REAP: Runtime Energy-Accuracy Optimization for Energy Harvesting IoT Devices

Ganapati Bhat[1], Kunal Bagewadi[1], Hyung Gyu Lee[2] and Umit Y. Ogras[1]
[1]Arizona State University, Tempe, AZ [2] Daegu University, Daegu, Korea

ABSTRACT

The use of wearable and mobile devices for health and activity monitoring is growing rapidly. These devices need to maximize their accuracy and active time under a tight energy budget imposed by battery and form-factor constraints. This paper considers energy harvesting devices that run on a limited energy budget to recognize user activities over a given period. We propose a technique to co-optimize the accuracy and active time by utilizing multiple design points with different energy-accuracy trade-offs. The proposed technique switches between these design points at runtime to maximize a generalized objective function under tight harvested energy budget constraints. We evaluate our approach experimentally using a custom hardware prototype and 14 user studies. It achieves 46% higher expected accuracy and 66% longer active time compared to the highest performance design point.

1 INTRODUCTION

Wearable low-power internet-of-things (IoT) devices enable health monitoring, activity tracking, and wide area sensing applications [1–3]. These devices must stay on for as long as possible to analyze user activities. At the same time, they have to provide the maximum quality of service. These two objectives compete with each other since higher accuracy comes at the cost of larger energy consumption. Since weight and form-factor constraints prohibit large batteries, the feasibility of these devices depends critically on optimizing the energy-accuracy trade-off optimally at runtime.

Widely used dynamic power management techniques optimize the power-performance trade-off by switching between different power states. High-performance states are used to execute computationally heavy workloads at the expense of power consumption, while low-performance states are used to save power. In analogy, energy-accuracy trade-off in self-powered devices can be optimized by utilizing multiple design points. This is a challenging task since characterizing the accuracy analytically is much harder than developing power consumption and performance models. For example, we consider an activity recognition application where a wearable device infers the user activities, such as jogging, by processing motion sensor data. The recognition accuracy is a strong function of the users. Hence, energy-accuracy optimization requires *user studies* and *optimally chosen design points*, in addition to a *runtime optimization algorithm* that utilizes multiple design points.

DAC '19, June 2–6, 2019, Las Vegas, NV, USA

ACM ISBN 978-1-4503-6725-7/19/06…$15.00
https://doi.org/10.1145/3316781.3317892

This paper presents a Runtime Energy-Accuracy oPtimization (*REAP*) framework for energy-constrained IoT devices. While our framework is general, we focus on health and activity monitoring applications where a wearable device processes sensor inputs to infer user activities. The recognized activities are sent to a gateway, such as a smartphone, for further processing. *REAP* co-optimizes the accuracy and active time under a tight energy budget. This optimization is enabled by the following three contributions.

User studies for accuracy evaluation: We perform experiments with 14 users to recognize the following activities: *sit*, *stand*, *walk*, *jump*, *lie down* and *transitions* among them. During these experiments, we collect 3-axis accelerometer and stretch sensor data. We obtain a total of 3553 activity windows from these experiments. After labeling, we utilize this data for evaluating the accuracy of the human activity classifiers used in this work.

Pareto-optimal design points: A common baseline in activity monitoring applications is to obtain a classifier with the highest recognition accuracy [4]. High accuracy is obtained by using a sophisticated set of sensors, features, and classification algorithms, all of which imply a larger energy consumption, hence, lower active time. Other design points can be obtained by reducing the number of sensors and feature set to save energy. In turn, the energy savings lead to longer active time under a given harvested energy budget, albeit with lower accuracy. *To enable this work, we implemented 24 design points (DPs) with varying energy-accuracy trade-offs* on our hardware prototype. Among them, we choose 5 Pareto-optimal DPs as our primary designs used at runtime. We provide detailed execution time and power consumption breakdown for sensing, feature generation and processing steps for each of these 5 DPs.

Runtime optimization algorithm: Given an energy budget, two fundamental objectives are to maximize the recognition accuracy and the amount of time the device is on, i.e., the active time. We first formulate this co-optimization problem assuming that there are N design points with different energy-accuracy trade-offs. We define a general objective function that enables us to tune the importance of active time versus recognition accuracy. Then, we propose an efficient runtime algorithm that determines how much each DP should be used so as to optimize the accuracy-active time trade-off.

