured sound, we use 12-bit samples. Fig. 7 shows the microphone connected to the Imote2 sensor node.

Our experiments show that the discriminating frequencies of house appliances are in the 0–1 kHz range. Thus, a sampling rate of 2 kHz is satisfactory. Fig. 8 shows the frequency spectrum of audio samples of refrigerator as an example.

A common way of sampling from audio sensors is to program a timer with the desired resolution and sample the ADC channel when the timer interrupt occurs. This method optimizes the CPU usage and allows other tasks to run concurrently. However, in the.Net Micro Framework the timer resolution is 1 ms and a single read operation from one of the ADC channels takes approximately 500 μs. Thus, the maximum sampling rate that can be achieved by using the built-in timer is around 600 Hz. An alternative way is to sample the ADC channel, and insert delays between readings to meet the desired resolution. Although this is clearly not an ideal solution, we had to rely upon this method to achieve the required 2 kHz sampling rate.

5.4. Implementing device classification on Imote2

In our study, we have implemented the HASRS component of TinyEARS on Imote2. Each mote is responsible for its individual room/space. Motes wait in idle state until they receive a command from the DFC. In our tests, the mote is always in the "on" state and monitors the channel for command messages. After receiving the command, the mote performs audio classification including sampling audio data, extracting features (MFCC), processing minimum distance classifier. After finalizing a decision, each mote transmits the decision class label and the distance value to the DFC.

In our implementation, the mote samples 4 s long audio clips. Each clip is segmented into 25 ms long frames that

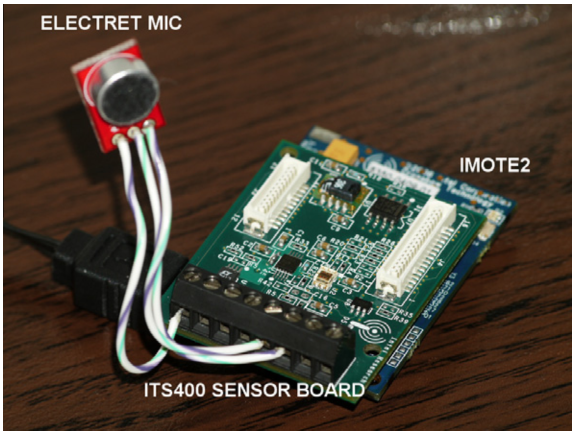


Fig. 7. Imote2, ITS400 and electret microphone.

overlaps 10 ms with the previous frame. The Hamming window, FFT, Mel-frequency bank filters and Discrete Cosine Transform are applied to each frame respectively. Only 13 linear Mel-frequency bank filters ranging from 80 Hz to 600 Hz are used, since the dominant frequencies of the devices are in this range. After several tests, it is determined that the optimal number of cepstral coefficients as 9 for each frame. Details of the procedure were discussed in Section 4.1.1.

We faced a number of challenges in implementing a fully functional HASRS component. The.Net Micro Framework has some restrictions that need to be worked around to implement complex applications like MFCC. This includes the lack of mathematical methods such as *log*(), *sin*(), *cos*(), which led us to override the existing math library with a new library from ALGLIB [44]. While importing existing libraries to the .Net Micro Framework, we defined a set of macros to use arrays as matrices, since the.Net Micro Framework does not support matrix data structure.

Algorithm 1. Rules for time correlation

```
search for node decisions with a distance value above
   the (ε)
calculate change (Δpm) in the power usage
if (Δpm is positive)
then
     apply Rule 1
end
else
     apply Rule 2
end
```

