

# Learning Datum-Wise Sampling Frequency for Energy-Efficient Human Activity Recognition

Weihao Cheng and Sarah Erfani and Rui Zhang and Kotagiri Ramamohanarao

School of Computing and Information Systems, The University of Melbourne

{weihao@student., sarah.erfani@, rui.zhang@, kotagiri@}unimelb.edu.au

## Abstract

Continuous Human Activity Recognition (HAR) is an important application of smart mobile/wearable systems for providing dynamic assistance to users. However, HAR in real-time requires continuous sampling of data using built-in sensors (e.g., accelerometer), which significantly increases the energy cost and shortens the operating span. Reducing sampling rate can save energy but causes low recognition accuracy. Therefore, choosing adaptive sampling frequency that balances accuracy and energy efficiency becomes a critical problem in HAR. In this paper, we formalize the problem as minimizing both classification error and energy cost by choosing dynamically appropriate sampling rates. We propose *Datum-Wise Frequency Selection* (DWFS) to solve the problem via a continuous state Markov Decision Process (MDP). A policy function is learned from the MDP, which selects the best frequency for sampling an incoming data entity by exploiting a datum related state of the system. We propose a method for alternative learning the parameters of an activity classification model and the MDP that improves both the accuracy and the energy efficiency. We evaluate DWFS with three real-world HAR datasets, and the results show that DWFS statistically outperforms the state-of-the-arts regarding a combined measurement of accuracy and energy efficiency.

## 1 Introduction

The task of Human Activity Recognition (HAR) is to recognize an individual's daily activities, such as walking, running and cycling. Due to the rapid development of ubiquitous computing technology, there is a huge demand for applying HAR on mobile/wearable devices that facilitate the understanding of a user's behaviors and provide intimate assistance to the user (Lockhart, Pulickal, and Weiss 2012). However, HAR in real-time requires continuous sampling data using built-in sensors (e.g., accelerometer, microphone, and camera), which causes excessive power consumption that greatly shortens the lifespan of the mobile devices (Khan et al. 2016). Simply using a low sampling frequency can save energy, but this comes at the cost of reduced recognition accuracy (Krause et al. 2005). Therefore, dynamically selecting proper sampling frequency to balance accuracy and energy efficiency becomes an emergent challenge for HAR on resource constrained mobile devices.

Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Most of the existing HAR methods use a fixed sampling frequency for data acquisition. A problem of these methods is that, they can either spend redundant energy for identifying highly discriminable instances, or obtain unsatisfactory accuracy for confusing instances. Several works have been proposed for seeking dynamic sampling frequency for HAR (Wang et al. 2010; Yan et al. 2012; Yurur et al. 2015). An A3R method (Yan et al. 2012) follows a table of heuristic rules for frequency selection, and it decides to switch frequency based on a predefined threshold of prediction probability. Some other methods (Wang et al. 2010; Yurur et al. 2015) solve the problem with a discrete state Markov Decision Process (MDP), which merely takes one of the predefined user states as input and returns limited actions (increase/keep/decrease) for changing the frequency. These methods lack direct information extraction from data sample and cannot find a desired optimal performance on balancing accuracy and energy efficiency.

In this paper, we propose an effective method that dynamically chooses sampling frequencies for HAR. In contrast to the existing approaches, our method directly exploits the information from recently observed data instances. Given a sequence of data entities, our goal is to dynamically choose frequencies to sample each data entity for inference, so that the recognition accuracy and the energy efficiency are balanced in a desired way. We formalize the problem as finding an optimal classification model and dynamically appropriate sampling frequencies that minimize an objective function regarding overall classification error and total energy cost. We propose Datum-Wise Frequency Selection (DWFS) to address the minimization problem through an MDP, where a policy function is learned for choosing the best sampling frequency based on a continuous state. We assign the policy function with the datum-wise property, where the feature representation of sampled data is utilized to build a connection between MDP states and sampling frequencies. DWFS models a mutual relationship between the classification model and the MDP, thereby, we propose to learn their parameters via an alternative optimization approach. More specifically, our main contributions can be summarized as follows:

- We formalize a problem which finds an optimal classification model and dynamically appropriate sampling frequencies to minimize an objective function regarding a

combined measurement of classification error and energy cost.

- We propose DWFS to solve the minimization problem. DWFS utilizes a continuous state MDP, where a datum-wise policy function is proposed to select the best sampling frequency by directly exploiting the information from recently sampled data.
- DWFS unifies the parameters of the classification model and the policy function in one model, thereafter, we propose an alternative optimization approach where the parameters are mutually enhanced.

We conduct extensive experiments on 3 real-world HAR datasets to evaluate DWFS. The results demonstrate that DWFS statistically outperforms the state-of-the-arts in terms of a combined measurement of accuracy and energy efficiency.

## 2 Related Works

Human Activity Recognition (HAR) has been widely investigated as an important topic of artificial intelligence (Ravi et al. 2005; Plötz, Hammerla, and Olivier 2011; Liu et al. 2015; Yang et al. 2015; Hammerla, Halloran, and Plötz 2016; Liu et al. 2016). Recently, the pervasive ubiquitous computing prompts a large number of mobile based HAR applications (Lockhart, Pulickal, and Weiss 2012). A significant problem for HAR on such resource constraint devices is to deal with the trade-off between recognition accuracy and power consumption. Krause et al. (2005) discovered that a high sensing sampling frequency will improve the recognition accuracy, but shorten the battery life of a device. A number of energy-efficient HAR systems, such as ‘SeeMon’ (Kang et al. 2008), ‘EEMSS’ (Wang et al. 2009), and ‘Jigsaw’ (Lu et al. 2010) apply a reduced sampling rate during runtime. Khan et al. (2016) proposed to find a minimal sampling rate that perverse a certain accuracy. However, these methods using fixed sampling rate can cause low performance of either accuracy or energy efficiency. Yan et al. (2012) and Qi et al. (2013) focused on finding energy-efficient features and sampling rate for each activity during training, and proposed to adaptively change sensor sampling rate based on detected activity and predefined thresholds. However, their methods are heuristic in testing that cannot achieve a dynamic optimal.

Markov Decision Process (MDP) is a framework successfully applied to solve dynamic problems (Hoey et al. 2010; Puterman 2014; Zhang and Shah 2014; Gilbert et al. 2015; Yadav et al. 2016; Fang, Li, and Cohn 2017). Several works especially focus on cost sensitive dynamic problems via MDP. Trapeznikov and Saligrama (2013) proposed an MDP based stage-wise decision method which selects a number of sensors for sequential classification under budgets constraint. Dulac-Arnold et al. (2011) proposed a datum-wise feature selection technique using an MDP, where the features of a data point are dynamically selected under dimension sparsity constraint. MDP is also applied in the field of energy-efficient sensing. Wang et al. (2010) proposed a method that obtains Markov-optimal sensing policy for user state estimation. Yurur et al. (2015) extended the work of

Wang et al. and considered the trend of user preferences to regulates sensor sampling settings. However, these methods only utilize predefined user state to determine the next sensing policy, where the beneficial information from data samples is not exploited.

## 3 Methodology

In this section, we first formalize the problem of balancing recognition accuracy and energy efficiency as minimizing an objective function regarding recognition error and energy cost. Then, we propose Datum-Wise Frequency Selection (DWFS) based on a continuous state MDP, which can sequentially choose the optimal sensor sampling frequencies for incoming data entities.