Experimental results using a custom prototype based on TI Sensortag [5] IoT board show that *REAP* outperforms all static design points under a range of energy budget constraints. *REAP* achieves both 46% higher expected accuracy and 66% longer active time compared to the highest performance DP. *REAP* also achieves comparable active time to the lowest energy design points while providing significantly higher expected accuracy. This makes *REAP* suitable for use in a wide range of energy harvesting profiles.

The major contributions of this paper are as follows:

- A runtime technique to co-optimize the accuracy and active time of energy-harvesting IoT devices.

- Pareto-optimal design points with varying energy-accuracy trade-offs for human activity recognition (HAR).
- Experiments on a custom prototype with 14 user studies that show significant improvements both in expected accuracy and active time compared to static design points.

2 RELATED WORK

Energy harvesting for IoT devices has received significant attention due to their small form factor and low capacity batteries [6, 7]. These devices can be broadly categorized into two classes. The first class of devices rely solely on harvested energy and turn off when no energy is harvested [8]. The second class of devices uses a small battery as a backup to extend the active time [9–11]. These approaches manage the power consumption of the device such that the total energy consumed over a finite horizon is equal to the harvested energy [12]. This ensures a long device lifetime without battery replacement or manual charging. *REAP* is applicable to all devices that operate under a fixed energy budget.

Using ambient energy sources necessitates the development of energy allocation and duty cycling algorithms [10, 11, 13]. For example, the work in [9] uses linear programming to determine the duty cycle of the application as a function of the harvested energy. Similarly, the algorithm in [10] uses a linear quadratic controller to assign the duty cycle of the device while maintaining a set battery level. There are also algorithms for dynamic energy management of energy-harvesting devices for long-term energy-neutral operation [11, 12]. However, these approaches choose between on and off power states leading to sub-optimal operation. They also lack the notion of accuracy or any concrete application unlike this work.

Human activity and health monitoring using wearable devices have significant impact potential to sports, patients with movement disorders and the elderly [2, 14, 15]. A recent work presents a wearable system for mobile analysis of running using motion sensors [1]. The authors selectively identify the best sampling points to maintain high accuracy while reducing sensing and analysis energy overheads. The work in [14] presents a framework to detect falls by using a wearable device equipped with accelerometers. Authors in [3] design a classifier that detects physical activity using a body-worn accelerometer. It offers an accurate classifier for human activity recognition, but it cannot sustain operation under tight energy budget constraints. Based on this observation, we find Pareto-optimal design points for the HAR application that offer varying levels of accuracy and energy consumption. Then, we use these design points to maximize the expected accuracy of HAR.

In summary, we present a unique combination of (1) a runtime energy-accuracy optimization technique, and (2) experimental evaluation with 5 concrete design points for HAR. We released the experimental data to public to stimulate research in this area [16].

3 RUNTIME ENERGY-ACCURACY OPTIMIZATION

3.1 Preliminaries

We consider human activity monitoring applications implemented on energy-constrained IoT devices. We denote the period over which the total energy budget is provided as T_P, as summarized in Table 1. *REAP* computes the energy allocations at runtime with a period of T_P, which is set to one hour in our experiments. If the energy consumption over this period exceeds the amount of harvested energy and remaining battery level, the device powers down and misses user activity. Hence, our goal is to maximize the active time and the expected accuracy over a given period T_P.

Table 1: Summary of symbols used in the optimization problem.

Symbol	Description	Symbol	Description
T_P	Activity period	$J(t)$	Objective function
E_b	Energy budget	t_i	Active time of DP_i
a_i	Recognition accuracy of DP_i	α	Accuracy-active time trade-off parameter
P_{off}	Power consumption in the off state	P_i	Power consumption of DP_i

Suppose that the IoT device can operate at N distinct DPs. The recognition accuracy achieved by design point i is denoted by a_i, while the corresponding power consumption is given as P_i for $1 \leq i \leq N$. In addition to these design points, we denote the time that the device remains off as t_{off}. Finally, the power consumption during the off period, which is due to the energy harvesting and the battery charging circuitry, is denoted by P_{off}.