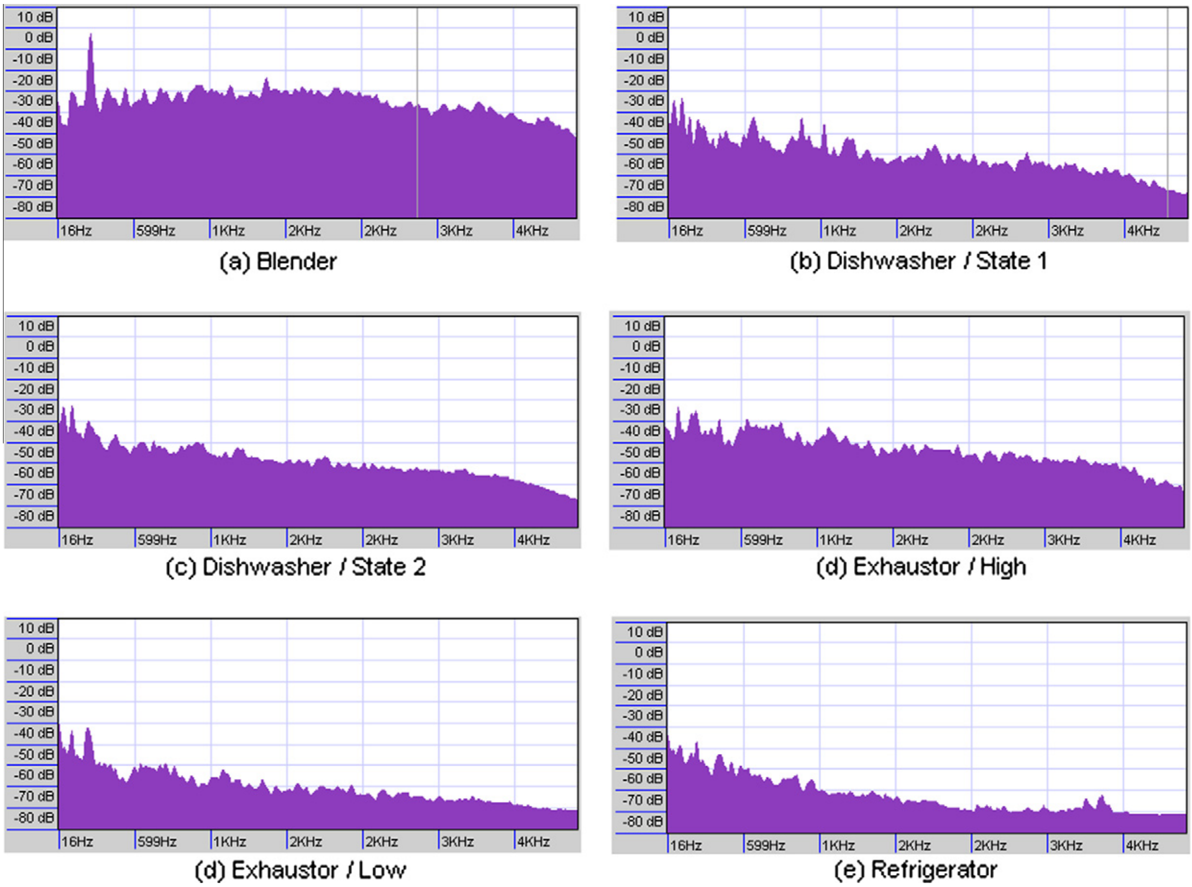


Fig. 8. Audio power spectral density of the house appliances and their states (Blender, Dishwasher/State1, Dishwasher/State2, Exhaustor/High, Exhaustor/Low, Refrigerator).

Algorithm 2. Rule 1

compare node decision (d_t) with previous node decision (d_{t-1});
if $d_t = d_{t-1}$ **then**
  find devices with power consumption (P_i) equal to the Δpm;
  if no match is found **then**
    leave d_t as it is;
else
    if only one device found **then**
      add the device to decision table (D_t);
    else
      leave D_t for further analysis;
    end
  end
end
  check the power consumption of d_t;
  if it matches the Δpm **then**
    add d_{t-1} to D_t;
  else
    leave D_t for further analysis;
  end
end

Algorithm 3. Rule 2

if D_{t-1} is silence **then**
  mark D_t as silence;
else
  check the power consumption of the device (s) in D_{t-1};
  if any of them matches Δpm **then**
    remove that device from D_t;
  else
    leave D_t for further analysis;
  end
end

5.5. Data fusion rules

When a sensor node performs classification of the sampled audio, it calculates the distance between the sample and all possible classes. Then, it classifies the sample as belonging to the class with minimum distance. In the multi-layer decision architecture of TinyEARS, the distance value is used by the TCM module. In the case where only a single device is working, the HASRS identifies the sample as belonging to one of the classes in the system with a very low distance value. The distance is higher when the sampled audio contains superimposed sounds of multiple devices. The distance value is then used by the TCM to discriminate and make decisions in situations when multiple devices are active at the same time.

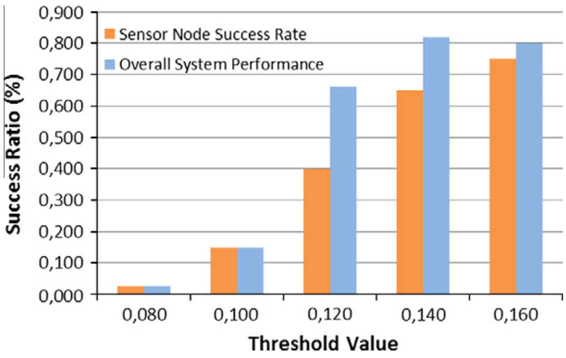


Fig. 9. Performance of audio classification and correlation with different threshold values.

If the distance value is lower than the predefined threshold, the DFC accepts the decision of the node as is. If the distance value is greater than the threshold, the DFC combines time and power consumption information and may override the node decision. This override operation is performed based on a set of predefined rules.