### 3.1 Problem Statement

Let  $F = \{f_1, f_2, \dots, f_K\}$  be a set of  $K$  sampling frequencies supported by a sensor, where  $f_1 < f_2 < \dots < f_K$ . The energy costs of using these frequencies are  $c_{f_1}, c_{f_2}, \dots, c_{f_K}$ , respectively, and we have  $c_{f_1} < c_{f_2} < \dots < c_{f_K}$ . Let  $\mathbf{x}$  be a data entity with a corresponding activity label  $y \in \{1, 2, \dots, m\}$ , where  $m$  is the number of the activities. Since a data entity  $\mathbf{x}$  represents an activity in the space of infinite dimension, it cannot be used without been sampled by a sensor. We define a sampling function  $g(\mathbf{x}, f)$  which returns an observed data instance  $\tilde{\mathbf{x}} = g(\mathbf{x}, f)$  by sampling  $\mathbf{x}$  with a frequency  $f \in F$ . It is worth noting that  $f$  is implicitly represented by  $\tilde{\mathbf{x}}$  for simplicity of notation. Given an  $\tilde{\mathbf{x}}$ , we can use a classification model parameterized by  $\theta$  to infer  $\tilde{\mathbf{x}}$ , then the model outputs a vector of label probabilities  $\mathbf{p} \in [0, 1]^m$  for the  $m$  activities:

$$\mathbf{p} = (p_1, p_2, \dots, p_m), \quad (1)$$

where the  $y$ -th element  $p_y$  of  $\mathbf{p}$  is the probability of the activity  $y$ :

$$p_y = p(y | \tilde{\mathbf{x}}; \theta). \quad (2)$$

Suppose there is a sequence of labeled data entities  $Q = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(N)}, y^{(N)})\}$ . Our goal is twofold: 1) we want to find an optimal  $\theta$  for the classification model, and 2) for each data entity  $\mathbf{x}^{(t)}$ , we want to select a proper frequency  $f^{(t)} \in F$  to obtain  $\tilde{\mathbf{x}}^{(t)} = g(\mathbf{x}^{(t)}, f^{(t)})$ , that a combined measurement of the overall classification error and the total energy cost is minimized:

$$\min_{\theta, f^{(1)}, \dots, f^{(N)}} \sum_{t=1}^N 1\{\hat{y}^{(t)} \neq y^{(t)}\} + \lambda \sum_{t=1}^N c_{f^{(t)}}, \quad (3)$$

where  $\hat{y}^{(t)} = \operatorname{argmax}_y p(y | \tilde{\mathbf{x}}^{(t)}; \theta)$  is the predicted label of  $\tilde{\mathbf{x}}^{(t)}$ , and  $\lambda$  is a predefined weight parameter between the error and the cost. We denote this combined measurement as **Error-Cost Index**. For the convenience of using function optimization to reduce the proposed Error-Cost Index, we replace the classification error  $1\{\hat{y}^{(t)} \neq y^{(t)}\}$  in Eq. 3 with cross-entropy loss, and formalize an objective function as:

$$\min_{\theta, f^{(1)}, \dots, f^{(N)}} \sum_{t=1}^N -\log p(y^{(t)} | \tilde{\mathbf{x}}^{(t)}; \theta) + \lambda \sum_{t=1}^N c_{f^{(t)}}, \quad (4)$$

where  $\lambda$  is a predefined weight parameter between the cross-entropy loss and the energy cost. In an experimental environment, one can naively test all the possible sampling frequencies for each data entity to obtain the optimum solution. However, this is infeasible for real-time testing, since a sensor can only sample an incoming entity once. Therefore, we need to find an effective way to solve the problem.

### 3.2 Datum-Wise Frequency Selection (DWFS)

We propose DWFS to solve the optimization problem of Eq. 4. DWFS effectively predicts the best sampling frequency for an incoming data entity based on recent context. A significant advantage of DWFS over the existing methods is that, it models a direct connection from data to sampling frequencies, thus the model is so called **Datum-Wise**. DWFS solves the minimization problem of Eq. 4 by incorporating a continuous state MDP, where a **policy**  $\pi$  is learned to select the next sampling frequency based on the current state of the MDP. A state of the MDP is described by a pair of the feature representation of a previously sampled data instance and the prediction probabilities of this instance. The detailed description of DWFS is provided in the rest of this section. We first define our continuous state MDP, and transform the original problem into the MDP problem. We then design the policy of the MDP. Next, we introduce an alternative learning approach to optimize the parameters of DWFS. Finally, we describe the algorithm of DWFS to select the sampling frequencies in real-time testing.

**The Markov Decision Process of DWFS** Let  $\phi(\tilde{\mathbf{x}}) \in \mathbb{R}^d$  be the feature representation of  $\tilde{\mathbf{x}} = g(\mathbf{x}, f)$ , where  $d$  is the dimension of  $\phi(\tilde{\mathbf{x}})$ . We introduce the MDP to solve our problem as follows:

- $S$  is an infinite space of states. Each state  $\mathbf{s} \in S$  is defined as a vector  $\mathbf{s} = (\phi(\tilde{\mathbf{x}}), \mathbf{p}) \in \mathbb{R}^{d+m}$ , which is concatenated by the feature representation  $\phi(\tilde{\mathbf{x}})$  and the label probabilities  $\mathbf{p}$  of  $\tilde{\mathbf{x}}$ . Specifically, we define  $\mathbf{s}^{(t)} = (\phi(\tilde{\mathbf{x}}^{(t)}), \mathbf{p}^{(t)})$ , where  $\mathbf{p}^{(t)}$  is the label probabilities of  $\tilde{\mathbf{x}}^{(t)}$ .
- $A$  is a set of actions that  $A = \{a_1, a_2, \dots, a_K\}$ . The action  $a_k \in A$  maps to the frequency  $f_k \in F$ , which indicates the choice of  $f_k$ . We can consider that  $a_k \equiv f_k$ .
- $P_{sa}(s')$  is the transition probability function. Given a state  $\mathbf{s} \in S$  and an action  $a \in A$ ,  $P_{sa}(s')$  returns the probability of the next state  $\mathbf{s}'$  by taking the action  $a$ .
- $\gamma$  is the discount factor. We set  $\gamma = 1$  as for no discount involved, thereby every incoming data entity can be considered equally.
- $R(\mathbf{s})$ :  $S \mapsto \mathbb{R}$  is the reward function. It represents a valuable rewarded by visiting a state  $\mathbf{s} = (\phi(\tilde{\mathbf{x}}), \mathbf{p})$ . We define the reward function for our problem as:

$$R(\mathbf{s}) = \log p(y | g(\mathbf{x}, f); \theta) - \lambda c_f, \quad (5)$$

where  $f \in F$  is the frequency used to sample  $\mathbf{x}$ . Particularly, we define  $R(\mathbf{0}) = 0$ , where  $\mathbf{0}$  is a zero vector.

