CapSense: Capacitor-based Activity Sensing for Kinetic Energy Harvesting Powered Wearable Devices

Guohao Lan
UNSW & Data61-CSIRO
Sydney, Australia
guohao.lan@unsw.edu.au

Dong Ma
UNSW & Data61-CSIRO
Sydney, Australia
dong.ma1@student.unsw.edu.au

Weitao Xu
UNSW
Sydney, Australia
w.t.xu@unsw.edu.au

Mahbub Hassan
UNSW & Data61-CSIRO
Sydney, Australia
mahbub.hassan@unsw.edu.au

Wen Hu
UNSW & Data61-CSIRO
Sydney, Australia
wen.hu@unsw.edu.au

ABSTRACT

We propose a new activity sensing method, CapSense, which detects activities of daily living (ADL) by sampling the voltage of the kinetic energy harvesting (KEH) capacitor at an ultra low sampling rate. Unlike conventional sensors that generate only instantaneous motion information of the subject, KEH capacitors accumulate and store human generated energy over time. Given that humans produce kinetic energy at distinct rates for different ADL, the KEH capacitor can be sampled only once in a while to observe the energy generation rate and identify the current activity. Thus, with CapSense, it is possible to avoid collecting time series motion data at high frequency, which promises significant power saving for the sensing device. We prototype a shoe-mounted KEH-powered wearable device and conduct experiments with 10 subjects for detecting 5 different activities. Our results show that compared to the existing time-series-based activity recognition, CapSense reduces sampling-induced power consumption by 99% and the overall system power, after considering wireless transmissions, by 75%. CapSense recognizes activities with up to 90%.

CCS CONCEPTS

•Computing methodologies →Activity recognition and understanding; •Human-centered computing →Ubiquitous computing;

KEYWORDS

Energy-efficiency, Activity Recognition, Wearable Device

ACM Reference format:
Guohao Lan, Dong Ma, Weitao Xu, Mahbub Hassan, and Wen Hu. 2017. CapSense: Capacitor-based Activity Sensing for Kinetic Energy Harvesting Powered Wearable Devices. In *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, Melbourne, VIC, Australia, November 7–10, 2017 (MobiQuitous 2017),* 11 pages.
DOI: 10.1145/3144457.3144459

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MobiQuitous 2017, Melbourne, VIC, Australia
© 2017 ACM. 978-1-4503-5368-7/17/11…$15.00
DOI: 10.1145/3144457.3144459

1 INTRODUCTION

The rapid development of embedded technology has enabled wearable systems [32] that provide autonomous health, fitness, and wellness monitoring services, such as step-counting [7, 27] and recognition of ADL [15, 16]. To improve user-experience, wearable devices are restricted to very small form factor, limiting the size of battery that can be embedded in the device. However, most users desire 24/7 monitoring, which leads to long-term continuous sampling of motion sensors, such as accelerometers, and frequent data transmission to the server for further processing. The high power consumption due to frequent sensor sampling and wireless transmission limits the battery life of these wearable devices.

The first stage in a typical activity recognition system is the acquisition of a time-series of samples from the sensors [5]. In general, the power consumption of sensor sampling is directly proportional to the sampling rate, as the higher the sampling rate, the more power is consumed by the sensors as well as the microcontroller (MCU), which has to wake up more frequently to read, process, and store the samples. A large volume of past research on human activity sensing, therefore, have focused on reducing the sampling rates of accelerometer-based systems. Depending on the type of activities, reported sampling rates for accelerometer-based activity sensing range between 25Hz to 100Hz for recognition accuracies over 85% [5, 8, 25].

To further reduce power consumption of sensor sampling, researchers have recently proposed the use of KEH transducer, which produces different AC voltage patterns for different activities, as an alternative to accelerometer [18, 35]. By saving the energy that would have been otherwise consumed by the accelerometer, transducer-based systems can reduce the sampling power consumption to some extent [18]. However, as we will demonstrate later in the paper, the sensor itself consumes a small fraction of the system power during sampling; a large fraction of the sampling power is actually consumed by the MCU. The transducer-based approach, which also relies on time series data, therefore still consumes significant amount of limited battery power in a small form-factor wearable device.

Based on this observation, we propose a new way to exploit the KEH circuit for detecting human activities. The new method, which we call CapSense, avoids the acquisition of time series data of instantaneous motion (for accelerometer) or AC voltage (for transducer) information, thereby drastically reducing the sensor sampling rate. To cut down the sampling rate, CapSense capitalizes two important facts in KEH-powered systems:

- First, instead of providing only *instantaneous information* (i.e., the instant acceleration or AC voltage), the KEH capacitor can provide *accumulated information.* That is, the energy generated continuously by the KEH transducer for a given period of T second is accumulated in the capacitor. Therefore, a capacitor can directly yield the energy generation rate during the last T second with a single sample of its current voltage. In essence, the capacitor works as a *free computation engine* that *computes* the current energy generation rate without involving the MCU or collecting any time series data.
- Second, it has been demonstrated that the energy generation rates of human activities are distinctively different [9, 39]. Thus, it should be possible to classify human activities by sampling the capacitor voltage only once in every T second. For example, for a choice of $T = 5$, we can achieve activity recognition with a sampling rate of only 0.2 Hz.

The novelty and contributions of this paper can be summarized as follows:

- We propose a new method for human context sensing, CapSense, to drastically reduce sampling rates and power consumption of KEH-powered wearable systems. CapSense detects activity from the voltage of the KEH-capacitor, which to the best of our knowledge has not been explored before.
- We have implemented the idea of CapSense in shoe form factor using piezoelectric bending energy harvester. We derive design parameters for the capacitor that produce effective voltage samples for human activity recognition.
- We conduct experiments with 10 subjects for detecting 5 different activities. We demonstrate that CapSense is capable of detecting ADL with up to 92% accuracy.
- We conduct a detailed power profiling to quantify the power saving opportunity of CapSense. Our measurement results indicate that, compared to the state-of-the-art, CapSense reduces sampling-related power consumption by 99% and the overall system power by 75%.

The rest of the paper is organized as follows. We review the related works in Section 2. Then, we present the design and implementation of CapSense in Section 3, followed by its performance evaluation in Section 4. The power measurement study is presented in Section 5. We conclude our work in Section 6.

2 RELATED WORK

In this section, we first review existing works on reducing the sampling frequency. Then, we discuss some recent efforts in utilizing KEH-transducer as an energy-efficient motion sensor.

2.1 Reducing Sampling Frequency

A large volume of works in the literature focused on reducing the sampling rates [5, 8, 29, 36] to improve energy efficiency in sensing. In [20] Krause et al., studied the trade-off between power consumption and classification accuracy for the e-Watch wearable device. They demonstrated that the lifetime of the device can be extended by selecting the optimal sampling strategy without accuracy losing. Similar results are presented in [36], the authors pointed out that there is a trade-off between sampling frequency and classification accuracy, and introduced the A3R algorithm which adapts the sampling frequency and classification features in real-time based on the activity type. In addition, researchers have also proposed the use of compressive sensing theory to reduce sampling frequency [6, 26, 28] by leveraging the temporal-sparsity of human activity.