TCM identifies node decisions with high distance values and calculates the change in power usage Δpm. A positive change means that a device has been turned on, whereas a negative change indicates that a device has been turned off. When Δpm is positive, a list of possible devices is generated based on Δpm and known levels of power consumption of devices. Then, unfeasible options are eliminated based on previous node decisions as shown in Algorithm 2. When Δpm is negative, the power consumption of each device in the previous time slot is compared with Δpm, and any matching device is marked as "turned off" for the current time slot as shown in Algorithm 3. In both algorithms, previous node decisions and power usage might be insufficient to make a decision. In such cases, decisions can be made on further analysis based on runtime estimation and device usage characteristics. We left the analysis of such cases for future work, since they only happen on rare occasions.

Table 3 shows an example of the time correlation operation. The TCM considers node decisions with a distance

Table 3
Example of data fusion operation. Columns in italics shows with time-stamps that have distance values higher than the threshold.

Time	10:10:54	*10:14:41*	10:21:12	10:33:28	10:40:28	*10:44:02*	10:54:38
Power usage	348.9	410.4	335.5	442.1	745.6	1783.5	210.1
Distance value	0.105	0.165	0.129	0.067	0.12	0.231	0.107
Node decision	Ref.	Exh. (H)	Ref.	Ref.	Ref.	D/W (State-2)	No activity
Correlated data	Ref.	Ref & Exh. (H)	Ref.	Ref.	Ref.	Ref.	No activity
Real data	Ref.	Ref & Exh. (H)	Ref.	Ref.	Ref.	Ref.	No activity

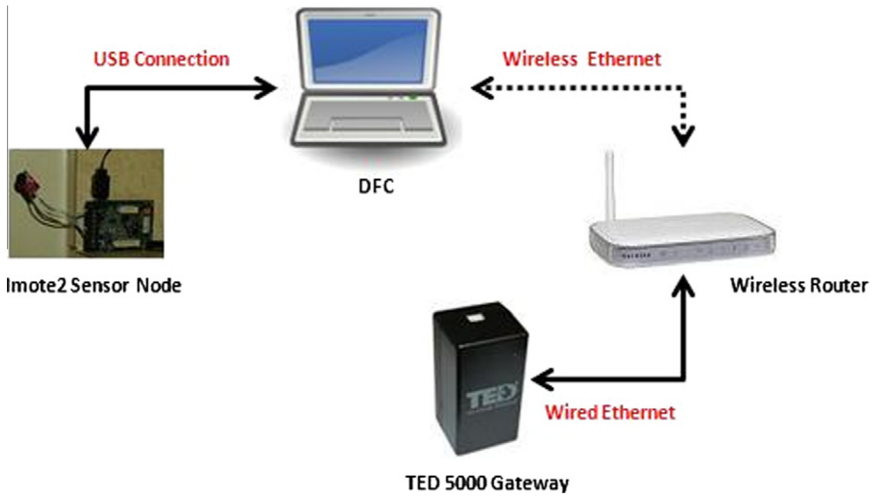


Fig. 10. Block diagram of test deployment.

value below the threshold (set to 0.140 in the experiments reported). The threshold value is determined based on experiment. We collected audio samples of house appliances in two different kitchens, and evaluated the performance of audio classification on the sensor node and correlation operation. Fig. 9 shows the success rate of audio classification on sensor node and the overall system performance after the correlation operation for different threshold values. Using higher values results poor overall system performance, since it will minimize the use of correlation operation.

6. System evaluation

Most of the house appliances do produce sound, thus can be identified by their audio signatures. There are exceptions including devices like modems, routers, lights, which do not produce any sound. Such devices can be detected by TinyEARS with the adoption of additional heterogeneous sensors. For example, a heat sensor can be used to detect an oven or HVAC, whereas a simple light sensor is enough to tell whether lights are on or off.

During the tests, sensor nodes are equipped with audio sensors only, since our primary concern in TinyEARS is to investigate the use of audio signatures for device-level energy monitoring. Devices that do not emit audio signals or have low power consumption are classified as a single unknown device.

6.1. Deployment

To test and validate our design, we have deployed TinyEARS in a two bedroom apartment in Buffalo, NY. A TED5000 power meter was connected to the electric panel, a laptop configured as DFC, and a sensor node deployed in the kitchen. The in situ training phase of TinyEARS enables a flexible deployment. Therefore, the sensor node can be placed anywhere in the room. The kitchen was selected since it is the most challenging room as multiple appli-