The dynamics of the MDP proceeds as follows: We start at an initial state  $\mathbf{s}^{(0)} = \mathbf{0}$ . We then choose the first action  $a^{(0)}$ ,

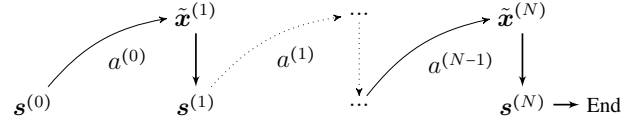


Figure 1: The dynamics of the MDP.

and use the corresponding sampling frequency  $f^{(1)}$  to sample the first data entity  $\tilde{\mathbf{x}}^{(1)}$ . As the result of  $f^{(1)}$ , we step into the next state  $\mathbf{s}^{(1)} = (\phi(\tilde{\mathbf{x}}^{(1)}), \mathbf{p}^{(1)})$ . We then choose the second action  $a^{(1)}$  based on  $\mathbf{s}^{(1)}$  and obtain a new state. We repeat this procedure until the last data entity  $\tilde{\mathbf{x}}^{(N)}$  is sampled. An intuitive representation of the MDP dynamics is shown in Figure 1, and the total reward of visiting the state sequence  $\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, \dots, \mathbf{s}^{(N)}$  is calculated as:

$$\begin{aligned} & R(\mathbf{s}^{(0)}) + \gamma R(\mathbf{s}^{(1)}) + \gamma^2 R(\mathbf{s}^{(2)}) + \dots + \gamma^N R(\mathbf{s}^{(N)}) \\ &= R(\mathbf{s}^{(0)}) + \sum_{t=1}^N \gamma^t R(\mathbf{s}^{(t)}). \end{aligned} \quad (6)$$

The essential problem of MDP is to find a policy function  $\pi: S \mapsto A$ , which chooses the best action for a given state to maximize a value function defined as:

$$V(\mathbf{s}, t) = \begin{cases} R(\mathbf{s}) + \int_{s' \in S} P_{sa}(s') V(s', t+1) & t < N \\ R(\mathbf{s}) & t = N \end{cases}. \quad (7)$$

By incorporating the MDP, we can rewrite the problem of Eq. 4 as:

$$\begin{aligned} & \min_{\theta, f^{(1)}, \dots, f^{(N)}} \sum_{t=1}^N -\log p(y^{(t)} | \tilde{\mathbf{x}}^{(t)}; \theta) + \lambda \sum_{t=1}^N c_{f^{(t)}} \\ &= \min_{\theta, f^{(1)}, \dots, f^{(N)}} \sum_{t=1}^N -\log p(y^{(t)} | \tilde{\mathbf{x}}^{(t)}; \theta) + \lambda c_{f^{(t)}} \\ &= \max_{\theta, f^{(1)}, \dots, f^{(N)}} \sum_{t=1}^N \log p(y^{(t)} | \tilde{\mathbf{x}}^{(t)}; \theta) - \lambda c_{f^{(t)}} \\ &= \max_{\theta, \pi} R(\mathbf{s}^{(0)}) + \sum_{t=1}^N R(\mathbf{s}^{(t)}) \\ &= \max_{\theta, \pi} V(\mathbf{s}^{(0)}, 0). \end{aligned} \quad (8)$$

Therefore, our goal is to find optimal classification parameter  $\theta^*$  and policy  $\pi^*$  that maximize the value function  $V(\mathbf{s}^{(0)}, 0)$ .

**Modeling the Policy** Suppose we have a fixed  $\theta$ . Let  $V^*(\mathbf{s}, t)$  be the optimal value function regarding  $\pi^*$ . For  $t = N$ , we have  $V^*(\mathbf{s}, t) = R(\mathbf{s})$ . For  $t < N$ , we can use Bellman's equation to recursively calculate  $V^*(\mathbf{s}, t)$  as:

$$V^*(\mathbf{s}, t) = R(\mathbf{s}) + \max_{a \in A} \int_{s' \in S} P_{sa}(s') V^*(s', t+1), \quad (9)$$

where the optimal values of  $V^*$  for each state are obtained in a dynamic programming manner. Correspond to  $V^*$ , the optimal policy function  $\pi^*$  regarding  $\mathbf{s}$  is calculated as:

$$\pi^*(\mathbf{s}) = \operatorname{argmax}_{a \in A} \mathbb{E}_{t \sim \mathcal{T}} \int_{s' \in S} P_{sa}(s') V^*(s', t), \quad (10)$$

where  $\mathcal{T}$  is a distribution of  $t$ . Since  $\pi^*(\mathbf{s})$  is independent of  $t$ , the output of  $\pi^*(\mathbf{s})$  given  $\mathbf{s}$  should be maximized on  $\mathcal{T}$ . As the state  $\mathbf{s} = (\phi(\tilde{\mathbf{x}}), \mathbf{p})$  is in a continuous space, we are unable to use the traditional tabular approach to obtain  $\pi(\mathbf{s})$ . Instead, we stimulate  $\pi(\mathbf{s})$  with a probabilistic model  $p(a | \mathbf{s}; \psi)$  parameterized by  $\psi$ :

$$\pi(\mathbf{s}) = \operatorname{argmax}_{a \in A} p(a | \mathbf{s}; \psi). \quad (11)$$

We consider the parameter  $\psi = \psi(\mathbf{s})$  as a function of  $\mathbf{s}$ . The probability  $p(a | \mathbf{s}; \psi)$  is then given through a softmax assignment based on  $\psi(\mathbf{s})$ :

$$p(a = a_k | \mathbf{s}; \psi) = \frac{e^{\psi_k(\mathbf{s})}}{\sum_{i=1}^K e^{\psi_i(\mathbf{s})}}. \quad (12)$$

Therefore, we can learn the parameter  $\psi$  to obtain the policy  $\pi(\mathbf{s})$  of the MDP, and our problem of Eq. 8 is then rewritten as:

$$\max_{\theta, \psi} V(\mathbf{s}^{(0)}, 0). \quad (13)$$

**Learning DWFS** Learning DWFS consists of two tasks: optimizing the parameter  $\theta$  for the classification model and optimizing the parameter  $\psi$  for the MDP. We consider that  $\theta$  and  $\psi$  are affected by each other, and can be mutually enhanced. It is clear that  $\psi$  can be refined based on  $\theta$ , as the MDP reward  $R(\mathbf{s})$  is a function of  $\theta$ . The problem is how to refine  $\theta$  based on  $\psi$ . We propose to use the results of the MDP to regulate the sample weights for learning  $\theta$ . Given a  $\psi$  and a training sequence  $Q$ , we can use the MDP to predict a sequence of frequencies  $\{f^{(1)}, \dots, f^{(N)}\}$ , one for each data entity  $\mathbf{x}^{(t)}$ . We assign a high weight to  $g(\mathbf{x}^{(t)}, f^{(t)})$  for learning  $\theta$ , so that  $\theta$  can put more emphasis on the instances sampled by the chosen frequencies, which improves the classification accuracy. Thereafter,  $\theta$  can be refined based on  $\psi$ . We propose an alternative learning approach to optimize  $\theta$  and  $\psi$ . Suppose we have a set of training sequences  $\{Q_0, Q_1, \dots, Q_n\}$ . At the beginning, we use  $Q_0$  to learn an initial  $\theta$  by:

$$\min_{\theta} \sum_{(\mathbf{x}, y) \in Q_0} \sum_{f \in F} -\log p(y | g(\mathbf{x}, f); \theta), \quad (14)$$

where all the sampled data instances are considered equally. Given a sequence  $Q$  iteratively picked from  $\{Q_1, \dots, Q_n\}$ , we fix  $\theta$ , and extract a set of state-action pairs  $H$  from  $Q$  based on Eq. 10. Each pair  $(\mathbf{s}, a) \in H$  is used as an instance for learning the MDP, thereby, the parameter  $\psi$  is optimized as:

$$\min_{\psi} \sum_{(\mathbf{s}, a) \in H} -\log p(a | \mathbf{s}; \psi). \quad (15)$$