2.2 KEH-transducer based Context Sensing

Recent efforts in the literature start to apply KEH transducer as a low power vibration sensor. In [18], Khalifa et al. proposed the idea of using the power signal generated by a VEH device for human activities recognition. The proposed system can achieve 83% of accuracy for classifying different daily activities. In [23], the authors investigate the feasibility of using KEH as a motion sensor for transportation mode detection. In [22], Lan et al. demonstrated the use of KEH voltage signal to estimate calorie expenditure of human activities. In [14], a piezoelectric energy harvester-based wearable necklace has been design for food-intake monitoring. The proposed system achieves over 80% of accuracy in distinguishing food categories. In [2], Blank et al. proposed a ball impact localization system using a piezoelectric embedded table tennis racket. More recently, Xu et al. [35] proposed an authentication system which utilizes the AC voltage signal to authenticate the user based on gait analysis. The proposed system can achieve an recognition accuracy of 95% when five gait cycles are used. In [24], the authors proposed the use of KEH-transducer as an energy-efficient receiver for acoustic communication. Although the use of KEH-transducer can save the energy consumption in powering the motion sensor, it still continuously sense and process the power signal from the KEH transducer at a high sampling rate, and thus, will continue to face the energy consumption problem.

3 SYSTEM OVERVIEW

3.1 CapSense Architecture

Although there are a various of designs in kinetic energy harvesting powered wearable devices, such as backpack [34], fabric [37], wristband [31], and footwear [21, 33], we designed our system in the form-factor of shoes for several reasons: first, shoes are worn by users for the majority of time in their daily lives, including working/studying in the office/school; second, unlike many other wearable devices that their form-factors have been shrunk dramatically, shoes have a much larger space to place an energy harvester; third, it has been widely reported that a desirable amount of energy can be harvested from human daily activities using the insole-based harvesting system [11, 12, 33].

Figure 1 presents the system architecture of CapSense which consists of two parts: the *Energy Harvesting* and *Load*. The *Energy Harvesting* corresponds to the functional components that harvest and accumulate energy from human activity. It includes a piezoelectric energy harvesting (PEH) transducer to generate electric power (i.e., AC voltage) from foot strikes, and an rectifying circuit

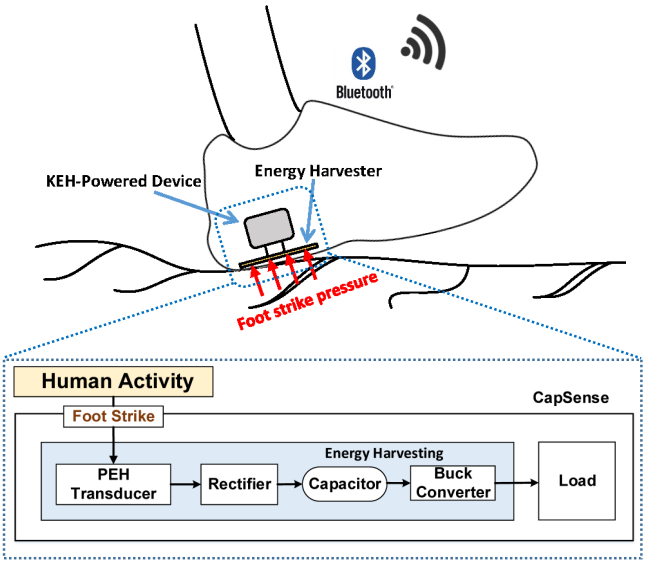


Figure 1: System architecture of CapSense.

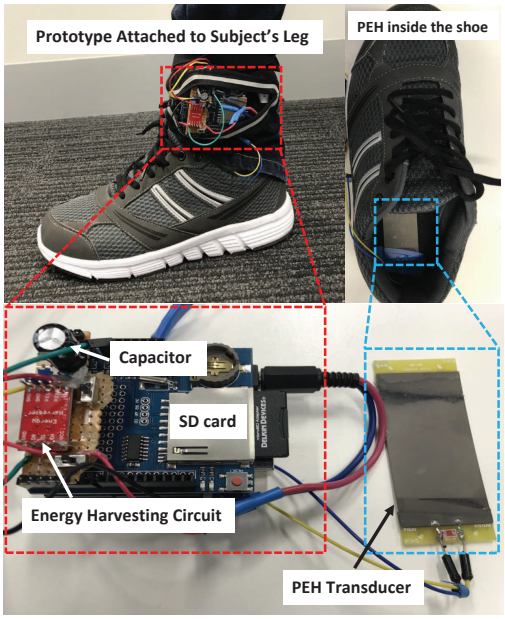


Figure 2: Pictures of CapSense prototype.

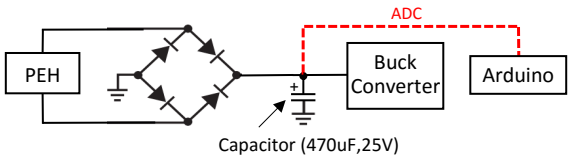


Figure 3: Circuit diagram of energy harvester.

that is able to rectify the intermittent or continuous AC voltage output from the PEH transducer into stable DC power. Then, the rectified DC voltage will be accumulated in a capacitor before it is sufficient to turn-on the buck converter and power any electronics (i.e., the voltage of the capacitor should reach a pre-defined threshold configured in the buck converter). The *Load* represents any system components responsible for data sensing, processing, and communication, or could be a rechargeable battery that can be used to power a wearable system.

As an illustration, considering the scenario in which a subject is wearing the KEH-powered shoes and doing some activities, e.g., walking or running, her heel will hit the ground floor and the foot strike induced pressure will bend the energy harvester inside the shoe. Consequently, the energy harvester will generate electric power from the foot strikes when the subject is doing different activities, and the generated energy are naturally accumulated in the capacitor [10, 11, 38]. In CapSense, the voltage of the capacitor is sampled periodically by the system load (e.g., once every T seconds), and is send to the remote server (i.e., smartphone or cloud-server) for further processing. As the power generated from different human activities such as walking, running, and relaxing, are distinctively different [9, 39], and the energy generated by the PEH transducer within the last T second(s) is accumulated in the capacitor, it would be possible to estimate the power generation rate during T by **a single sample of the capacitor voltage**. Then, the estimated rates can be used to recognize the activity performed by the user in the last T second with a sampling rate of $\frac{1}{T}$. As we will demonstrate later in Section 4, CapSense can detect activities with sampling rates as low as 0.14Hz compared to tens of Hz required by the state-of-the-art. Thus, CapSense can reduce sensing-induced power consumption of wearable devices by several orders.

The fundamental novelty of CapSense is that, unlike accelerometer or KEH transducer that can only generate ***instantaneous motion information*** of the subject at a particular time, capacitor stores the generated KEH energy over time, and thus, it provides ***accumulated information*** that can be used to detect the subject's activity within a period of time. Therefore, unlike accelerometer-based [5] or KEH transducer-based [18, 35] sensing systems that require **a time-series of signal** sampled from the sensor (i.e., a sequence of signal sampled from the accelerometer or KEH transducer), CapSense utilizes **a single sample** of the capacitor voltage for activity recognition.