3.2 Optimization Problem Formulation

In a given activity period, the system may operate at different design points, resulting in varying levels of active time and accuracy. Let t_i denote the amount of time DP i is utilized during T_P. The active time of the device is simply given by the sum of the active times of each DP: $\sum_{i=1}^{N} t_i$. Likewise, the expected accuracy over the activity period can be expressed as $E\{a\} = \frac{1}{T_P} \sum_{i=1}^{N} a_i t_i$. The expected accuracy is a useful metric that incorporates both active time and accuracy, but it does not allow emphasis of one over the other. Therefore, we define a generalized cost function and solve the following optimization problem:

$$maximize \qquad J(t) = \frac{1}{T_P} \sum_{i=1}^{N} a_i^{\alpha} t_i \tag{1}$$

$$subject\ to \qquad t_{off} + \sum_{i=1}^{N} t_i = T_P \tag{2}$$

$$and \qquad P_{off} t_{off} + \sum_{i=1}^{N} P_i t_i \leq E_b \tag{3}$$

$$t_i \geq 0 \quad 0 \leq i \leq N \tag{4}$$

Objection function $J(t)$: The parameter α in Equation 1 enables a smooth trade-off between the active time and accuracy. When $\alpha = 1$, the objective function reduces to the expected accuracy. Similarly, when $\alpha = 0$, the objective function reduces to total active time. In general, the objective function gives a higher weight to the accuracy when $\alpha > 1$, and to active time when $\alpha < 1$.

Constraints: The constraint given in Equation 2 states that the sum of the active times and off period is equal to the activity period T_P. Similarly, Equation 3 specifies the energy budget constraint. The left-hand side gives the sum of the energy consumed in the off state and active states. The energy budget E_b on the right-hand side is determined by energy allocation techniques using the expected amount of harvested energy and battery capacity [9]. Finally, Equation 4 ensures that all active times are non-negative.

3.3 Runtime Optimization Algorithm

The solution to the optimization problem formulated in Section 3.2 gives the active times of each design point that maximize the objective function in Equation 1. We must solve this problem at runtime because the available energy budget is not known at design time. Furthermore, the importance given to accuracy versus active time (i.e., α) may change due to user preferences.

The optimization objective and the constraints in Equations 1-3 are linear in the decision variables t_i and t_{off}. Thus, we use a procedure based on the simplex algorithm [17], as outlined in Algorithm 1. The inputs are the energy budget E_b, Pareto-optimal DPs, and the maximum number of iterations. The output is a vector with the values of decision variables t_i $1 \leq i \leq N$ and t_{off} that maximize the objective. We start the optimization process by constructing a tableau with the initial conditions. The first row of the tableau describes the objective function, while the other rows describe the constraints. In each iteration of the procedure, we find the pivot column by finding the column with the largest value in the last row of the tableau. Using the pivot column, we next find the pivot row in the tableau in Line 8 of the algorithm. Then, we update the tableau using the pivot column and row. The procedure terminates when all the entries in the last row are non-positive. In this case, the pivot column is set as negative and the optimal solution is returned. *Our implementation takes 1.5 ms on our prototype.* Since the optimization algorithm runs every hour, it takes a negligible portion of the activity period and energy budget.

Algorithm 1: The *REAP* Procedure

Input : Design points, energy budget E_b, max. iterations
Output: Time allocated to each design point

```
1  Initialize the tableau with objective function and constraints
2  Add slack variables for inequality constraints
3  while iter ≤ max. iterations do
4  |  PivotCol ← findPivotCol(tableau)
5  |  if PivotCol < 0 then
6  |  |  return Optimal Solution
7  |  end
8  |  PivotRow ← findPivotRow(tableau, PivotCol)
9  |  Update the tableau using the PivotCol and PivotRow
10 end
```

4 HUMAN ACTIVITY RECOGNITION CASE STUDY

REAP is broadly applicable to energy-harvesting IoT devices that operate under a fixed energy budget. To illustrate the optimization results on a real example, we employ human activity recognition, i.e., HAR, as a driver application [4].

4.1 Background and Baseline Implementation

There is a steady increase in the use of wearable and mobile devices for the treatment of movement disorders and obesity-related diseases [15]. This technology enables data collection while the patients perform their daily activities. The first step in this effort is to understand what activity the user is performing at a given time. For example, the gait quality of the patient cannot be checked unless we know the user is walking. Therefore, HAR on mobile devices has recently attracted significant attention [2].

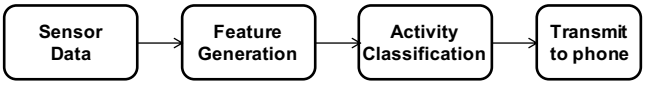

Figure 1: Overview of the human activity recognition application.