We then fix  $\psi$ , and use the MDP to predict  $Q$ , that a sequence of chosen frequencies  $\{f^{(1)}, \dots, f^{(N)}\}$  is obtained. Let  $\mu_{\mathbf{x}, f}$  be the weight of the instance  $g(\mathbf{x}, f)$ , and we optimize  $\theta$  as:

$$\min_{\theta} \sum_{(\mathbf{x}, y) \in Q} \sum_{f \in F} -\mu_{\mathbf{x}, f} \log p(y | g(\mathbf{x}, f); \theta), \quad (16)$$

$$\text{with } \mu_{\mathbf{x}, f} = \beta, \text{ if } \mathbf{x} = \mathbf{x}^{(t)} \text{ and } f = f^{(t)}, \\ \mu_{\mathbf{x}, f} = 1, \text{ otherwise,}$$

---

### Algorithm 1 DWFS Generate $H$

---

**Input:** A sequence of training entities  $Q = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(N)}, y^{(N)})\}$   
**Output:** A set of state-action pairs  $H$

- 1: **while**  $t = N, N - 1, \dots, 1$  **do**
- 2:     **for**  $k = 1, 2, \dots, K$  **do**
- 3:          $\tilde{\mathbf{x}}_k^{(t)} = g(\mathbf{x}^{(t)}, f_k)$
- 4:          $\mathbf{s}_k^{(t)} = (\phi(\tilde{\mathbf{x}}_k^{(t)}), \mathbf{p}_k^{(t)})$
- 5:         **if**  $t == N$  **then**
- 6:              $V(\mathbf{s}_k^{(t)}) = R(\mathbf{s}_k^{(t)})$
- 7:         **else**
- 8:              $V(\mathbf{s}_k^{(t)}) = R(\mathbf{s}_k^{(t)}) + \max_{1 \leq k' \leq K} V(\mathbf{s}_{k'}^{(t+1)})$
- 9:         **end if**
- 10:     **end for**
- 11: **end while**
- 12:  $H = \emptyset$
- 13: **while**  $t = 1, 2, \dots, N - 1$  **do**
- 14:     **for**  $k = 1, 2, \dots, K$  **do**
- 15:          $k' = \operatorname{argmax}_{1 \leq k'' \leq K} V(\mathbf{s}_{k''}^{(t+1)})$
- 16:          $H = H \cup \{(\mathbf{s}_k^{(t)}, a_{k'})\}$
- 17:     **end for**
- 18: **end while**
- 19: **return**  $H$

---

where  $\beta > 1$  is a predefined parameter for imposing the weight. Accordingly, we alternately optimize  $\theta$  and  $\psi$  until the training set has been iterated for  $L$  rounds. Before concluding the entire DWFS learning algorithm, we first provide the algorithm for obtaining the state-action pairs set  $H$ . Based on the general framework of Eq. 10, we iteratively calculate the optimal  $V^*(\mathbf{s}, t)$  from  $t = N$  to 1, then we collect a number of  $K$  state-action pairs for each  $t$  from  $t = 1$  to  $N - 1$ . Let  $\tilde{\mathbf{x}}_k^{(t)}$  be the  $t$ -th instance  $\tilde{\mathbf{x}}^{(t)}$  sampled by the frequency  $f_k$  that  $\tilde{\mathbf{x}}_k^{(t)} = g(\mathbf{x}^{(t)}, f_k)$ . Let  $\mathbf{p}_k^{(t)}$  be the probabilities of  $\tilde{\mathbf{x}}_k^{(t)}$ , and  $\mathbf{s}_k^{(t)} = (\phi(\tilde{\mathbf{x}}_k^{(t)}), \mathbf{p}_k^{(t)})$ . We set the transition probability function  $P_{sa}(\mathbf{s}')$  as:

$$P_{sa}(\mathbf{s}') = \begin{cases} 1, & \mathbf{s} = \mathbf{s}_k^{(t)}, a = a_{k'}, \mathbf{s}' = \mathbf{s}_{k'}^{(t+1)}, \\ 0, & \text{otherwise.} \end{cases} \quad (17)$$

Therefore, the state  $\mathbf{s}_k^{(t)}$  only transits to  $\mathbf{s}_{k'}^{(t+1)}$  by taking the action  $a_{k'}$ , which indicates to use  $f_{k'}$  for sampling the next entity  $\mathbf{x}^{(t+1)}$ . The pseudocode of  $H$  generating algorithm is provided in Algorithm 1. Up to now, we can conclude the DWFS learning algorithm, where the pseudocode is provided in Algorithm 2.

**DWFS Inference Algorithm** After learning the parameter  $\theta^*$  of the classification model and the parameter  $\psi^*$  of the policy function  $\pi^*(\mathbf{s})$ , we can use the DWFS inference algorithm to select frequencies on a testing sequence. We start at an initial state  $\mathbf{s}^{(0)} = \mathbf{0}$ . For an incoming data entity  $\mathbf{x}^{(t)}$ , we obtain the frequency via the policy as:

$$f^{(t)} = a^{(t-1)} = \pi^*(\mathbf{s}^{(t-1)}). \quad (18)$$

---

**Algorithm 2** DWFS Learning

---

**Input:** A set of training sequences  $\{Q_0, Q_1, \dots, Q_n\}$ **Output:** Parameters  $\theta$  and  $\psi$ 

```
1: Initialize  $\theta, \psi$ 
2: Learn  $\theta$  by Eq. 14 using  $Q_0$ 
3: for  $l = 1, 2, \dots, L$  do
4:   for  $Q \in \{Q_1, \dots, Q_n\}$  do
5:     Fix  $\theta$ , update  $\psi$  by Eq. 15 using  $Q$ 
6:     Fix  $\psi$ , update  $\theta$  by Eq. 16 using  $Q$ 
7:   end for
8: end for
9: return  $\theta, \psi$ 
```

---

---

**Algorithm 3** DWFS Inference

---

**Input:** A sequence of data entities  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)}$   $\triangleright$   
 $N$  can be an arbitrary number or infinity.**Output:** A sequence of predicted labels  $\hat{y}^{(1)}, \hat{y}^{(2)}, \dots, \hat{y}^{(N)}$ 

```
1:  $\mathbf{s}^{(0)} = \mathbf{0}$ 
2: while  $t = 1, 2, \dots, N$  do
3:    $a^{(t-1)} = \pi^*(\mathbf{s}^{(t-1)})$ 
4:   Choose  $f^{(t)}$  corresponding to  $a^{(t-1)}$ 
5:    $\tilde{\mathbf{x}}^{(t)} = g(\mathbf{x}^{(t)}, f^{(t)})$ 
6:    $\hat{y}^{(t)} = \operatorname{argmax}_y p(y | \tilde{\mathbf{x}}^{(t)}; \theta^*)$ 
7:    $\mathbf{s}^{(t)} = (\phi(\tilde{\mathbf{x}}^{(t)}), \mathbf{p}^{(t)})$ 
8: end while
9: return  $\hat{y}^{(1)}, \hat{y}^{(2)}, \dots, \hat{y}^{(N)}$ 
```

---

We then change the sampling frequency of the sensor to  $f^{(t)}$ , and the sensor will sample out an observed data instance  $\tilde{\mathbf{x}}^{(t)} = g(\mathbf{x}^{(t)}; f^{(t)})$ . We use the classification model with  $\theta^*$  to predict the activity label  $\hat{y}^{(t)}$  for  $\tilde{\mathbf{x}}^{(t)}$ :

$$\hat{y}^{(t)} = \operatorname{argmax}_y p(y | \tilde{\mathbf{x}}^{(t)}; \theta^*). \quad (19)$$

We obtain the current state as  $\mathbf{s}^{(t)} = (\phi(\tilde{\mathbf{x}}^{(t)}), \mathbf{p}^{(t)})$ , which will be used to predict the frequency for the next data entity  $\mathbf{x}^{(t+1)}$ . We provide the pseudocode of the DWFS inference algorithm in Algorithm 3. It is worth noting that the length  $N$  of the testing sequence can be an arbitrary number or infinity, therefore, DWFS is capable to be applied in real-world scenarios.