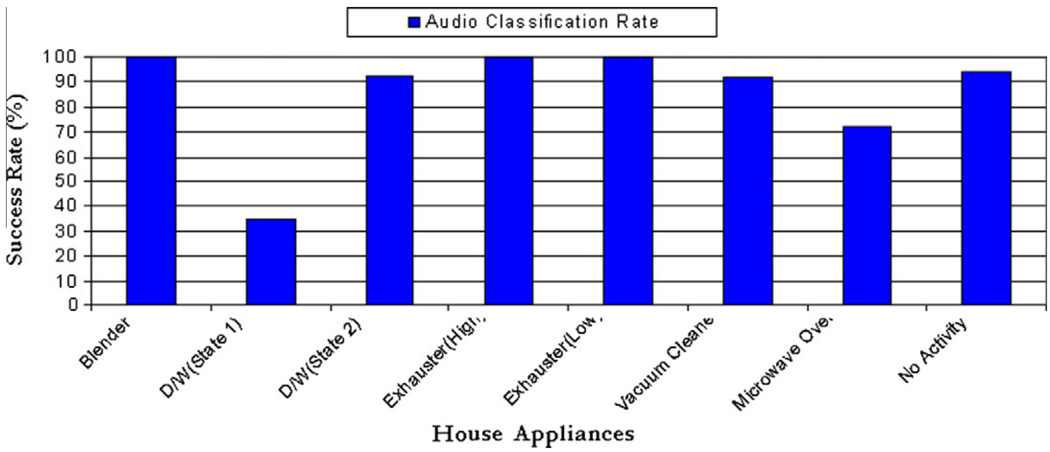


Fig. 11. Success rate of audio classification performed on the sensor node.

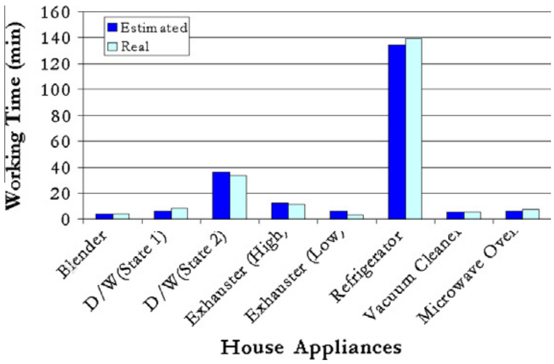


Fig. 12. True working time vs. estimated working time by DFC.

Table 4
The success rate of TinyEARS.

Device name	Power consumption		
	Real (W)	Estimated (W)	Success rate (%)
Refrigerator	357.68	370.66	96.50
D/W (State-1)	43.02	54.87	78.40
D/W (State-2)	544.2	501.6	92.17
Exh. (High)	22.45	21.64	96.39
Exh. (Low)	6.99	3.79	54
Blender	20.6	20.6	100
Vacuum cleaner	157.5	154.5	98.10
Microwave oven	145.48	183.43	79.31
Overall	1297.92	1311.09	98.99

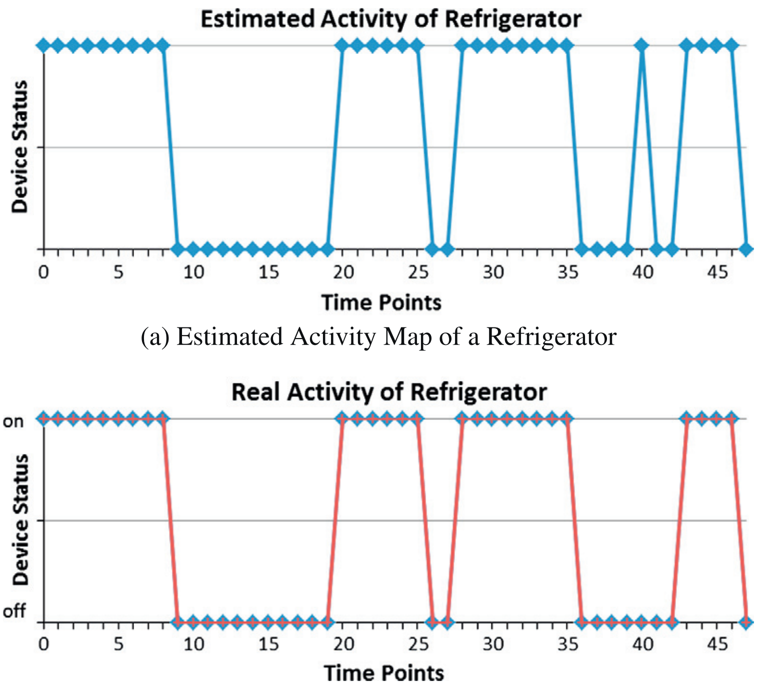
ances are often used there concurrently. A block diagram of our test deployment is shown in Fig. 10.

During the tests, occupants of the house took accurate notes of "turn on" and "turn off" times of each device in the house. A ground truth for working devices and their power consumption was constructed by combining these notes and real-time power readings gathered from power meter on every second. This ground truth is only used to evaluate the system performance during the test phase. This experimental deployment was held for three days. Audio samples from the first day were used as a training set for the system. The performance of TinyEARS was evaluated in the next two days.