We implement a HAR application on a custom prototype based on the TI-Sensortag IoT board [5] and a passive stretch sensor, as outlined in Figure 1. It starts with sampling of the accelerometer and stretch sensors. The streaming sensor data is processed using the TI-CC2650 MCU to generate the feature vector. This feature vector is then processed by a parameterized neural network to infer the activity of the user. Finally, the inferred activity is transmitted to a host device, such as a phone, using the Bluetooth Low Energy (BLE) protocol.

The energy consumption and recognition accuracy of HAR depends on the types of sensors used, active time of the sensors, the type of features and the complexity of the classifier. The left side of Figure 2 shows the different configurations available for the sensors. For instance, we can use all three axes of the accelerometer or turn off selected axes to lower the energy consumption. In the extreme case, we can turn off the accelerometer to completely eliminate its energy consumption. Once we choose the configuration of the sensors, we can choose the sensing period, i.e. the time for which sensors are active, for each activity duration. By default, the sensors are turned on during the full activity duration. Turning off the sensors early, such as after 50% of the activity duration, provides energy savings at the cost of missed data points, hence, accuracy. We can also control the complexity of the features to trade off accuracy and energy consumption. Complex features, such as Fast Fourier Transform (FFT) and Discrete Wavelet Transform (DWT), offer higher accuracy at the expense of higher energy consumption. In contrast, statistical features have lower energy consumption, albeit with lower accuracy. Finally, the structure and depth of the NN classifier can be controlled to obtain further energy-accuracy trade-off, as illustrated in Figure 2.

We exploit this trade-off between energy and accuracy to design 24 different DPs implemented on the TI-Sensortag based prototype, as described in the following section.

Sensors		Computation		
Accel. axes	Stretch	Sensing period (%)	Signal features	NN structure
X, Y, Z	Yes	100	DWT of accel.	4×12×7
X, Y		75	16-FFT of stretch	
X or Y	No	50	Statistics of accel.	4×8×7
None		40	Statistics of stretch	4×7

Accuracy ↑ Energy ↓

Figure 2: The knobs used to obtain design points with different energy-accuracy trade-offs.

4.2 Pareto-Optimal Design Points

We design a total of 24 DPs by exploiting the energy-accuracy trade-off illustrated in Figure 2. We start by using all the axes of the accelerometer, generating complex features, and using an NN classifier with 2 hidden layers, which provide the highest recognition accuracy. Then, we progressively reduce the number of axes of the accelerometer and sensing period to reduce the energy consumption of the DPs. We always use the passive stretch sensor in our DPs, since it has a low energy consumption. There is a need

Table 2: Accuracy, execution time, power and energy consumption of different human activity recognition application design points.

Design point description			MCU exec. time distribution (ms)				Per activity summary			
DP no.	Features	Accuracy (%)	Accel. features	Stretch features	NN classifier	Total	MCU energy (mJ)	Sensor energy (mJ)	Energy (mJ)	Power (mW)
1	Statistical acceleration, 16-FFT stretch	94	0.83	3.83	1.05	5.71	2.38	2.10	4.48	2.76
2	Statistical y-axis accel., 16-FFT stretch	93	0.27	3.83	1.00	5.10	2.29	1.43	3.72	2.30
3	Statistical x- and y-axis accel. (0.8 s), 16-FFT stretch	92	0.27	3.83	0.90	5.00	2.10	0.84	2.94	1.82
4	Statistical y-axis accel. (0.6 s), 16-FFT stretch	90	0.14	3.83	1.00	4.97	2.09	0.57	2.66	1.64
5	16-FFT stretch	76	0.00	3.83	0.88	4.71	1.85	0.08	1.93	1.20

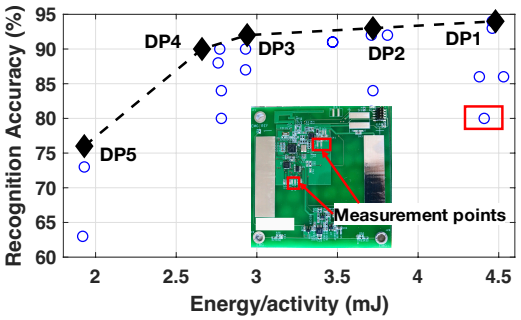

Figure 3: The energy-accuracy trade-off of various design points. The dashed line connects the selected 5 design points.

for detailed accuracy and energy consumption characterization of each DP to obtain the Pareto-optimal design points. To find the accuracy of each design point, we performed experiments with 14 different users. We use a total of 3553 activity windows from the experiments and labeled each window with the corresponding activity. Each DP is designed using 60% of this data for training, 20% for validation and the remaining 20% for testing.