## 4 Empirical Evaluation

In this section, we evaluate the performance of DWFS in terms of recognition accuracy, energy cost and the Error-Cost Index as shown in Eq. 3. The experimental scripts are written in Python 2.7 on a 64-bit Ubuntu 14.04 LTS operating system.

**Datasets:** We use the 3-axis acceleration data of 3 real-world HAR datasets to evaluate the performance of DWFS. 1) *Human Activity Sensing Consortium (HASC) 2011* (Kawaguchi et al. 2011): The data of 6 activities was collected with 100 readings per second by 7 subjects using iPhone/iPod. 2) *Human Activity Recognition on Smartphones Dataset (HARSD)* (Anguita et al. 2013): The data of

6 activities was collected with 50 readings per second by 30 subjects using a Samsung Galaxy S II. 3) *Daily Sport Activities dataset (DSA)* (Barshan and Yüsek 2013): The data of 19 activities was collected with 25 readings per second by 8 subjects using body-worn sensors (we use the data collected by the sensor placed on a subject’s torso).

**Data Preparation:** Let  $\mathcal{D}$  be a dataset containing a number of time series data. We conduct the experiments based on 5-fold cross-validation, where we take 1/5 of  $\mathcal{D}$  in turn for testing, and the other 4/5 of  $\mathcal{D}$  for training. For each training time series data, we divide it into several 5 seconds time series segments without overlapping, that each segment is considered as a data entity  $\mathbf{x}$  paired with a label  $y$ . We preserve the order of the segments to generate a sequence  $Q = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(N)}, y^{(N)})\}$ . Accordingly, we obtain a set of training sequences. We randomly concatenate all the testing time series into one time series, and transform it into a testing sequence following the same way of processing training data. We particularly denote  $N$  as the length of the testing sequence for the rest of the paper. The experimental results are reported as the average results of the 5 folds.

**Experiment Settings:** The frequencies  $F$  used for each dataset are set as follows: 1) *HASC*: 5Hz, 16Hz, 50Hz, 100Hz; 2) *HARSD*: 2Hz, 5Hz, 16Hz, 50Hz; 3) *DSA*: 2Hz, 5Hz, 16Hz, 25Hz. The highest frequency in  $F$  equals to the frequency used to collect the dataset. Given  $F = \{f_1, f_2, \dots, f_K\}$ , for each frequency  $f_k \in F$ , the corresponding energy cost is set as  $c_{f_k} = f_k/f_K$ , so that  $c_{f_k}$  is normalized in the range of  $(0, 1]$ . The normalization of  $c_{f_k}$  ensures the same loss of misclassification and using the highest frequency for a data entity when  $\lambda = 1.0$ . The feature representation  $\phi(\tilde{\mathbf{x}})$  is set as Fourier coefficients extracted from  $\tilde{\mathbf{x}}$ , where the coefficients are the intensities of the frequencies from 0Hz to 2Hz (step-size of 0.1Hz). The activity classification model takes  $\phi(\tilde{\mathbf{x}})$  as input. The iteration round  $L$  for training is set to 5.

**Settings of DWFS:** We use softmax regression as the classification model of DWFS, that the parameter  $\theta$  is an  $m \times (d+1)$  matrix including intercepts, and we optimize  $\theta$  iteratively by Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm. We model the parameter  $\psi$  as a  $K \times (d+m)$  matrix, that  $\psi_k(\mathbf{s}) = \psi_k \cdot \mathbf{s}$ , and we optimize  $\psi$  iteratively via BFGS as well. We set the parameter  $\beta = 1.2$  which is used to impose instance weight.

**Settings of Baselines:** We test 6 baseline methods in comparison with DWFS. The last two baselines are the variants of DWFS, where we test two different learning approaches instead of alternative learning. The settings of the baselines are as following. 1) *The method using constant sampling frequency:* We report  $K$  results for this method, one for each frequency. 2) *Random:* We randomly choose sampling frequencies. 3) *MDP-DS:* We implement an MDP process which takes predicted activity of the most recent instance as input state (Yurur et al. 2015). 4) *RNN:* We implement a Recurrent Neural Network (RNN) based method, where the architecture of the RNN consists of an LSTM layer with 32-dim output, a dense layer with  $K$ -dim output, and a softmax layer (Hammerla, Halloran, and Plötz 2016). 5) *DWFS-SL*

Table 1: Recognition Accuracy versus Energy Cost (Accelerometer). The unit of the energy cost values is Joule per hour (J/h).

HASC				
$f$	100Hz	50Hz	16Hz	5Hz
Accuracy	88.79 ( $\pm 1.84$ )	88.20 ( $\pm 2.10$ )	84.92 ( $\pm 2.93$ )	71.46 ( $\pm 2.78$ )
Energy	327.42 J/h	81.20 J/h	51.16 J/h	10.62 J/h
HARSD				
$f$	50Hz	16Hz	5Hz	2Hz
Accuracy	81.59 ( $\pm 2.21$ )	81.65 ( $\pm 2.32$ )	78.33 ( $\pm 2.56$ )	73.70 ( $\pm 2.82$ )
Energy	81.20 J/h	51.16 J/h	10.62 J/h	3.01 J/h
DSA				
$f$	25Hz	16Hz	5Hz	2Hz
Accuracy	77.33 ( $\pm 3.63$ )	76.18 ( $\pm 4.33$ )	72.49 ( $\pm 3.39$ )	63.57 ( $\pm 3.79$ )
Energy	55.45 J/h	51.16 J/h	10.62 J/h	3.01 J/h

(*Separate Learning*): We first learn  $\theta$  on the entire training data, and learn  $\psi$  based on  $\theta$ , that there is no dependence of  $\theta$  on  $\psi$ . 6) *DWFS-CVL (Cross-Validation Learning)*: We learn  $\theta$  on the entire training data, and learn  $\psi$  via 4-fold cross-validation on the training data, where 3/4 of the data is used to learn a temporal  $\tilde{\theta}$ , and  $\psi$  is iteratively learned based on  $\tilde{\theta}$  of each fold.

#### 4.1 Recognition Accuracy versus Energy Cost

We study the recognition accuracy and the energy cost using different sampling frequencies, where an accelerometer is utilized as the sensor. Given a dataset, we train  $K$  classifiers, one for each frequency in  $F$ , where the  $k$ -th classifier is trained with the data instances sampled using the  $k$ -th frequency. The recognition accuracy of each classifier is calculated as  $\frac{\#\{\text{correct prediction}\}}{N}$ . The energy cost of each frequency is derived from (Qi et al. 2013), where the authors provided a comprehensive list of energy consumptions regarding accelerometer sampling rates. According to the results shown in Table 1, one can generally obtain a better recognition accuracy with a higher sampling frequency, however, this will also bring more energy expenditures. Therefore, learning dynamic sampling frequency that finds a balanced performance between accuracy and cost is important for HAR on resource constrained mobile platforms.