3.2 CapSense Prototype

In this subsection, we present the design and implementation of our prototype. Figure 2 gives the pictures of our prototype which we implemented in the form of shoe. As discussed previously, our prototype consists of two parts, the *Energy Harvesting* and *Load*. For the Energy Harvesting part, we select the EH220-A4-503YB PEH bending transducer from Piezo Systems[1] as our PEH transducer and attached it to the shoe-pad. The PEH transducer is only 10.4 grams weighted with a dimension of 76.2×31.75×2.28 mm^3, which is easy to be placed inside the shoe. The output pins of the PEH transducer are connected to an energy harvesting circuit,

[1]Piezo System: http://www.piezo.com/prodexg8dqm.html.

namely the LTC3588-1 from the Linear Technology[2]. The LTC3588-1 integrates a low power-loss bridge rectifier that can be used to rectify the AC voltage output from the PEH transducer, and a high efficiency buck converter that is able to transfer the energy stored in the capacitor into stable DC power to power/charge the load. We select an electrolytic capacitor with a capacitance of 470μF and a rating voltage (i.e., the maximum voltage) of 25V to store the generated energy (we will discuss how we select the capacitor in Section 3.3). When the voltage of the capacitor rises above the undervoltage lockout rising threshold of the buck converter (i.e., denoted by V_{UVLO} *rising*, and equals to $4V$ in our setting), the buck converter will be enabled to discharge the energy stored in the capacitor. On the other hand, when the capacitor voltage has been discharged below the lockout falling threshold (i.e., denoted by V_{UVLO} *falling*, and equals to $3.08V$ in our setting), the buck converter will be turned off, and the capacitor starts to be charged again. In our current prototype, an Arduino Uno is used for data logging. The voltage of the capacitor is sampled by the Arduino through onboard 10-bit ADC at 100Hz and stored on the SD card for offline data analysis. The circuit diagram of the prototype is shown in Figure 3.

3.3 Ensuring Linearity in Capacitor Voltage

Before presenting the details of capacitor-based sensing, we first analyze some properties of the capacitor when the system is powered by an energy harvester, and discuss the feasibility and design requirement of leveraging the capacitor voltage for activity sensing. For this purpose, we use our CapSense hardware to collect some voltage samples from a subject when she is conducting different daily activities, namely, walking, running, ascending/descending stairs, and stationary.

The voltage of the capacitor, $V_C(t)$, at time t during the charging is given by:

$$V_C(t) = V_{max}(1 - e^{-t/\tau}), \tag{1}$$

in which, V_{max} is the maximum voltage to which the capacitor can be charged, and it is bounded by $V_{max} = min\{V_{rating}, V_S\}$, where V_S is the voltage applied to the capacitor (i.e., the rectified DC voltage from the rectifier in our case), and V_{rating} is the rating voltage of the capacitor; and τ is defined as $\tau = RC$, in which, R is the resistance of the resistor in the equivalent resistor-capacitor charging circuit (RC circuit), and C is the capacitance of the capacitor. For a given capacitor, τ is known as the *time constant* of the equivalent RC circuit, which is a constant value (in second).

Figure 4 plots the voltage of a capacitor when it is charged over time (in the unit of RC). The first observation is that, within the examining time of $5RC$, the increasing rate of capacitor voltage is not constant. For instance, the voltage increment in the first RC interval (i.e., from time 0 to RC) is not equal to that increased in the second RC interval (i.e., from time RC to $2RC$). As the only information we can obtain from the capacitor is its voltage, and we are using the voltage increasing rate for activity sensing, the non-linear increment in the capacitor voltage will introduce additional uncertainties in the activity recognition.

[2]Linear Technology LTC3588: http://www.linear.com/product/LTC3588-1.

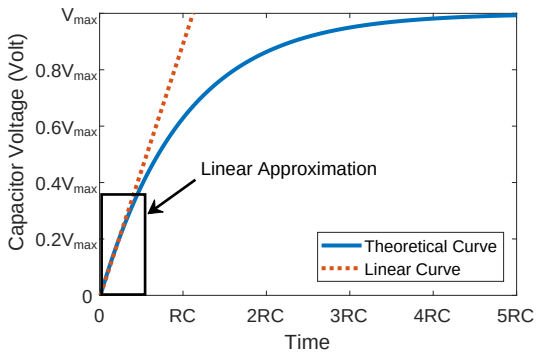


Figure 4: The theoretical curve shows the voltage of a capacitor when it is charged over time. And the theoretical curve is approximated by an linear curve. The unit of time is RC (i.e., the time constant).

Fortunately, as we can observe in Figure 4, with time $t \leq \frac{1}{2}RC$, the theoretical curve of V_C, defined in Equation (2), can be approximated by a linear curve. According to the RC circuit theory, a capacitor can be charged to 39.3% of V_{max} with a charging time of $\frac{1}{2}RC$, this means that to ensure the linearity in the capacitor voltage for a maximum time of $\frac{1}{2}RC$, V_{max} should satisfy:

$$V_{max} \geq \frac{V_{UVLO\ rising}}{0.393}, \tag{2}$$

which yields:

$$min\{V_{rating}, V_S\} \geq \frac{V_{UVLO\ rising}}{0.393}, \tag{3}$$

in which, V_{UVLO} *rising* is the undervoltage lockout rising threshold of the buck converter, at which the capacitor starts to be discharged. This means that, to ensure the linearity in the capacitor voltage, the selection and configuration of the hardware components (i.e., the PEH transducer, the buck converter, and capacitor) should be considered interactively.

As discussed in Section 3.2, in our prototype, V_{UVLO} *rising* of the buck converter has been configured to 4V[3]. This means that, given Equation (3), we have: $min\{V_{rating}, V_S\} \geq \frac{4V}{0.393} = 10.18V$. The rectified DC voltage from the rectifier, V_S, depends on the energy harvester that is used in the system. Given different materials and configurations of the energy harvester, V_S could be as high as tens of volts. In our case, the rectified voltage from the PEH transducer we used in the current prototype is up to 20.8V which is much higher than 10.18V. Therefore, it turns out that the rating voltage of the capacitor, V_{rating}, should be larger than 10.18V. We select a capacitor with rating voltage of 25V to meet the requirement.

3.4 Activity Sensing using Capacitor Voltage

In this section, we discuss how does CapSense leverage the capacitor voltage for activity sensing.

An actual voltage trace showing the capacitor charging and discharging status is presented in Figure 5. Initially, the capacitor

[3]According to the datasheet of LTC3588, to ensure an output DC voltage of 2.5V, the lowest voltage threshold is 4V.

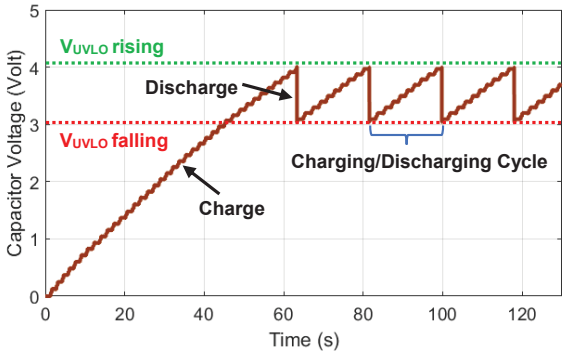


Figure 5: Voltage trace showing the capacitor is charged and discharged periodically when the subject is running.

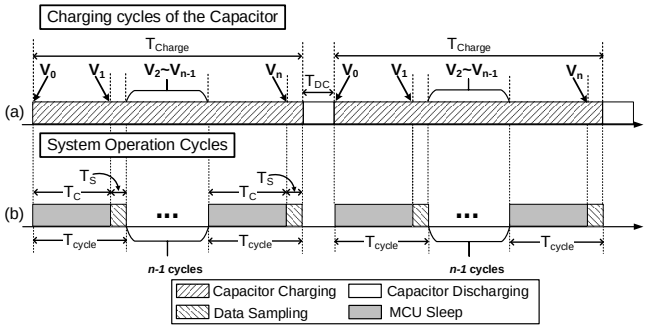


Figure 6: Timeline (a) is the charging cycles of the capacitor. Timeline (b) is the system operation cycles of CapSense.

voltage starts from 0, and takes approximately 60 seconds to reach 4V and triggers the buck converter to discharge the accumulated energy (i.e., the V_{UVLO} of the buck converter is 4V). Then, when the capacitor voltage is discharged to 3.08V, the buck converter is shut off until the capacitor voltage being charged above the V_{UVLO} rising threshold again. As shown, the capacitor is charged and discharged periodically under the control of the energy harvesting circuit depending on its voltage level. The length of each charging cycle is approximately 20 seconds when the subject is running (i.e., the most intense activity we consider). The cycle length will be bigger if the subject is doing some modest activities such as walkings. This means that there is at least 20 seconds of time, during which the voltage samples could be used for activity sensing.