All 24 design points are implemented on our prototype to profile the execution time and measure the power consumption using the test pads on our prototype. Figure 3 shows the recognition accuracy and energy per activity for each design point. As expected, each DP offers a unique energy-accuracy trade-off. For example, DP1 shows the highest accuracy with the highest energy consumption while DP5 shows the lowest recognition accuracy and energy consumption. However, some design points do not offer any benefit in the energy-accuracy trade-off. For example, the design point marked with a red rectangle is dominated by DP2, DP3, and DP4. Hence, we consider 5 Pareto-optimal design points shown using black diamonds (DP1 to DP5) to validate the REAP framework. Table 2 summarizes the details of the configuration, accuracy, execution time and energy for 5 Pareto-optimal DPs.

Design Point-1 (DP1): DP1 offers the highest accuracy by utilizing all three axes of the accelerometer for the entire activity window of 1.6 s. It uses 16-FFT of the stretch sensor data and statistical features of the accelerometer, such as the mean and standard deviations. DP1 has the highest accuracy of 94% at the cost of highest energy consumption of 4.48 mJ per activity. The

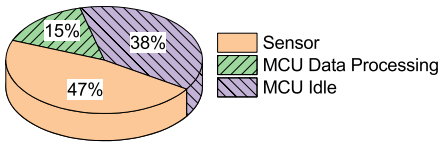

Figure 4: Energy consumption distribution of DP1 over one-hour activity period T_P. Total energy consumption is 9.9 J.

energy breakdown in Figure 4 shows that about 47% of the energy consumption is due to the sensors. Thus, reducing the sensor activity is an effective mechanism to save energy.

Design Point-2 (DP2): DP2 reduces the sensory energy by utilizing only the y-axis of the accelerometer along with the stretch sensor. As depicted in Table 2, the energy consumption of the sensor reduces from 2.10 mJ to 1.43 mJ. It achieves an accuracy of 93%, which is 1% lower than DP1.

Design Point-3 (DP3): As shown in Figure 2, reducing the sensing period leads to a lower energy consumption. DP3 exploits this by sampling the x- and y-axes of the accelerometer for 50% of each activity window, i.e., 0.8 s. As a result, the energy consumption of the sensor reduces to 0.84 mJ and the total energy consumption of DP3 becomes 2.94 mJ per activity, while the recognition accuracy drops to 92%.

Design Point-4 (DP4): DP4 is similar to DP3, except that the sensing period of the accelerometer is further reduced to 40% (0.6 s). This reduces the energy consumption of DP4 to 2.66 mJ per activity with recognition accuracy of 90%.

Design Point-5 (DP5): DP5 uses only the stretch sensor for data features to minimize energy consumption. The energy consumption reduces to 1.93 mJ per activity, the lowest energy consumption among all our design points. However, it also has the lowest recognition accuracy of 76%.

Offloading to a host: Finally, we note that the raw sensor data can be directly sent to a host device, such as a smartphone or server, for processing. To assess the viability of this alternative, we implemented and measured its energy consumption. Sending the raw sensor data over BLE consumes 5.5 mJ per activity without any significant increase in the recognition accuracy. In contrast, transmitting just the recognized activity consumes only about 0.38 mJ per activity. Hence, offloading is not an energy-efficient choice.

5 EXPERIMENTAL EVALUATION

5.1 Experimental Setup

IoT device: We use a custom prototype based on the TI-Sensortag IoT platform [5] to implement the proposed design points. The prototype consists of a TI CC2650 MCU, Invensense MPU-9250 motion sensor unit, a stretch sensor and energy harvesting circuitry. Sensors are sampled at 100 Hz and the MCU runs at 47 MHz frequency. Power measurements from the prototype and data from 14 user subject studies are used to obtain the 24 design points.

Energy harvesting data: We use the solar radiation data measured by the NREL Solar Radiation Research Laboratory to obtain the energy harvesting profile from January 2015 to October 2018 [18]. We use the profile for each hour within this data to generate the energy budget. These energy budgets are then used to evaluate *REAP* and the static design points in Section 5.4.

5.2 Expected Accuracy and Active Time Analysis

We first analyze the results of the proposed optimization approach as a function of the allocated energy over one-hour activity period T_P. In the most energy-constrained scenario, the minimum energy required to run the energy harvesting and monitoring circuitry is 0.18 J. In the opposite extreme, 9.9 J energy is sufficient to run DP1, the most power-hungry design point, throughout T_P. Therefore, we sweep the allocated energy starting with 0.18 J, and find the optimal active time of each DP using the proposed approach.