#### 4.2 Evaluating the Performance of DWFS

We compare the performance of DWFS to the 6 baselines regarding the Error-Cost Index shown in Eq. 3. We test the methods with various settings of  $\lambda$  from 0.0 to 1.0. For easy comparison, the final testing results of the Error-Cost Indexes are multiplied by  $\frac{100}{N}$ . The results on datasets: HASC, HARSD, and DSA, are shown in Table 2. The proposed DWFS outperforms than other methods for most of the settings of  $\lambda$  from 0.1 to 1.0. For  $\lambda = 0$ , DWFS fails to select the best frequency, i.e., the highest frequency. Since the energy cost is not considered when  $\lambda = 0$ , this is out of the scope of our problem. To statistically compare the performance of DWFS with the baselines, we conduct the Wilcoxon signed-ranked test on the results of the 3 datasets. The returned

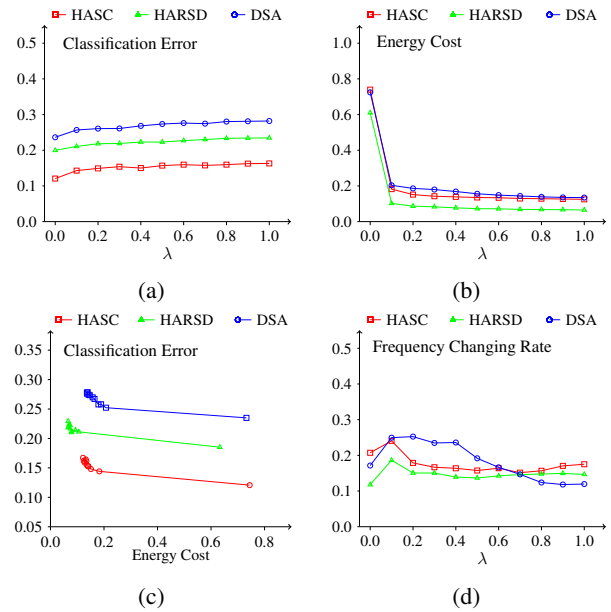


Figure 2: The 4 relationships on HASC, HARSD, and DSA datasets. (a) Classification error with respect to  $\lambda$ . (b) Energy cost with respect to  $\lambda$ . (c) Classification error with respect to energy cost. (d) Frequency changing rate with respect to  $\lambda$ .

$R^+$  and  $R^-$  correspond to the sum of the ranks of the differences above and below zero, respectively. The returned  $p$ -value represents the lowest level of significance of a hypothesis that results in rejection. This value allows one to determine whether two methods have significantly different performances. According to the results shown in Table 3, the  $p$ -values in comparisons of DWFS with Random, MDP-DS, RNN, and DWFS-CVL, reject the null hypotheses for the Error-Cost Index with a level of significance of  $\alpha = 0.05$  on all the datasets. The  $p$ -values in comparisons of DWFS with DWFS-SL, reject the null hypotheses for the Error-Cost Index with a level of significance of  $\alpha = 0.05$  on HARSD and DSA datasets.

#### 4.3 Insight Study of DWFS

We further investigate DWFS regarding 4 relationships: 1) classification error and  $\lambda$ ; 2) energy cost and  $\lambda$ ; 3) classification error and energy cost; 4) frequency changing rate and  $\lambda$ . The frequency changing rate is defined as  $\frac{\#\{\text{frequency changes}\}}{N}$ , which can be used to reveal the effectiveness of DWFS. According to the results shown in Figure 2a, 2b, and 2c, the classification error continuously increases and the energy cost continuously decreases with respect to  $\lambda$ , due to the imposed penalty on the energy cost causing sampling frequencies are dynamically reduced. According to the results shown in Figure 2d, there is no significant pattern between frequency changing rate and  $\lambda$ , but we can observe that DWFS maintains an approximately stable changing rate between 0.1 and 0.3, which demonstrates the robustness of DWFS in terms of  $\lambda$ .

Table 2: The Error-Cost Index (i.e.,  $\frac{100}{N} \sum_{t=1}^N 1\{\hat{y}^{(t)} \neq y^{(t)}\} + \lambda c_{f(t)}$ ) with respect to  $\lambda$  on HASC, HARSD, and DSA datasets.

HASC											
Method	$\lambda = 0.0$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.3$	$\lambda = 0.4$	$\lambda = 0.5$	$\lambda = 0.6$	$\lambda = 0.7$	$\lambda = 0.8$	$\lambda = 0.9$	$\lambda = 1.0$
100Hz	<b>11.21</b> (±1.84)	21.21(±1.84)	31.21(±1.84)	41.21(±1.84)	51.21(±1.84)	61.21(±1.84)	71.21(±1.84)	81.21(±1.84)	91.21(±1.84)	101.21(±1.84)	111.21(±1.84)
50Hz	11.80(±2.10)	16.80(±2.10)	21.80(±2.10)	26.80(±2.10)	31.80(±2.10)	36.80(±2.10)	41.80(±2.10)	46.80(±2.10)	51.80(±2.10)	56.80(±2.10)	61.80(±2.10)
16Hz	15.08(±2.93)	16.68(±2.93)	18.28(±2.93)	19.88(±2.93)	21.48(±2.93)	23.08(±2.93)	24.68(±2.93)	26.28(±2.93)	27.88(±2.93)	29.48(±2.93)	31.08(±2.93)
5Hz	28.54(±2.78)	29.04(±2.78)	29.54(±2.78)	30.04(±2.78)	30.54(±2.78)	31.04(±2.78)	31.54(±2.78)	32.04(±2.78)	32.54(±2.78)	33.04(±2.78)	33.54(±2.78)
Random	16.54(±2.15)	21.14(±2.14)	25.74(±2.12)	30.34(±2.11)	34.94(±2.09)	39.54(±2.08)	44.14(±2.06)	48.74(±2.05)	53.34(±2.04)	57.94(±2.03)	62.54(±2.02)
MDP-DS	13.88(±2.26)	18.85(±3.25)	21.83(±4.07)	23.52(±3.66)	25.14(±3.84)	26.40(±3.49)	27.91(±3.92)	29.39(±3.90)	30.81(±3.92)	32.09(±3.86)	33.23(±3.88)
RNN	12.01(±2.32)	16.74(±3.42)	18.39(±2.96)	20.01(±2.47)	21.32(±3.07)	23.12(±3.42)	24.81(±3.62)	26.38(±3.08)	27.47(±3.35)	28.32(±3.45)	29.87(±2.78)
DWFS-CVL	12.22(±1.71)	18.12(±3.04)	19.88(±3.31)	21.16(±3.07)	22.73(±2.87)	23.72(±2.66)	25.61(±2.34)	27.01(±2.44)	28.33(±1.84)	29.00(±2.20)	30.45(±2.03)
DWFS-SL	11.54(±2.21)	16.30(±2.58)	18.31(±3.03)	<b>19.29</b> (±2.94)	21.08(±3.45)	22.51(±3.63)	<b>23.56</b> (±3.81)	26.57(±4.06)	27.74(±3.39)	29.01(±3.41)	30.45(±2.03)
DWFS	12.07(±2.91)	<b>16.24</b> (±2.67)	<b>17.83</b> (±3.01)	19.54(±2.67)	<b>20.87</b> (±3.69)	<b>22.27</b> (±3.44)	24.17(±3.70)	25.67(±3.10)	<b>26.10</b> (±3.26)	<b>27.38</b> (±3.35)	<b>28.87</b> (±3.12)