Figure 6 presents the system operation cycles of CapSense, together with the charging cycles of the capacitor. As shown, within each capacitor charging cycle, T_{Charge}, CapSense duty-cycles the system MCU to sample capacitor voltage periodically. At the beginning of each charging cycle, MCU is in the sleep mode, while the capacitor starts to accumulate the energy generated by the KEH transducer. After an accumulation time of T_C, MCU awakes and starts to sample the voltage V_C of the capacitor. It takes time T_S for MCU to complete the data sampling (according to our measurement in Section 5, T_S equals to 0.6ms). Once the sampling is finished, MCU goes back to sleep again. As shown in Figure 6, in each capacitor charging cycle, CapSense reads a series of capacitor voltage,

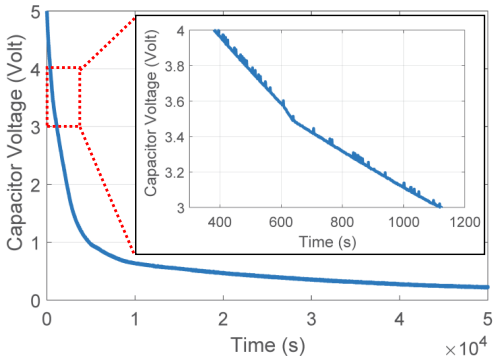


Figure 7: The measured voltage of a self-discharge capacitor.

$V_1, V_2, ..., V_{n-1}, V_n$. By using two adjacent voltage samples, for instance, V_{k-1} and V_k, it is straightforward to estimate the increase rate, r, of the capacitor voltage within the last accumulation time of T_C by:

$$r = \frac{V_k - V_{k-1}}{T_C}. \quad (4)$$

Therefore, using the voltage reading of the capacitor, we can estimate the energy generation rate of the transducer in time T_C, without the need of high frequency data sampling. It is also noteworthy that, as the MCU has no knowledge about the charging/discharge status of the capacitor, it is possible that those two adjacent voltage samples are obtained in two different charging cycles (i.e., it may result in $V_{k-1} \geq V_k$, as V_k is sampled at the initial charging state of the capacitor in a new charging cycle). In this case, when calculate the increase rate, r, we disregard all adjacent voltage peer, $\{V_{k-1}, V_k\}$, with $V_k \leq V_{k-1}$.

However, the estimation of r may be affected by the discharging time of the capacitor (i.e., T_{DC} shown in Figure 6). As if it takes a long time for the buck converter to discharge the capacitor from 4V to 3V, it is possible that the MCU may wake-up and sample an incorrect voltage value during the discharge of the capacitor, which will result in a wrong estimation in r. Fortunately, as shown in Figure 5, the time required to discharge the capacitor from 4V to 3V, T_{DC}, is extremely short which is less than ten ms in our current design. Thus, it is highly unlikely that the voltage is sampled incorrectly when the capacitor is discharging. Another factor that may affect the estimation is the capacitor leakage. Figure 7 shows the measured voltage of our capacitor when it self-discharges (i.e., leakage) from 5V to 0V. Specifically, we are interested in the behavior of the capacitor within the 3-4V range. As specified in the subfigure, we can observe that it takes more than 700 seconds (i.e., 12 minutes) for the capacitor to self-charge from 4V to 3V, which means the leakage of the capacitor is negligible within a short time of a few seconds. Thus, the capacitor leakage will not affect our estimation with a few seconds accumulation time T_C.

As an example, Figure 8 plots the voltage samples of the capacitor when a subject is doing different activities. We can observe that, for all five activities, our hardware design ensures a linear increasing in the voltage when the capacitor is powered by the energy harvester. As different human activities can generate distinctive amount of energy [9, 39], intuitively, the increasing rate of capacitor voltage

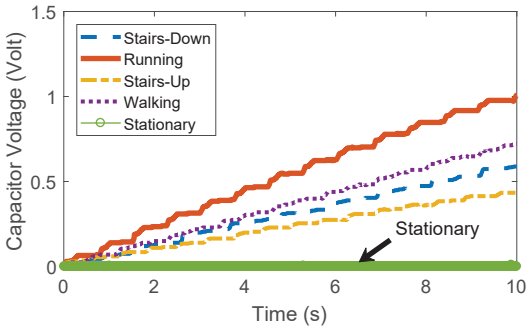


Figure 8: Voltage traces of different activities.

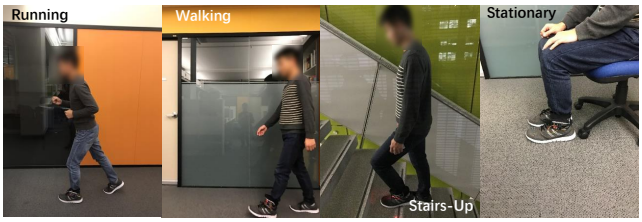
would directly yield the energy harvesting rate during the last few seconds, thus, we can achieve activity recognition by simply using the capacitor voltage. CapSense utilizes solely the capacitor voltage to classify different human activities using conventional machine learning algorithms, which enables system level power saving by enabling the MCU to stay in the energy-saving low-power mode for extended periods of time. In the following section, we will evaluate the performance of CapSense using our prototype.

4 SYSTEM EVALUATION

4.1 Data Collection

The dataset we used to evaluate the proposed system is collected from 10 healthy subjects volunteered in our lab[4]. The subjects are diverse in gender (8 males and 2 females), age (range from 24 to 30), weight (from 55 to 75Kg), and height (from 168 to 183cm). We considered five different activities, including: walking, running, ascending/descending stairs, and stationary (i.e., sitting or standing). During data collection, the subjects were asked to wear our energy-harvesting embedded shoe (we prepared shoes with different sizes to meet the requirement of our subjects) and the prototype system is attached to the subject's ankle (as shown in Figure 2). Then, the subjects were asked to perform the activities normally in their own way without any specific instruction. As illustrated in Figure 9, for activities such as running and walking, they are performed in both indoor and outdoor environments to capture the influence of different terrains. For ascending and descending stairs, we conducted data collection in two building environments with different styles of stairs. For all the five activities, each volunteer participated at least two data collection sessions for both indoor and outdoor environments. For walking, running, and stationary, each session lasts at least 20 seconds, whereas, for ascending/descending stairs (i.e., the slope and steps of the stairs are different), each session may last 7 to 10 seconds depending on the number of steps and the speed of the subject. For each of the five activities, we have collected at least four sessions of samples from each of the 10 subjects. In total, we have 210 sessions of data.

[4]Ethical approval for carrying out this experiment has been granted by the corresponding organization (Approval Number HC15888).



(a) Subject doing different activities.

(b) Indoor environment.

(c) Outdoor environment.

Figure 9: The illustration of data collection.