6.2. Test results

To evaluate the performance of TinyEARS, we have observed six house appliances and their different run levels, i.e., *refrigerator, dishwasher (state 1 and state 2), exhauster (low and high), blender, vacuum cleaner and microewave oven*. Since TinyEARS is based on a multi-layer decision architecture, we assess the success rates of each layer individually.

The first layer of TinyEARS is the node level decision. The audio classification module was tested for 123 different activities. There might occur nine different individual activities in the kitchen, i.e., *no activity, blender, dishwasher (state 1), dishwasher (state 2), exhauster (low), exhauster (high), microwave oven, refrigerator and vacuum cleaner*. The success rate of recognition process for each appliance and its run level is shown in Fig. 11.



(a) Estimated Activity Map of a Refrigerator

(b) Real Activity Map of a Refrigerator

Fig. 13. Activity map of refrigerator.

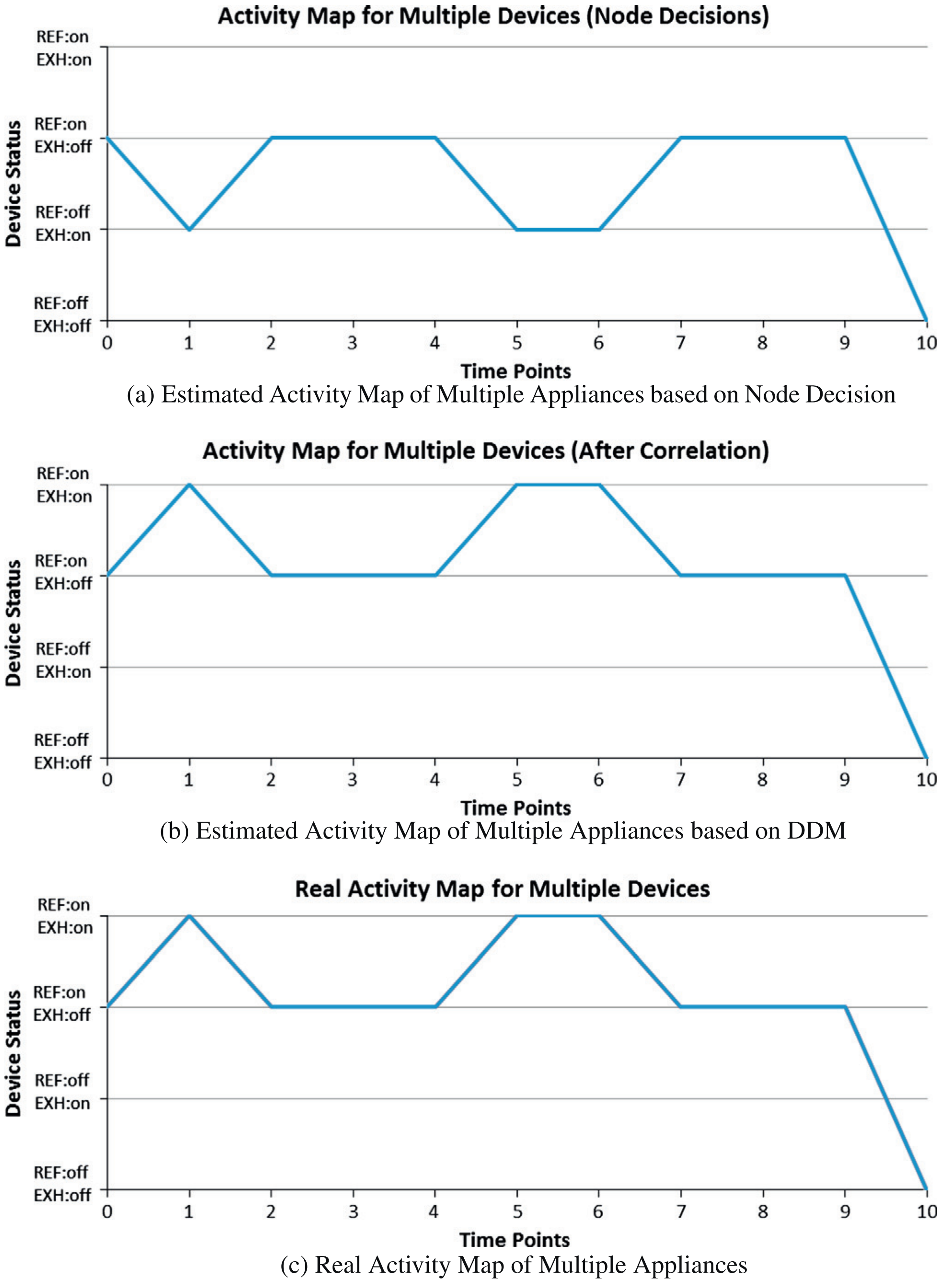


Fig. 14. Activity maps of multiple appliances.