HARSD											
Method	$\lambda = 0.0$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.3$	$\lambda = 0.4$	$\lambda = 0.5$	$\lambda = 0.6$	$\lambda = 0.7$	$\lambda = 0.8$	$\lambda = 0.9$	$\lambda = 1.0$
50Hz	18.41(±2.21)	28.41(±2.21)	38.41(±2.21)	48.41(±2.21)	58.41(±2.21)	68.41(±2.21)	78.41(±2.21)	88.41(±2.21)	98.41(±2.21)	108.41(±2.21)	118.41(±2.21)
16Hz	<b>18.35</b> (±2.32)	21.55(±2.32)	24.75(±2.32)	27.95(±2.32)	31.15(±2.32)	34.35(±2.32)	37.55(±2.32)	40.75(±2.32)	43.95(±2.32)	47.15(±2.32)	50.35(±2.32)
5Hz	21.67(±2.56)	22.67(±2.56)	23.67(±2.56)	24.67(±2.56)	25.67(±2.56)	26.67(±2.56)	27.67(±2.56)	28.67(±2.56)	29.67(±2.56)	30.67(±2.56)	31.67(±2.56)
2Hz	26.30(±2.82)	26.70(±2.82)	27.10(±2.82)	27.50(±2.82)	27.90(±2.82)	28.30(±2.82)	28.70(±2.82)	29.10(±2.82)	29.50(±2.82)	29.90(±2.82)	30.30(±2.82)
Random	20.16(±2.93)	24.12(±2.93)	28.09(±2.92)	32.05(±2.92)	36.01(±2.92)	39.97(±2.92)	43.93(±2.92)	47.90(±2.92)	51.86(±2.92)	55.82(±2.92)	59.78(±2.92)
MDP-DS	19.91(±2.16)	22.85(±2.10)	24.45(±2.06)	25.17(±2.18)	26.06(±2.52)	26.87(±2.48)	27.57(±2.60)	28.44(±2.78)	29.27(±2.86)	30.07(±2.87)	31.19(±2.80)
RNN	18.97(±2.41)	22.77(±2.22)	24.45(±2.51)	25.23(±2.71)	25.48(±3.06)	25.99(±2.94)	26.70(±2.76)	27.68(±2.59)	28.73(±3.37)	29.58(±2.61)	30.57(±2.75)
DWFS-CVL	18.86(±2.49)	<b>21.50</b> (±1.91)	23.34(±2.09)	25.15(±2.85)	25.66(±3.13)	26.56(±2.99)	27.71(±3.09)	28.43(±3.07)	28.67(±2.75)	29.15(±3.05)	30.57(±3.01)
DWFS-SL	19.33(±2.47)	22.60(±2.71)	23.44(±2.40)	24.99(±2.66)	25.86(±2.58)	26.44(±2.49)	27.01(±2.66)	27.85(±2.61)	28.64(±2.49)	29.05(±2.31)	30.00(±3.03)
DWFS	18.52(±1.76)	22.19(±2.45)	<b>23.29</b> (±1.93)	<b>23.41</b> (±2.10)	<b>24.33</b> (±2.95)	<b>25.37</b> (±2.26)	<b>26.67</b> (±3.11)	<b>27.20</b> (±2.94)	<b>27.19</b> (±2.37)	<b>27.95</b> (±2.58)	<b>29.53</b> (±2.49)

DSA											
Method	$\lambda = 0.0$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.3$	$\lambda = 0.4$	$\lambda = 0.5$	$\lambda = 0.6$	$\lambda = 0.7$	$\lambda = 0.8$	$\lambda = 0.9$	$\lambda = 1.0$
25Hz	<b>22.67</b> (±3.63)	32.67(±3.63)	42.67(±3.63)	52.67(±3.63)	62.67(±3.63)	72.67(±3.63)	82.67(±3.63)	92.67(±3.63)	102.67(±3.63)	112.67(±3.63)	122.67(±3.63)
10Hz	23.82(±4.33)	27.82(±4.33)	31.82(±4.33)	35.82(±4.33)	39.82(±4.33)	43.82(±4.33)	47.82(±4.33)	51.82(±4.33)	55.82(±4.33)	59.82(±4.33)	63.82(±4.33)
5Hz	27.51(±3.39)	29.51(±3.39)	31.51(±3.39)	33.51(±3.39)	35.51(±3.39)	37.51(±3.39)	39.51(±3.39)	41.51(±3.39)	43.51(±3.39)	45.51(±3.39)	47.51(±3.39)
2Hz	36.43(±3.79)	37.23(±3.79)	38.03(±3.79)	38.83(±3.79)	39.63(±3.79)	40.43(±3.79)	41.23(±3.79)	42.03(±3.79)	42.83(±3.79)	43.63(±3.79)	44.43(±3.79)
Random	30.26(±3.46)	34.55(±3.46)	38.85(±3.46)	43.14(±3.46)	47.43(±3.46)	51.72(±3.46)	56.01(±3.46)	60.30(±3.46)	64.60(±3.46)	68.89(±3.46)	73.18(±3.46)
MDP-DS	24.78(±3.85)	29.19(±3.86)	31.72(±3.94)	33.97(±3.72)	35.98(±3.62)	37.91(±3.56)	39.79(±3.56)	41.46(±3.51)	43.00(±3.51)	44.54(±3.45)	45.86(±3.48)
RNN	23.26(±3.60)	29.04(±3.81)	31.15(±3.61)	32.53(±3.97)	33.93(±3.63)	35.29(±3.89)	36.58(±3.45)	38.27(±3.99)	39.40(±3.40)	40.53(±3.37)	42.19(±4.00)
DWFS-CVL	26.04(±3.84)	29.94(±3.21)	32.16(±2.97)	34.05(±3.06)	35.89(±3.15)	37.35(±3.42)	38.79(±3.29)	40.25(±3.50)	41.82(±3.37)	42.70(±3.29)	44.21(±3.49)
DWFS-SL	23.53(±3.88)	27.66(±3.69)	29.78(±3.63)	31.74(±3.38)	33.45(±3.52)	34.89(±3.39)	36.54(±3.42)	37.84(±3.59)	39.04(±3.46)	40.18(±3.50)	41.64(±3.45)
DWFS	23.50(±3.77)	<b>27.30</b> (±3.25)	<b>29.58</b> (±3.06)	<b>31.15</b> (±3.29)	<b>33.29</b> (±2.78)	<b>34.87</b> (±2.90)	<b>36.14</b> (±3.23)	<b>37.40</b> (±3.28)	<b>38.96</b> (±3.26)	<b>39.92</b> (±3.44)	<b>41.41</b> (±3.30)

Table 3: The Wilcoxon test to compare the Error-Cost Indexes of DWFS, DWFS-CVL, DWFS-SL, RNN, MDP-DS, and Random regarding  $R^+$ ,  $R^-$ , and  $p$ -values.