4.2 Evaluation Methodology

The evaluation is carried out in WEKA[5] using 10-folds cross validation with 10 repetitions for each test. Six typical machine learning algorithms are used: the C4.5 decision tree algorithm (C4.5) [30], IBk K-Nearest Neighbor classifier [1], Naive Bayes with kernel estimation [13], RandomForest [4], DecisionTable [19], and BayesNets [3]. The parameters of the classifiers are optimized using the CVParameterSelection algorithm [19]. We evaluate CapSense performance in terms of activity recognition accuracy (i.e., True Positive Rate). The experiment is conducted in a subject-dependent manner. It is noteworthy that, unlike conventional accelerometer-based [5] or KEH transducer-based activity sensing system [18] that require complex signal processing and feature extractions, CapSense relies only on the voltage difference of the capacitor for activity recognition.

4.3 Performance Evaluation

4.3.1 Recognition Accuracy v.s. Subject. The first thing we are interested in is the impact of subject difference on the recognition accuracy, as subjects will perform activity in different ways which may affect the system accuracy [5]. Intuitively, given the diversity in subject's gender, weight, and height, the foot strike pressures applied on the energy harvester differ in the way subjects perform the activity. The classifier we used in this experiment is Naive Bayes.

Figure 10 exhibits the achieved accuracy of CapSense for all the 10 subjects given different accumulation windows T_C. As expected, given a specific classifier and a fixed T_C, the achieved classification varies with subjects. More specifically, Figure 11 plots the confusion matrix of the classification results for two subjects, Subject 4 and 8, with T_C=3s. Subject 4 achieves the lowest accuracy among all subjects, whereas, Subject 8 achieves the highest. As shown in Figure 11, for both subjects, the major classification error occur between the class 'Stairs' (i.e., ascending/descending stairs) and

[5]WEKA: http://www.cs.waikato.ac.nz/ml/weka/.

Subject	1s	2s	3s	4s	5s	6s	7s
S1	64.54	68.82	83.85	87.13	88.54	89.86	93.17
S2	65.03	76.28	82.47	83.48	84.71	86.16	87.58
S3	82.18	89.01	93.71	96.78	97.44	99.80	99.84
S4	67.82	74.78	76.82	78.57	78.88	78.90	79.46
S5	83.28	86.99	86.65	88.65	88.61	89.53	89.52
S6	85.97	90.13	93.88	93.98	94.66	96.03	96.00
S7	81.20	82.85	85.79	87.85	89.95	91.33	91.47
S8	84.93	95.36	96.79	97.98	98.71	99.31	99.94
S9	71.26	82.94	85.71	85.56	87.79	89.05	89.86
S10	72.77	86.74	92.71	95.10	96.64	99.23	99.93
AVG	75.90	83.39	87.84	89.51	90.59	91.92	92.68

T_C (s)

Figure 10: CapSense accuracy (in %) with Naive Bayes classifier for all the ten subjects given different accumulation window T_C. (results with accuracy ≥ 85% are highlighted)

	Stairs	Walk	Run	Sit
Stairs	0.47	0.34	0.19	0.00
Walk	0.23	0.73	0.04	0.00
Run	0.13	0.00	0.87	0.00
Sit	0.00	0.00	0.00	1.00

(a) Subject 4, $T_C = 3s$.

	Stairs	Walk	Run	Sit
Stairs	0.92	0.08	0.00	0.00
Walk	0.05	0.95	0.00	0.00
Run	0.00	0.00	1.00	0.00
Sit	0.00	0.00	0.00	1.00

(b) Subject 8, $T_C = 3s$.

Figure 11: Confusion matrix of CapSense with Naive Bayes classifier for Subject 4 and 8. The $T_C = 3s$.

'Walk', due to the high similarity between those activities. For Subject 4, there are more classification errors appear between those two classes than that for Subject 8. We can also notice that about 13% of the 'Run' instances have been classified incorrectly as 'Stairs' for Subject 4, whereas, there is no classification error for Subject 8. Clearly, as expected, the results indicate that different subjects perform the activities in different ways, and those differences have impacted the CapSense performance. Fortunately, CapSense can still achieve high classification accuracy for 90% of the subjects based on our evaluation.

4.3.2 Recognition Accuracy v.s. Accumulation Time. Now, we investigate the impact of accumulation time T_C on the recognition accuracy. First, as we have shown in Figure 10, for a specific classifier (i.e., Naive Bayes), the achieved accuracy increases with T_C. Similarly, Figure 12 exhibits the averaged accuracy achieved by the six classifier given different T_C for all the subjects. We can see that the accuracy increases with T_C irrespective of the classifier. The second observation is that, 9 out of 10 subjects can achieve over 85% of accuracy with $T_C \geq 6$s. Intuitively, this is because a larger accumulation window can lead to a more distinctive difference in the capacitor voltage. Thus, a large T_C is preferable in improving the classification accuracy. However, the sojourn time for a subject in performing a specific activity is short, and transitions between activities may occur in the middle of T_C. Therefore, T_C should not be set too large that exceeds the activity sojourn time. For instance, during our data collection, we have noticed that, for ascending/descending a 10-steps stair, the sojourn time is usually

Subject	1s	2s	3s	4s	5s	6s	7s
S1	65.00	69.86	84.85	86.63	90.06	90.74	93.32
S2	64.73	76.00	81.66	83.36	84.28	86.61	87.18
S3	82.02	88.77	94.03	97.04	97.45	99.97	99.96
S4	68.45	72.71	75.02	78.47	78.63	77.94	80.85
S5	84.95	87.36	87.85	88.94	88.70	88.85	88.73
S6	85.80	89.75	93.83	93.95	94.74	96.10	95.84
S7	80.67	83.00	85.06	88.20	89.48	90.90	91.92
S8	84.55	94.95	96.10	97.77	98.77	99.62	99.96
S9	71.00	83.22	85.58	86.18	87.42	90.57	89.59
S10	72.97	87.67	92.53	94.79	96.44	99.20	99.91
AVG	76.02	83.33	87.65	89.53	90.50	91.95	92.72

T_C (s)

Figure 12: The accuracy (in %) achieved by CapSense given different subjects and accumulation window T_C. The results are the averaged accuracy achieved by the seven different classifiers.

Classifier	1s	2s	3s	4s	5s	6s	7s
NaiveBayes	75.89	83.39	87.84	89.51	90.59	91.92	92.68
C4.5	76.12	83.45	87.75	89.47	90.53	92.22	92.81
RandomForest	76.11	83.48	87.71	89.47	90.75	92.04	92.97
IBK	76.04	83.49	87.67	89.54	90.72	92.05	92.93
DecisionTable	75.97	83.17	87.32	89.63	90.27	91.91	92.69
BayesNets	75.96	83.05	87.62	89.55	90.12	91.57	92.29
AVG	76.02	83.33	87.65	89.51	90.65	91.95	92.72

T_C (s)

Figure 13: The accuracy (in %) achieved by CapSense with different classifiers given different accumulation window T_C. The results are the averaged accuracy across the ten subjects.

within 8 to 10 seconds depending on the subject's speed. Thus, the maximum value of T_C is configured to 7 seconds based on our current dataset. Moreover, as reported in [36], for activities such as walking, running, and standing/sitting, the sojourn time is at least 1 to 2 minutes. For activities such as ascending/descending stairs, the sojourn time is much shorter, but still longer than 5 seconds for over 99.9% of the time. Therefore, as a trade-off between the system performance and robustness, we recommend the maximum value of T_C for CapSense to be configured to 5 seconds.