Analyzing audio classification results shows that house appliances are clearly distinguished from each other by their acoustic signatures. However, different states of the same device, such as dishwasher (state1, state2), could be confused, since these states produce similar sounds. On the other hand, the quality of the microphone has a great impact on the success rate of the audio classification. For example, the overall success rate of the audio classification implemented with high quality microphone is 93.69%, shown in Table 2. However, the low quality microphone has caused a 8% reduction on the success rate of the audio classification in overall (Fig. 11).

At the second layer, the DFC correlates node decisions, time and power usage information to detect working house appliances. Fig. 12 show the estimated working time of the monitored house appliances based on this correlation and their real working time. Thanks to the multi-layer architecture of the TinyEARS, the low recognition rate of dishwasher (state1) has been increased from 34% up to 76%. Thus, the power consumption of individual household appliances are reported within a 10% error margin.

The overall system performance was evaluated by comparing power consumption information reported by TinyEARS to the ground truth data of two-day long activities. Table 4 shows the success rate of TinyEARS about estimating the power consumption of each house appliance.

As TinyEARS tracks running devices, we created activity maps of each house appliance to show the accuracy rate of our estimation results for on/off device status. Fig. 13 shows the real and estimated activity maps for the refrigerator. We believe that TinyEARS could exploit the activity maps of house appliances with steady working characteristics. This kind of information can be used by the TCM module especially when previous node decisions and power usage are insufficient to make a decision. In such cases, decisions can be made on further analysis based on runtime estimation and device usage characteristics. We left the analysis of such cases for future work, since they only happen on rare occasions.

Fig. 14 shows the activity map of a multiple device case where refrigerator and exhauster (in high state) are turned on and off. In this case, when the exhauster is turned on, EDM module detects the change on the power consumption and alerts the sensor nodes. Audio classifier on the sensor node recognizes the collected audio sample as exhauster with a distance value higher than the threshold. Thus, a correlation operation is performed on the node decision based on the rules defined on Section 5.5. It is clearly shown that correlation operation on the DDM increases the overall system performance remarkably.

6.3. Lifetime issues

In TinyEARS, each sensor node waits for a command from the DFC to start sampling audio signals. Then, the sensor node processes the acquired audio samples, and sends its decision back to the DFC. The nature of this operation requires periodic sleep of sensor nodes. The lifetime of TinyEARS is highly dependent on this sleep interval. Increasing the sleep period increases the network lifetime. However, it also affects the accuracy of the overall system. When the sleep interval is increased, sensors might not be able to detect appliances that are turned on for short periods of time. On the other hand, decreasing the sleep interval increases system accuracy, since it increases the chance of sensor nodes to detect appliances that are turned on for short periods of time.

7. Conclusions and future work

In this paper, we presented TinyEARS, an energy monitoring system for residential spaces using audio sensors. TinyEARS is designed to eliminate the complexity of the installation and maintenance procedures of existing power metering solutions. TinyEARS employs a multi-layer decision architecture that exploits the true capabilities of sensor nodes by processing sensor data on the node itself, thus limiting transmissions over the wireless channel. This architecture combines various sources of information to reach an accurate estimation of the consumption patterns. Experiments show that the multi-layer architecture of TinyEARS enables monitoring device-level power consumption with less than 10% error. In addition, TinyEARS can easily monitor the power consumption of multiple simultaneously active appliances as well as appliances with variable power consumption.

TinyEARS has an in situ training phase, which is performed by manually selecting audio clips of each kitchen appliance. We plan to design a simple user interface, which enables end-users to perform the training phase of the system. In our experiments, we have connected our sensor node to DFC via USB to eliminate possible problems related to the wireless communication. One of the existing reliable communication protocols can be used to provide reliable transport between the sensors and the DFC, since nodes directly communicate with DFC and transmit only their decisions and distance values.

Acknowledgement

This work was supported in part by the US National Science Foundation under Grant CNS-1117121.

References

[1] Z. Taysi, M.A. Guvensan, T. Melodia, Tinyears: spying on house appliances with audio sensor nodes, in: ACM Workshop On Embedded Sensing Systems For Energy-Efficiency in Buildings (BuildSys 2010), Zurich, Switzerland, 2010.

[2] Y. Kim, T. Schmid, M. Srivastava, Y. Wang, Challenges in resource monitoring for residential spaces, in: ACM Workshop On Embedded Sensing Systems For Energy-Efficiency in Buildings (BuildSys 2009), San Francisco, USA, 2009.

[3] S. Darby, The effectiveness of feedback on energy consumption: a review for defra of the literature on metering, billing and direct displays, 2002 <http://www.eci.ox.ac.uk/research/energy/electric-metering.php>.

[4] P.C. Stern, Information, incentives, and proenvironmental consumer behavior, Journal of Consumer Policy 22 (4) (1999) 393–509.

[5] Y. Kim, T. Schmid, Z. Charbiwala, M. Srivastava, Viridiscope: design and implementation of a fine grained power monitoring system for homes, in: ACM Proceedings of International Conference on Ubiquitous Computing (UbiComp 2009), Orlando, USA, 2009, pp. 245–254.