Figure 5(a) shows the expected accuracy ($\alpha = 1$) as a function of the energy budget. The expected accuracy of all the design points approaches to zero when the energy budget is close to 0.18 J, since the device is almost always off. As the energy budget increases (Region 1), the accuracy of all DPs starts growing since they can become active. None of the design points can afford to stay 100% active under the energy budget in Region 1. We observe that design point with the lowest energy consumption (DP5) achieves significantly higher accuracy because it can stay in the active state much longer. *REAP* successfully matches or exceeds the accuracy of DP5 under the most energy constrained scenario. When the energy budget goes over 4.3 J, DP5 can remain active throughout the activity period but its recognition accuracy saturates. The other DPs benefit from more energy in Region 2, while *REAP* outperforms all by utilizing them optimally. At 5 J energy budget, for example, *REAP* utilizes DP4 42% of the time and DP5 for 58% of the time to optimize the expected accuracy. Finally, all design points can remain active throughout the activity period when the energy budget is larger than 9.9 J. Hence, their accuracy saturates, and *REAP* reduces to DP1 beyond this point. In summary, *REAP* consistently outperforms or matches the accuracy of all individual DPs by utilizing *multiple DPs* optimally.

The active time of each DP *normalized to REAP* is plotted in Figure 5(b). DP5 is expected to have the longest active time since it has the least energy consumption. *REAP* successfully matches its active time in all the regions. In Region 1, *REAP* also achieves 2.3× larger active time compared to DP1 while providing significantly better accuracy. *REAP* consistently provides longer active times compared to DP1, DP2, and DP3 until the energy budget becomes large enough to sustain them throughout the activity period T_P.

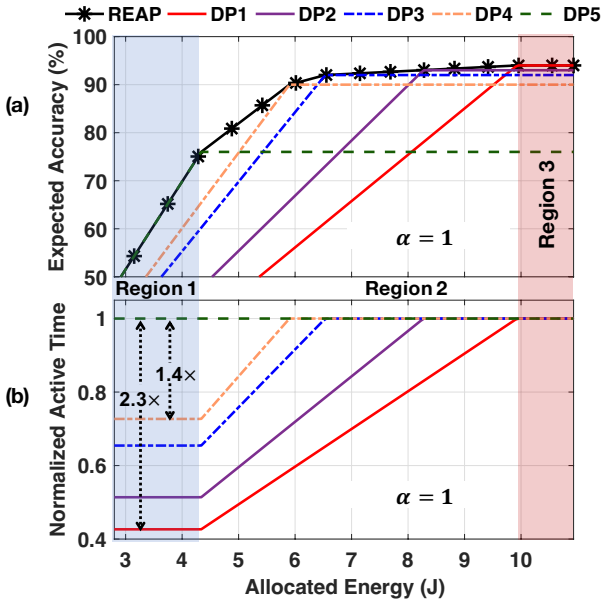

Figure 5: (a) Expected accuracy of *REAP* and design points. (b) Active time of each DP normalized to *REAP*.

5.3 Accuracy – Active Time Trade-off Analysis

Next, we analyze how *REAP* can exploit the trade-off between the accuracy and active time using the parameter α in objective function $J(t)$ in Equation 1. Since Section 5.2 considered the expected accuracy (α= 1), this section considers $\alpha > 1$, which gives more emphasis for higher accuracy.

As a representative example, Figure 6 shows the comparison of objective values of the 5 design points with *REAP* when α is set to 2. *REAP* always achieves higher performance than the lowest energy design DP5, since accuracy is given higher weight. The difference between *REAP* and DP5 increases further as *alpha* grows. When the energy budget is less than 6 J, DP4 outperforms all the other DPs, while *REAP* successfully matches it. In contrast, DP1, DP2, and DP3 have a very low performance, since they are mostly in the off state. When the energy budget exceeds 6 J, there is sufficient energy to provide a higher accuracy, but DP4 cannot exploit it. Hence, the higher accuracy design points become affordable and start outperforming DP4 one by one. Notably, *REAP* consistently outperforms or matches the static DPs, as we have also observed in Figure 5. For example, DP3 is able to provide the