4.3.3 Recognition Accuracy v.s. Classifier. Lastly, we analyze the performance of CapSense with different classifiers. Figure 13 exhibits the averaged performance of CapSense across all the ten subjects with different T_C when apply different classifiers. We can observe that, all the six classifiers can achieve over 85% of accuracy with $T_C \geq 3s$. With $T_C \geq 5s$, the average accuracy of CapSense across the 10 subjects will increase to 90% for all the classifiers. With $T_C = 7s$, CapSense is able to achieve over 92% of accuracy which is comparable to the performance achieved by conventional motion sensor-based systems [25].

An interesting observation is that CapSense exhibits no bias on the selection of classifier, as all the six classifiers achieved similar

classification results. In general, the selection of relevant features and the decision on how to construct a classification model on the selected attributes can have an enormous impact on the classification result. Thus, for conventional accelerometer-based or KEH transducer-based activity sensing systems [5, 18, 35], they require very large feature set and complex feature selection algorithm to build the learning model carefully. Differently, as CapSense does not require very complex feature sets and relies only on the voltage difference for activity recognition, it shows no bias on the selection of classifier. Therefore, it simplifies the procedure in building the classification model and minimizes the errors that could result from selecting an inappropriate classifier.

5 POWER CONSUMPTION ANALYSIS

Battery lifetime is the major roadblock for the mainstream uptake of wearable technology[6], and system energy consumption optimization is an active research area for both academics and industries. Existing efforts [18, 23, 35] have demonstrated the superiority of KEH transducer-based sensing system in energy saving. Comparing to accelerometer-based system, KEH transducer-based system can save the energy consumption in power energy-hungry motion sensor. In this section, we will conduct an extensive power consumption profiling of off-the-shelf wearable activity recognition systems. We will demonstrate that, for KEH transducer-based system, a non-negligible amount of system energy consumption still comes from powering the MCU for sampling the AC voltage signal. As KEH transducer-based activity recognition system still requires tens of Hz sampling frequency, it turns out that the MCU needs to wake-up and polling the signal from the transducer frequently. In contrast, by appropriately use the properties of the capacitor, CapSense enables system level power saving by enabling the MCU to stay in the energy-saving low-power mode for extended periods of time comparing to the state-of-the-art KEH transducer-based system. Consequently, we will show that CapSense can reduce the system power consumption by several factors.

5.1 Measurement Setup

We use an off-the-shelf Texas Instrument SensorTag[7] as the target device, which is embedded with the ultra-low power ARM Cortex-M3 MCU that is specifically designed for today's energy-efficient wearable devices[8]. The SensorTag is running the Contiki 3.0 operating system[9] which duty-cycles the MCU to save energy. To further reduce the power consumption of MCU in processing, a lower clock frequency of 12MHz is used instead of the default 48MHz. Moreover, all unnecessary components, including the onboard ADC, SPI bus, and the on board accelerometers are powered-off when it is not sampling. The SensorTag is connected with a 10Ω resistor in series and powered by a 3V coin battery. The average power consumption and time requirement for each sampling event are measured by using the built-in function in the Agilent DSO3202A oscilloscope. We are interested in the power consumption of SensorTag in data sampling (i.e., either in sampling the capacitor voltage or the AC voltage signal from KEH transducer), and the power consumption in data transmission. In the following, we present our measurement results of those parts in order.

[6]http://www.theverge.com/2016/1/15/10775660/fitness-tracker-smartwatch-battery-problems-apple-motorola-lumo (accessed on July 29th 2016).
[7]SensorTag: http://www.ti.com/ww/en/wireless_connectivity/sensortag/.
[8]Mainstream wearable devices such as FitBit are using ARM Cortex-M3: see https://www.ifixit.com/Teardown/Fitbit+Flex+Teardown/16050.
[9]Contiki OS: http://www.contiki-os.org/.

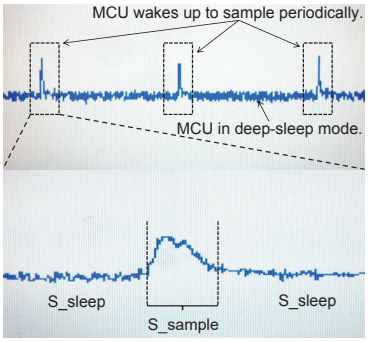


Figure 14: Profiling of voltage sampling.

Table 1: States of MCU in sampling the ADC signal.

State	Time (ms)	Power (μW)	Description
S_{sample}	0.6	480	MCU wakes up to sample ADC signal.
S_{sleep}	null	6	MCU in deep-sleep mode.

Table 2: The power consumption (μW) in data sampling.

	KEH transducer-25Hz	CapSense-0.14Hz
Sensing	7.11	0.04
MCU Leakage	6	6
Overall	13.11	6.04

5.2 Power Consumption in Sampling

First, we investigate the power consumption in data sampling. In the measurement, both capacitor voltage and KEH transducer signal are simpled through the on board ADC of SensorTag. The sampling frequency of MCU is configured as 25Hz to meet the requirement of KEH transducer-based sensing system. Figure 14 presents an oscilloscope trace when MCU sampling the signal from ADC periodically. As shown, MCU is triggered by the timer to sample periodically. It takes approximately $0.6ms$ for the MCU to complete a single voltage sampling event (i.e., state S_{sample} shown in Figure 14). After that, MCU turns back into the deep sleep mode (i.e., LPM3 in Contiki OS) to save power. The average power consumption of the system for a single ADC sampling is 480μW, and the baseline system power consumption when MCU is in the deep-sleep-mode is only 6μW. The details of power consumption and time duration for MCU sampling the ADC signal are summarized in Table 1.

In general, for the duty-cycled activity sensing system, the average power consumption in data sampling, P_{sense}, can be obtained by the following equation:

$$P_{sense} = \begin{cases} \frac{T_S \times n}{1000} P_{sample} + (1 - \frac{T_S \times n}{1000}) P_{sleep} & \text{if } 0 \le n \le \frac{1000}{T_S}, \\ P_{sample} & \text{if } \frac{1000}{T_S} < n. \end{cases} \quad (5)$$

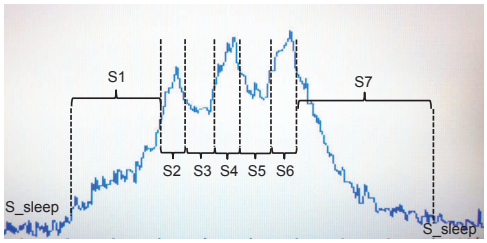


Figure 15: Profiling of BLE broadcasting event.

where, P_{sample} is the average power consumption of the system during the sampling event, and P_{sleep} is the average power consumption when the MCU is in deep-sleep mode (with all the other system components power-off). n is the sampling frequency, and T_S is the duration of time (in milli-second) required by a single sampling event. Based on the measurement results given in Table 1, we can have the average power consumption for KEH voltage sampling event, P_{sample}, equals to 480μW with a duration, T_S, equals to 0.6ms. The power consumption when MCU in deep-sleep mode, P_{sleep}, is 6μW. For KEH transducer-based system, given different application scenarios, a sampling frequency of 25Hz-50Hz is required to achieve good accuracy for human activity recognition [17, 18, 23, 35]. Therefore, given the minimum required sampling frequency of 25Hz, following Equation 5 we can obtain the power consumption in data sampling for KEH transducer-based system equals to 13.11μW. On the other hand, as demonstrated in Section 4, to achieve an overall classification accuracy of 90%, CapSense need only to sample the ADC signal once every 7s. Following Equation 5, the power consumption in data sampling for CapSense is only 6.04μW.