[6] Crossbow Imote2 Mote Specifications, 2010 <http://www.xbow.com>.

[7] Power Cost Monitor Specifications, 2010 <http://www.bluelineinnovations.com>.

[8] WATTSON Specifications, 2010 <http://www.diykyoto.com>.

[9] Onzo Smart Energy Kit Specifications, 2010 <http://onzo.com>.

[10] EM-2500 Specifications, 2010 <http://www.energymonitor.com>.

[11] The Energy Detective Specifications, 2010 <http://www.theenergydetective.com>.

[12] Google Power Meter, 2010 <http://www.google.org/powermeter/>.

[13] C. Laughman, K. Lee, R. Cox, S. Shaw, S. Leeb, L. Norford, P. Amstrong, Power signature analysis, IEEE Power and Energy Magazine 1 (2) (2003) 56–63.

[14] G. Hart, Residential energy monitoring and computerized surveillance via utility power flows, IEEE Technology and Society Magazine 8 (2) (1989) 12–16.

[15] A. Rowe, M. Berges, R. Rajkumar, Contactless sensing on appliance state transitions through variations in electromagnetic fields, in: ACM Workshop On Embedded Sensing Systems For Energy-Efficiency in Buildings (BuildSys 2010), Zurich, Switzerland, 2010.

[16] Energy Optimizers Ltd., 2010 <http://www.plogginternational.com>.

[17] Kill-a-Watt Specifications, 2010 <http://www.p3international.com>.

[18] Tweet-a-Watt Specifications, 2010 <http://www.ladyada.net>.

[19] Electronic educational devices, 2010 <https://www.wattsupmeters.com>.
[20] Smartlabs Inc., 2010 <http://www.smarthome.com>.
[21] N. Shah, C.-F. Tsai, K.-M. Chao, Monitoring appliances sensosr data in home environment: Issues and challenges, in: Proceedings of International Conference on Commerce and Enterprise Computing (CEC 2009), 2009, pp. 439–444.
[22] X. Jiang, M. Ly, J. Taneja, P. Dutta, D. Culler, Experiences with a high-fidelity wireless building energy auditing network, in: Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys 2009), San Francisco, USA, 2009.
[23] A. Schoofs, A. Guerrieri, D. Delaney, G. O'Hare, A. Ruzzelli, Annot: Automated Electricity Data Annotation Using Wireless Sensor Networks, Boston, MA, USA, 2010, pp. 1–9.
[24] A. Marchiori, Q. Han, Using circuit-level power measurements in household energy management systems, in: ACM Workshop On Embedded Sensing Systems For Energy-Efficiency in Buildings (BuildSys 2009), San Francisco, USA, 2009.
[25] H. Murata, T. Onoda, Estimation of power consumption for household electric appliances, in: Proceedings of International Conference on Neural Information Processing (ICONIP 2002), vol. 5, Singapore, 2002, pp. 2299–2303.
[26] H. Murata, T. Onoda, Applying kernel based subspace classification to a non-intrusive monitoring system for household electric appliances, in: Proceedings of International Conference on Artificial Neural Network (ICANN 2001), vol. 2130, Vienna, Austria, 2001, pp. 692–698.
[27] T. Onoda, G. RStsch, K.-R. Mnller, Applying support vector machines and boosting to a non-intrusive monitoring system for household electric appliances with inverters, in: Proceedings of International Conference on Neural Computation (NC 2000), Berlin, Germany, 2000.
[28] S. Patel, T. Robertson, J. Kientz, M. Reynolds, G. Abowd, At the flick of a switch: Detecting and classifying unique electrical events on the residential power line, in: Proceedings of International Conference on Ubiquitous Computing (UbiComp 2007), vol. 4717, Innsbruck, Austria, 2007, pp. 271–288.
[29] H. Lam, G. Fung, W. Lee, A novel method to construct taxonomy electrical appliances based on load signatures, IEEE Transactions on Consumer Electronic 53 (2) (2007) 653–660.
[30] A. Barbato, L. Borsani, A.A. Capone, S. Melzi, Home energy saving through a user profiling system based on wireless sensors, in: ACM Workshop On Embedded Sensing Systems For Energy-Efficiency in Buildings (BuildSys 2009), San Francisco, USA, 2009.
[31] D. Delaney, G. O'Hare, A. Ruzelli, Evaluation of energy efficiency in lighting systems using sensor networks, in: ACM Workshop On Embedded Sensing Systems For Energy-Efficiency in Buildings (BuildSys 2009), San Francisco, USA, 2009.
[32] G. Gao, K. Whitehouse, The self-programming thermostat: Optimizing setback schedules based on home occupancy patterns, in: ACM Workshop On Embedded Sensing Systems For Energy-Efficiency in Buildings (BuildSys 2009), San Francisco, USA, 2009.
[33] V.L. Erickson, Y. Lin, A. Kamthe, R. Brahme, A. Surana, A.E. Cerpa, M.D. Sohn, S. Narayanan, Energy efficient building environment control strategies using real-time occupancy measurements, in: ACM Workshop On Embedded Sensing Systems For Energy-Efficiency in Buildings (BuildSys 2009), San Francisco, USA, 2009.
[34] A. Marchiori, Q. Han, Distributed wireless control for building energy management, in: ACM Workshop On Embedded Sensing Systems For Energy-Efficiency in Buildings (BuildSys 2010), Zurich, Switzerland, 2010.
[35] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, D. Culler, The nesc language: A holistic approach to networked embedded systems, in: PLDI '03: Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation, ACM, San Diego, California, USA, 2003, pp. 1–11, doi:http://doi.acm.org/10.1145/781131.781133.
[36] M. Cowling, Non-Speech Environmental Sound Classification System for Autonomous Surveillance, Ph.D. Thesis, Faculty of Engineering and Information Technology, Griffith University, 2004.
[37] S. Chu, S. Narayanan, C.-C. Kuo, Environmental sound recognition with time-frequency audio features, IEEE Transactions on Audio, Speech, and Language Processing 17 (6) (2009) 1142–1158.
[38] M.A. Guvensan, Z.C. Taysi, T. Melodia, Environmental sound classification for house appliances, in: Proceedings of the 18th IEEE Conference on Signal Processing and Communication Applications, Diyarbakir, Turkey, 2010.
[39] R. Vergin, D. O'Shaughnessy, V. Gupta, Compensated mel frequency cepstrum coefficients, in: Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1996), Minneapolis, USA, 1996, pp. 323–326.
[40] J.-C. Wang, Jhing-Fa, Y.-S. Weng, Chip design of MFCC extraction for speech recognition, Integration, VLSI Journal 32 (1–3) (2002) 111–131.
[41] E. Alpaydin, Introduction to Machine Learning, MIT Press, 2004.
[42] Crossbow Imote2, ITS400 Sensor Board Specifications, 2010 <http://www.xbow.com>.
[43] MD9745APZ-F Electret Microphone Datasheet, 2010 <http://www.knowles.com>.
[44] ALGLIB Specifications, 2010 <http://www.alglib.net/>.