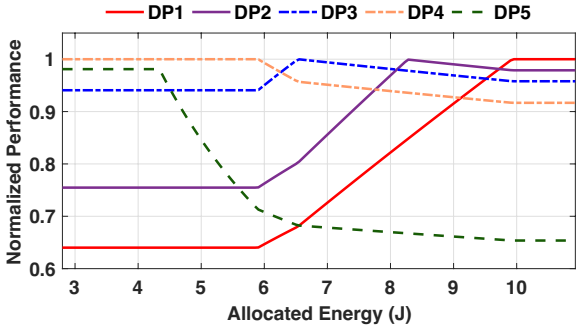

Figure 6: The objective value $J(t)$ of static design points (Equation 1) normalized to $J(t)$ of *REAP* with α= 2.

same performance as *REAP* when the energy budget is 6.5 J. As the energy allocation increases beyond 6.5 J, *REAP* starts outperforming DP3 by optimally switching between DP1, DP2 and DP3. This trend continues until the energy allocation reaches 9.9 J, beyond which there is sufficient energy to support DP1 alone. Thus, *REAP* reduces to DP1 in this region. In summary, *REAP* exceeds or matches the performance of any individual DP.

5.4 Case Study using Real Solar Energy Data

In this section, we evaluate *REAP* under real solar radiation data measured by NREL Solar Radiation Research Laboratory at Golden, Colorado. This data is used to calculate the amount of energy that can be harvested by a flexible solar cell [19] on our prototype. Using the harvested energy budget, we compare the performance of *REAP* against the static DPs over an entire month. Figure 7 shows the performance of *REAP* normalized to DP1, DP3, and DP5 as a function of α. Due to space limitation, we plot the DPs with the highest performance (DP1), lowest energy (DP5), and best trade-off (DP3). Our gains with respect to DP2 and DP4 are larger than that of DP3.

When active time is emphasized in the objection function ($\alpha = 0.5$), *REAP* outperforms DP1 by 1.4×–2.2× with an average improvement of 1.6× across the month. DP1 suffers the most in this case as it has the largest energy consumption among all the DPs. Since accuracy becomes more important with larger α, the improvement of *REAP* over DP1 reduces. However, we still obtain 1.1×–1.3× improvement even for $\alpha = 8$. We observe a similar trend in improvements for *REAP* over DP3 as well. The improvement is 1.1×–1.4× for $\alpha = 0.5$, and it gradually decreases with larger α. The improvements over DP3 are relatively lower, since DP3 offers the best trade-off between energy consumption and accuracy among our Pareto-optimal design points.

Finally, we see that the improvements over DP5 follow the opposite trend. When $\alpha = 0.5$, DP5 is able to match the active time of *REAP* due to its lower energy consumption. However, the performance of DP5 diminishes severely with increasing α. In summary, *REAP* can provide higher performance than any individual design point under any optimization objective. If the user needs higher accuracy, *REAP* can successfully adapt to new requirements. This can be utilized by the IoT device to tune its performance as user needs change.

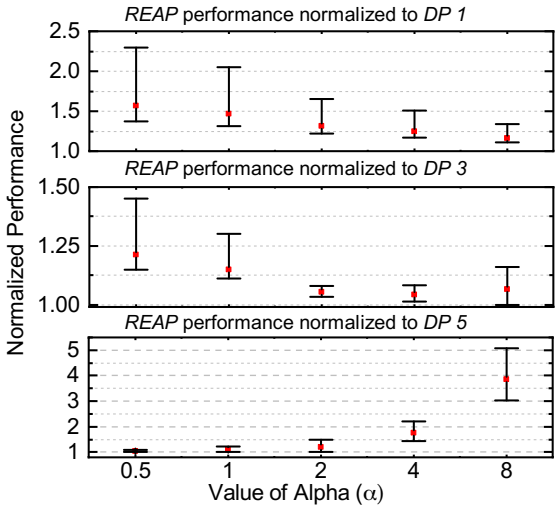

Figure 7: Performance (i.e., $J(t)$) achieved by *REAP* normalized to DP1, DP3, and DP5 during the month of September 2015. Error bars represent the range of improvement.

6 CONCLUSIONS

This paper presented a runtime accuracy-active time optimization technique for energy-constrained IoT devices. The proposed approach dynamically chooses design points with different energy-accuracy trade-offs to co-optimize the accuracy and active time under energy budget constraints. To demonstrate the effectiveness in a realistic setting, we implemented a human activity recognition application on a custom IoT prototype. We presented 5 Pareto-optimal design points with different energy-accuracy trade-offs. We achieve 46% higher expected accuracy and 66% longer active time compared to the highest performance design point, and 22% to 29% higher accuracy than low-power design points without sacrificing the active time.