The results are summarized in Table 2. As shown, **CapSense is able to save 54% of the overall power consumed by the transducer-based system in data sampling**. It is noteworthy that, in case of Sensing, the power consumption of CapSense is only 0.6% of the KEH transducer-based system (i.e., $\frac{0.04\mu W}{7.11\mu W}$). We can also notice that for CapSense, the main energy expenditure is the MCU Leakage (i.e., the power consumption of the system when MCU is in deep-sleep mode), which consumes 99% (6μW over 6.04μW) of the overall sampling power consumption. Fortunately, with the rapid development of energy-efficient micro-controllers, we can expect the power consumption in System Leakage could be further reduced. For instance, the EFM32 Gecko MCU[10] consumes only 2.7μW in the deep-sleep mode, it can help CapSense achieving an ultra-low system power consumption of 2.71μW.

5.3 Power Consumption for Data Transmission

In this subsection, we study the power consumption for the wireless transmission using the BLE beacons in broadcasting mode. We programmed the Contiki OS which wake-ups the CC2650 wireless MCU periodically to transmit a BLE beacon packet, which is repeated three times on three separate channels (repetition improves reliability of broadcasting). The transmitted beacon packets are all 19 Bytes (all for protocol payloads. The details for each BLE broadcasting event are visualized in Figure 15, and summarized in Table 3. Note that, the transmission time is depending on the packet size. The time stated in Table 3 (0.28ms for state S2, S4, and S6) is the minimum required time to transmit the 19 Bytes packet per channel. For every additional Byte[11] to be transmitted, 8μs time needs to be added to the total transmission time.

Table 3: States of BLE broadcasting event.

State	Time (ms)	Power (uW)	Description
S1	1.12	1008	Radio setup.
S2, S4, S6	0.28	3990	Radio transmits a beacon packet of 19 Bytes.
S3, S5	0.30	2460	Transition between transmissions.
S7	1.72	744	Post-processing before sleep.
S_sleep	null	6	Radio off; MCU in deep-sleep mode.

Table 4: Summary of data transmission power consumption.

	KEH-25Hz	CapSense-0.14Hz
Power	3.212mW	1.716mW
Time	33.9ms	4.3ms
Energy	108.88μJ	7.43μJ

For transducer-based system with a sampling frequency of 25Hz, it has $25Hz \times 7s = 175$ voltage samples (2 Bytes for each 12-Bits ADC reading, and 350 Bytes in total) to be transmitted per Channel for every seven seconds. Given the maximum additional data can be added to each beacon packet is 28 Bytes, this requires $\lceil \frac{350}{28} \rceil = 13$ packets to be transmitted per channel. As a result, for transducer-based system, it consumes 108.88μJ[12] to transmit the 13 packets on three different channels. The average power consumption is 3.212mW with time duration of 33.9ms. On the other hand, for CapSense, it has only one voltage sample to be transmitted for every seven seconds (in total, 2 Bytes), results in one packet to be transmitted per channel. The total energy consumption for CapSense is only 7.43μJ[13]. The average power consumption is 1.716mW with time duration of 4.3ms. The results are summarized in Table 4. This means that, CapSense is able to **save over 93% of the energy consumption in data transmission**. Clearly, for KEH transducer-based systems, the radio has to stay for a longer period of time to transmit more sampling data. Although, different transmission mechanisms (data aggregation and feature selection) can be applied to reduce the amount of data to be transmitted [25], and thus, reduce the transmission power consumption. However, additional on-board computations in those mechanisms may still introduce inevitable power consumption. Combining the power consumption in data sampling and transmission together, the overall system power consumption for KEH transducer-based system is 28.53μW[14], whereas, the overall system power consumption for

[10] EFM32 Gecko 32-bit MCU: http://www.silabs.com/products/mcu/32-bit/efm32-gecko.

[11] According the protocol, up to 28 Bytes of data could be added to the 19 Bytes payloads per packet.

[12] Obtained by: $1.12\mu s \times 1.008\mu W + 1.72\mu s \times 0.744\mu W + 39 \times (0.28\mu s + 0.008\mu s \times 28) \times 3.99\mu W + 38 \times 0.3\mu s \times 2.46\mu W = 108.88\mu J$.

[13] Obtained by: $1.12\mu s \times 1.008\mu W + 1.72\mu s \times 0.744\mu W + 3 \times (0.28\mu s + 0.008\mu s \times 2) \times 3.99\mu W + 2 \times 0.3\mu s \times 2.46\mu W = 7.43\mu J$.

[14] Obtained by: $(\frac{13.11\mu W \times 7sec + 3.212mW \times 33.9ms}{7sec + 33.9ms}) = 28.53\mu W$.

CapSense is only 7.09μW[15]. This means that **CapSense is able to save 75% of the overall system power consumption of state-of-the-art KEH transducer-based system**.

6 CONCLUSION AND FUTURE DIRECTIONS

In this paper, we present CapSense, a novel activity sensing scheme for KEH-powered wearable devices. By simply using the voltage readings of the capacitor which is sampled at a frequency down to 0.2Hz, CapSense is able to achieve over 90% of accuracy in classifying different daily activities, and reduce sampling-related power consumption of the state-of-the-art system by 99% and the overall system power consumption by 75%.

The current work is a first step in capacitor-based human activity recognition. As such, it can be extended in many directions. First of all, we have only evaluated its potential under the ideal condition when activities do not overlap within the sampling interval. In the future work, we will investigate solutions that address the more practical cases where activity changes occur at arbitrary time boundaries. Second, as the current hardware prototype is quite cumbersome, another direction for future work is to design the prototype with a smaller form-factor, and provide detailed user study on the practical user experience of this device. Finally, as our hardware can harvest energy from different user activities, we will investigate ways to utilize the harvested energy to power our system, thus making it battery-free.

REFERENCES

[1] David W Aha, Dennis Kibler, and Marc K Albert. 1991. Instance-based learning algorithms. *Machine learning* 6, 1 (1991), 37–66.

[2] Peter Blank, Thomas Kautz, and Bjoern M Eskofier. 2016. Ball impact localization on table tennis rackets using piezo-electric sensors. In *Proceedings of ISWC*. ACM, 72–79.

[3] Remco R Bouckaert. 2004. *Bayesian network classifiers in weka*. Department of Computer Science, University of Waikato Hamilton.

[4] Leo Breiman. 2001. Random Forests. *Machine Learning* 45, 1 (2001), 5–32.

[5] Andreas Bulling, Ulf Blanke, and Bernt Schiele. 2014. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)* 46, 3 (2014), 33.

[6] Chun Tung Chou, Rajib Rana, and Wen Hu. 2009. Energy efficient information collection in wireless sensor networks using adaptive compressive sensing. In *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*. IEEE, 443–450.

[7] Sunny Consolvo, David W McDonald, Tammy Toscos, Mike Y Chen, Jon Froehlich, Beverly Harrison, Predrag Klasnja, Anthony LaMarca, Louis LeGrand, Ryan Libby, and others. 2008. Activity sensing in the wild: a field trial of ubifit garden. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1797–1806.

[8] Hassan Ghasemzadeh, Navid Amini, Ramyar Saeedi, and Majid Sarrafzadeh. 2015. Power-aware computing in wearable sensor networks: An optimal feature selection. *IEEE Transactions on Mobile Computing* 14, 4 (2015), 800–812.