M. Amac Guvensan received his B.Sc., M.Sc., and Ph.D. degrees in computer engineering from the Yildiz Technical University, Turkey, in 2002, 2006, and 2011 respectively. He is a research assistant at the Department of Computer Engineering, YTU. His current research interests include coverage optimization in directional sensor networks, deployment strategies and signal processing algorithms in wireless multimedia sensor networks. Mr. Guvensan visited the Wireless Networks and Embedded Systems Laboratory, University at Buffalo, The State University of New York, for 6 months within 2009–2010.

Z. Cihan Taysi received his B.S. and M.S. in Department of Computer Engineering in Yildiz Technical University in Turkey in 2002 and 2006, respectively. He is pursuing his Ph.D. in Intelligent Systems Laboratory (ISL) under Asistant Professor Gokhan YAVUZ in Yildiz Technical University in Turkey. Mr. Taysi visited the Wireless Networks and Embedded Systems Laboratory Department of Electrical Engineering, State University of New York at Buffalo under Prof.Tommaso Melodia, for 6 months within 2009–2010. His current research interests include Inter-Vehicle Communications, mobile and wireless communication, and signal processing in wireless multimedia sensor networks.

Tommaso Melodia is an Associate Professor with the Department of Electrical Engineering at the State University of New York (SUNY) at Buffalo, where he directs the Wireless Networks and Embedded Systems Laboratory. He received his Ph.D. in Electrical and Computer Engineering from the Georgia Institute of Technology in 2007. He had previously received his "Laurea" (integrated B.S. and M.S.) and Doctorate degrees in Telecommunications Engineering from the University of Rome "La Sapienza", Rome, Italy, in 2001 and 2005, respectively. He is a recipient of the National Science Foundation CAREER award, and coauthored a paper that was recognized as the Fast Breaking Paper in the field of Computer Science for February 2009 by Thomson ISI Essential Science Indicators and a paper that received an Elsevier Top Cited Paper Award. He is the Technical Program Committee Vice Chair for IEEE Globecom 2013 and the Technical Program Committee Vice Chair for Information Systems for IEEE INFOCOM 2013, and serves in the Editorial Boards of IEEE Transactions on Mobile Computing, IEEE Transactions on Wireless Communications, and Computer Networks (Elsevier), among others. His current research interests are in modeling, optimization, and experimental evaluation of wireless networks, with applications to cognitive and cooperative networking, ultrasonic intra-body area networks, multimedia sensor networks, and underwater networks.