Dataset	DWFS vs. Random			DWFS vs. MDP-DS			DWFS vs. RNN			DWFS vs. MDP-CVL			DWFS vs. DWFS-SL		
	$R^+$	$R^-$	$p$ -value	$R^+$	$R^-$	$p$ -value	$R^+$	$R^-$	$p$ -value	$R^+$	$R^-$	$p$ -value	$R^+$	$R^-$	$p$ -value
HASC	<b>0.0</b>	1540.0	0.000000	<b>2.0</b>	1538.0	0.000000	<b>288.0</b>	1252.0	0.000054	<b>112.0</b>	1428.0	0.000000	744.5	<b>740.5</b>	0.986261
HARSD	<b>9.0</b>	1531.0	0.000000	<b>110.0</b>	1430.0	0.000000	<b>398.0</b>	1142.0	0.001828	<b>323.0</b>	1162.0	0.000304	<b>212.0</b>	1273.0	0.000005
DSA	<b>0.0</b>	1540.0	0.000000	<b>1.0</b>	1539.0	0.000000	<b>166.0</b>	1374.0	0.000000	<b>0.0</b>	1540.0	0.000000	<b>437.0</b>	1103.0	0.005270

## 5 Conclusion & Future Work

In this paper, we address an emergent problem of Human Activity Recognition (HAR) on mobile/wearable platforms, which is adaptively determining sampling frequencies to balance recognition accuracy and energy efficiency. We formalize the problem as minimizing an objective function regarding classification error and energy cost, by finding an optimal classification model and dynamically appropriate sampling rates. We propose Datum-Wise Frequency Selection (DWFS) to solve the problem via a continuous state Markov Decision Process (MDP). The MDP learns a policy function that selects the best frequency for sampling an incoming data entity by exploiting the information of previously sampled instances. We propose an alternative learn-

ing method, where the parameters of the classification model and the policy function are mutually enhanced. We evaluate the performance of DWFS on 3 real-world HAR datasets, and the results show that DWFS statistically outperforms the state-of-the-arts regarding a combined measurement of classification error and energy cost.

In future work, we will investigate using imitation learning (He, Daume III, and Eisner 2012) as an auxiliary model for the selection of sampling frequencies, since imitation learning can handle much complex sensing tasks where the reward function regarding accuracy and energy cost cannot be explicitly defined. We will also test our model on various smart devices to evaluate the performance in real-world scenarios.

## Acknowledgment

This project is partially funded by ARC Discovery projects DP180103332 and DP180102050.

## References

- Anguita, D.; Ghio, A.; Oneto, L.; Parra, X.; and Reyes-Ortiz, J. L. 2013. A public domain dataset for human activity recognition using smartphones. In *Proceedings of ESANN*.
- Barshan, B., and Yükses, M. C. 2013. Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units. *The Computer Journal* 57(11):1649–1667.
- Dulac-Arnold, G.; Denoyer, L.; Preux, P.; and Gallinari, P. 2011. Datum-wise classification: A sequential approach to sparsity. In *Proceedings of ECML/PKDD*, 375–390.
- Fang, M.; Li, Y.; and Cohn, T. 2017. Learning how to active learn: A deep reinforcement learning approach. In *Proceedings of EMNLP*, 606–616.
- Gilbert, H.; Spanjaard, O.; Viappiani, P.; and Weng, P. 2015. Solving mdps with skew symmetric bilinear utility functions. In *Proceedings of IJCAI*, 1989–1995.
- Hammerla, N. Y.; Halloran, S.; and Plötz, T. 2016. Deep, convolutional, and recurrent models for human activity recognition using wearables. In *Proceedings of IJCAI*, 1533–1540.
- He, H.; Daume III, H.; and Eisner, J. 2012. Imitation learning by coaching. In *Proceedings of NIPS*, 3149–3157.
- Hoey, J.; Poupart, P.; von Bertoldi, A.; Craig, T.; Boutilier, C.; and Mihailidis, A. 2010. Automated handwashing assistance for persons with dementia using video and a partially observable markov decision process. *Computer Vision and Image Understanding* 114(5):503–519.
- Kang, S.; Lee, J.; Jang, H.; Lee, H.; Lee, Y.; Park, S.; Park, T.; and Song, J. 2008. Seemon: Scalable and energy-efficient context monitoring framework for sensor-rich mobile environments. In *Proceedings of MobiSys*, 267–280.
- Kawaguchi, N.; Yang, Y.; Yang, T.; Ogawa, N.; Iwasaki, Y.; Kaji, K.; Terada, T.; Murao, K.; Inoue, S.; Kawahara, Y.; Sumi, Y.; and Nishio, N. 2011. Hasc2011corpus: Towards the common ground of human activity recognition. In *Proceedings of UbiComp*, 571–572.
- Khan, A.; Hammerla, N.; Mellor, S.; and Plötz, T. 2016. Optimising sampling rates for accelerometer-based human activity recognition. *Pattern Recognition Letters* 73:33–40.
- Krause, A.; Ihmig, M.; Rankin, E.; Leong, D.; Gupta, S.; Siewiorek, D.; Smailagic, A.; Deisher, M.; and Sengupta, U. 2005. Trading off prediction accuracy and power consumption for context-aware wearable computing. In *Proceedings of ISWC*, 20–26.
- Liu, Y.; Nie, L.; Han, L.; Zhang, L.; and Rosenblum, D. S. 2015. Action2activity: Recognizing complex activities from sensor data. In *Proceedings of IJCAI*, 1617–1623.
- Liu, L.; Cheng, L.; Liu, Y.; Jia, Y.; and Rosenblum, D. S. 2016. Recognizing complex activities by a probabilistic interval-based model. In *Proceedings of AAAI*, volume 30, 1266–1272.
- Lockhart, J. W.; Pulickal, T.; and Weiss, G. M. 2012. Applications of mobile activity recognition. In *Proceedings of UbiComp*, 1054–1058.
- Lu, H.; Yang, J.; Liu, Z.; Lane, N. D.; Choudhury, T.; and Campbell, A. T. 2010. The jigsaw continuous sensing engine for mobile phone applications. In *Proceedings of SenSys*, 71–84.
- Plötz, T.; Hammerla, N. Y.; and Olivier, P. 2011. Feature learning for activity recognition in ubiquitous computing. In *Proceedings of IJCAI*, 1729–1734.
- Puterman, M. L. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.
- Qi, X.; Keally, M.; Zhou, G.; Li, Y.; and Ren, Z. 2013. Adasense: Adapting sampling rates for activity recognition in body sensor networks. In *Proceedings of RTAS*, 163–172.
- Ravi, N.; Dandekar, N.; Mysore, P.; and Littman, M. L. 2005. Activity recognition from accelerometer data. In *Proceedings of AAAI*, volume 5, 1541–1546.
- Trapeznikov, K., and Saligrama, V. 2013. Supervised sequential classification under budget constraints. In *Proceedings of AISTATS*, 581–589.
- Wang, Y.; Lin, J.; Annavaram, M.; Jacobson, Q. A.; Hong, J.; Krishnamachari, B.; and Sadeh, N. 2009. A framework of energy efficient mobile sensing for automatic user state recognition. In *Proceedings of MobiSys*, 179–192.
- Wang, Y.; Krishnamachari, B.; Zhao, Q.; and Annavaram, M. 2010. Markov-optimal sensing policy for user state estimation in mobile devices. In *Proceedings of IPSN*, 268–278.
- Yadav, A.; Chan, H.; Jiang, A.; Rice, E.; Kamar, E.; Grosz, B.; and Tambe, M. 2016. Pomdps for assisting homeless shelters—computational and deployment challenges. In *Proceedings of AAMAS*, 67–87. Springer.
- Yan, Z.; Subbaraju, V.; Chakraborty, D.; Misra, A.; and Aberer, K. 2012. Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach. In *Proceedings of ISWC*, 17–24.
- Yang, J. B.; Nguyen, M. N.; San, P. P.; Li, X. L.; and Krishnaswamy, S. 2015. Deep convolutional neural networks on multichannel time series for human activity recognition. In *Proceedings of IJCAI*, 3995–4001.
- Yurur, O.; Liu, C. H.; Perera, C.; Chen, M.; Liu, X.; and Moreno, W. 2015. Energy-efficient and context-aware smartphone sensor employment. *IEEE Transactions on Vehicular Technology* 64(9):4230–4244.
- Zhang, C., and Shah, J. A. 2014. Fairness in multi-agent sequential decision-making. In *Proceedings of NIPS*. 2636–2644.