[9] Maria Gorlatova, John Sarik, Guy Grebla, Mina Cong, Ioannis Kymissis, and Gil Zussman. 2014. Movers and shakers: Kinetic energy harvesting for the internet of things. In *Proceedings of SIGMETRICS*, Vol. 42. ACM, 407–419.

[10] Jeremy Gummeson, Shane S Clark, Kevin Fu, and Deepak Ganesan. 2010. On the limits of effective hybrid micro-energy harvesting on mobile CRFID sensors. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 195–208.

[11] Qianyi Huang, Yan Mei, Wei Wang, and Qian Zhang. 2015. Battery-free Sensing Platform for Wearable Devices: The Synergy Between Two Feet. In *Proceedings of INFOCOM*.

[12] Koichi Ishida, Tsung-Ching Huang, Kentaro Honda, Yasuhiro Shinozuka, Hiroshi Fuketa, Tomoyuki Yokota, Ute Zschieschang, Hagen Klauk, Gregory Tortissier, Tsuyoshi Sekitani, and others. 2013. Insole pedometer with piezoelectric energy harvester and 2 V organic circuits. *IEEE Journal of Solid-State Circuits* 48, 1 (2013), 255–264.

[15] Obtained by:$(\frac{6.04\mu W \times 7sec + 1.716mW \times 4.3ms}{7sec + 4.3ms}) = 7.09\mu$W.

[13] George H John and Pat Langley. 1995. Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 338–345.

[14] Haik Kalantarian, Nabil Alshurafa, Tuan Le, and Majid Sarrafzadeh. 2015. Monitoring eating habits using a piezoelectric sensor-based necklace. *Computers in biology and medicine* 58 (2015), 46–55.

[15] Seungwoo Kang, Jinwon Lee, Hyukjae Jang, Hyonik Lee, Youngki Lee, Souneil Park, Taiwoo Park, and Junehwa Song. 2008. Seemon: scalable and energy-efficient context monitoring framework for sensor-rich mobile environments. In *Proc. of MobiSys*. ACM, 267–280.

[16] Matthew Keally, Gang Zhou, Guoliang Xing, Jianxin Wu, and Andrew Pyles. 2011. Pbn: towards practical activity recognition using smartphone-based body sensor networks. In *Proc. of SenSys*. ACM, 246–259.

[17] Sara Khalifa, Mahbub Hassan, and Aruna Seneviratne. 2015. Pervasive self-powered human activity recognition without the accelerometer. In *Proceedings of PerCom*. IEEE, 79–86.

[18] Sara Khalifa, Mahbub Hassan, Aruna Seneviratne, and Sajal K Das. 2015. Energy-harvesting wearables for activity-aware services. *IEEE Internet Computing* 19, 5 (2015), 8–16.

[19] Ron Kohavi. 1995. The Power of Decision Tables. In *8th European Conference on Machine Learning*. Springer, 174–189.

[20] Andreas Krause, Matthias Ihmig, Edward Rankin, Derek Leong, Smriti Gupta, Daniel Siewiorek, Asim Smailagic, Michael Deisher, and Uttam Sengupta. 2005. Trading off prediction accuracy and power consumption for context-aware wearable computing. In *Proc. of ISWC*. IEEE, 20–26.

[21] John Kymissis, Clyde Kendall, Joseph Paradiso, and Neil Gershenfeld. 1998. Parasitic power harvesting in shoes. In *Proceedings of ISWC*. IEEE, 132–139.

[22] Guohao Lan, Sara Khalifa, Mahbub Hassan, and Wen Hu. 2015. Estimating calorie expenditure from output voltage of piezoelectric energy harvester: an experimental feasibility study. In *Proceedings of EAI BodyNets*. 179–185.

[23] Guohao Lan, Weitao Xu, Sara Khalifa, Mahbub Hassan, and Wen Hu. 2016. Transportation mode detection using kinetic energy harvesting wearables. In *Proceedings of PerCom (WiP)*. IEEE, 1–4.

[24] Guohao Lan, Weitao Xu, Sara Khalifa, Mahbub Hassan, and Wen Hu. 2017. VEH-COM: Demodulating vibration energy harvesting for short range communication. In *Proceedings of PerCom*. IEEE, 170–179.

[25] O.D. Lara and M.A. Labrador. 2013. A Survey on Human Activity Recognition using Wearable Sensors. *IEEE Communications on Surveys and Tutorials* 15, 3 (2013), 1192–1209.

[26] Shancang Li, Li Da Xu, and Xinheng Wang. 2013. Compressed sensing signal and data acquisition in wireless sensor networks and internet of things. *IEEE Transactions on Industrial Informatics* 9, 4 (2013), 2177–2186.

[27] Thomas Olutoyin Oshin and Stefan Poslad. 2013. ERSP: An energy-efficient real-time smartphone pedometer. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*. IEEE, 2067–2072.

[28] Liu Qi, James Williamson, Wyatt Mohrman, Kun Li, Qin Lv, Robert Dick, and Li Shang. 2016. Gazelle: Energy-Efficient Wearable Analysis for Running. *IEEE Transactions on Mobile Computing* (2016).

[29] Xin Qi, Matthew Keally, Gang Zhou, Yantao Li, and Zhen Ren. 2013. AdaSense: Adapting sampling rates for activity recognition in Body Sensor Networks. In *Proc. of RTAS*. IEEE, 163–172.

[30] J Ross Quinlan. 2014. *C4. 5: programs for machine learning*. Elsevier.

[31] SEIKO. 2016. SEIKO Kinetic Watch. (2016). Retrieved June 16, 2017 from https://www.seikowatches.com/world/technology/kinetic/

[32] Suranga Seneviratne, Yining Hu, Tham Nguyen, Guohao Lan, Sara Khalifa, Kanchana Thilakarathna, Mahbub Hassan, and Aruna Seneviratne. 2017. A Survey of Wearable Devices and Challenges. *IEEE Communications Surveys & Tutorials* (2017).

[33] Nathan S Shenck and Joseph A Paradiso. 2001. Energy scavenging with shoe-mounted piezoelectrics. *IEEE micro* 21, 3 (2001), 30–42.

[34] Longhan Xie and Mingjing Cai. 2014. Human motion: sustainable power for wearable electronics. *IEEE Pervasive Computing* 13, 4 (2014), 42–49.

[35] Weitao Xu, Guohao Lan, Qi Lin, Sara Khalifa, Neil Bergmann, Mahbub Hassan, and Hu Wen. 2017. KEH-Gait: Towards a Mobile Healthcare User Authentication System by Kinetic Energy Harvesting. In *Proceedings of NDSS*.

[36] Zhixian Yan, Vigneshwaran Subbaraju, Dipanjan Chakraborty, Archan Misra, and Karl Aberer. 2012. Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach. In *Proc. of ISWC*. IEEE, 17–24.

[37] Boram Yang and Kwang-Seok Yun. 2012. Piezoelectric shell structures as wearable energy harvesters for effective power generation at low-frequency movement. *Sensors and Actuators A: Physical* 188 (2012), 427–433.

[38] Yong Yang, Lili Wang, Dong Kun Noh, Hieu Khac Le, and Tarek F Abdelzaher. 2009. Solarstore: enhancing data reliability in solar-powered storage-centric sensor networks. In *Proc. of MobiSys*. ACM, 333–346.

[39] Jaeseok Yun, Shwetak Patel, Matt Reynolds, and Gregory Abowd. 2008. A quantitative investigation of inertial power harvesting for human-powered devices. In *Proc. of UbiComp*. ACM, 74–83.