Acknowledgment: This work was supported in part by NSF CAREER award CNS-1651624, DARPA Young Faculty Award (YFA) Grant D14AP00068, and National Research Foundation of Korea Grant NRF-2017R1D1A1B03032382.

REFERENCES

[1] Q. Liu *et al.*, "Gazelle: Energy-Efficient Wearable Analysis for Running," *IEEE Trans. Mobile Comput.*, no. 9, pp. 2531–2544, 2017.
[2] O. D. Lara *et al.*, "A Survey on Human Activity Recognition Using Wearable Sensors," *IEEE Commun. Surveys and Tut.*, vol. 15, no. 3, pp. 1192–1209, 2013.
[3] A. G. Bonomi, A. H. Goris, B. Yin, and K. R. Westerterp, "Detection of Type, Duration, and Intensity of Physical Activity Using an Accelerometer," *Medicine & Science in Sports & Exercise*, vol. 41, no. 9, pp. 1770–1777, 2009.
[4] G. Bhat *et al.*, "Online Human Activity Recognition using Low-Power Wearable Devices," in *Proc. of ICCAD*, 2018, pp. 72:1–72:8.
[5] TI SensorTag. https://store.ti.com/cc2650stk.aspx accessed 24 Nov. 2018.
[6] S. Sudevalayam and P. Kulkarni, "Energy Harvesting Sensor Nodes: Survey and Implications," *IEEE Commun. Surveys & Tutorials*, vol. 13, no. 3, pp. 443–461, 2011.
[7] H. Jayakumar, K. Lee, W. S. Lee, A. Raha, Y. Kim, and V. Raghunathan, "Powering the internet of things," in *Proc. of ISLPED*, 2014, pp. 375–380.
[8] N. S. Shenck and J. A. Paradiso, "Energy Scavenging With Shoe-Mounted Piezoelectrics," *IEEE Micro*, vol. 21, no. 3, pp. 30–42, 2001.
[9] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power Management in Energy Harvesting Sensor Networks," *ACM Trans. Embedd. Comput. Syst.*, vol. 6, no. 4, p. 32, 2007.
[10] C. M. Vigorito, D. Ganesan, and A. G. Barto, "Adaptive Control of Duty Cycling in Energy-Harvesting Wireless Sensor Networks," in *Proc. of IEEE Comm. Society Conf. on Sensor, Mesh and Ad Hoc Comm. and Networks*, 2007, pp. 21–30.
[11] B. Buchli, F. Sutton, J. Beutel, and L. Thiele, "Dynamic Power Management for Long-Term Energy Neutral Operation of Solar Energy Harvesting Systems," in *Proc. Conference on Embedd. Network Sensor Syst.*, 2014, pp. 31–45.
[12] G. Bhat, J. Park, and U. Y. Ogras, "Near-Optimal Energy Allocation for Self-Powered Wearable Systems," in *Proc. of ICCAD*, 2017, pp. 368–375.
[13] F. K. Shaikh and S. Zeadally, "Energy Harvesting in Wireless Sensor Networks: A Comprehensive Review," *Renew. Sust. Energ. Rev.*, vol. 55, pp. 1041–1054, 2016.
[14] R. Jafari *et al.*, "Physical Activity Monitoring for Assisted Living at Home," in *Int. Workshop on Wearable and Implantable Body Sensor Net.*, 2007, pp. 213–219.
[15] A. J. Espay *et al.*, "Technology in Parkinson's Disease: Challenges and Opportunities," *Movement Disorders*, vol. 31, no. 9, pp. 1272–1282, 2016.
[16] G. Bhat, R. Deb, and U. Y. Ogras, "OpenHealth: Open Source Platform for Wearable Health Monitoring," *IEEE Design Test*, 2019, to be published, doi: 10.1109/MDAT.2019.2906110.
[17] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. MIT press, 2009.
[18] "NREL Solar Radiation Research Laboratory: Baseline Measurement System (BMS)," https://midcdmz.nrel.gov/srrl_bms/, accessed 24 Nov. 2018.
[19] FlexSolarCells, "SP3-37 Datasheet," 2013, http://www.flexsolarcells.com/PowerFilm-Solar-OEM-Components.php , accessed 11 Nov. 2018.