

# Physical Activity Recognition from Accelerometer Data Using a Multi-Scale Ensemble Method

**Yonglei Zheng, Weng-Keen Wong, Xinze Guan**

School of EECS  
Oregon State University  
Corvallis, OR 97331

**Stewart Trost**

School of Human Movement Studies  
University of Queensland  
Brisbane, Queensland, Australia

## Abstract

Accurate and detailed measurement of an individual's physical activity is a key requirement for helping researchers understand the relationship between physical activity and health. Accelerometers have become the method of choice for measuring physical activity due to their small size, low cost, convenience and their ability to provide objective information about physical activity. However, interpreting accelerometer data once it has been collected can be challenging. In this work, we applied machine learning algorithms to the task of physical activity recognition from triaxial accelerometer data. We employed a simple but effective approach of dividing the accelerometer data into short non-overlapping windows, converting each window into a feature vector, and treating each feature vector as an i.i.d training instance for a supervised learning algorithm. In addition, we improved on this simple approach with a multi-scale ensemble method that did not need to commit to a single window size and was able to leverage the fact that physical activities produced time series with repetitive patterns and discriminative features for physical activity occurred at different temporal scales.

## Introduction

Although physical activity is well-known by the general public to be essential for maintaining a healthy body, researchers continue to seek a better understanding of the relationship between physical activity and health. A key requirement of this research is the accurate and detailed measurement of an individual's physical activity. Researchers can use this data to identify people at risk of certain diseases, evaluate the efficacy of intervention strategies for increasing physical activity and understand why some groups of people are more active than others (Bauman et al. 2006).

Self-reports have traditionally been the means of providing information about physical activity (de Vries et al. 2011). However, self-reports are susceptible to subjective factors, such as recall bias and social desirability, thus lacking accuracy. For instance, self-reports tend to overestimate the time spent in unstructured daily physical activities, such as walking (Tudor-Locke and Myers 2001).

Copyright © 2013, Association for the Advancement of Artificial Intelligence ([www.aaai.org](http://www.aaai.org)). All rights reserved.

One of the most promising alternatives to self-reports is the use of accelerometers (Trost, McIver, and Pate 2005), which are free from subjective biases. Accelerometers for physical activity monitoring capture acceleration in different planes, with triaxial accelerometers being one of the most common types. Figure 1 illustrates triaxial accelerometer data collected from seven different activity classes, with the data from each axis shown in a different color or style. Once this data has been collected, the challenge is to interpret this three dimensional time series. Interpreting this data requires two steps. First, if the accelerometer data were collected under free-living conditions, the time series needs to be divided into segments corresponding to one particular type of physical activity. Then, each smaller time series corresponding to a segment needs to be classified as a physical activity type. In our work, we focus on the classification task in the second step because our data has already been segmented by the nature of our data collection process. For future work, we will investigate algorithms for segmenting data obtained under free-living conditions.

The traditional approach in exercise science for classifying a time series into physical activity classes is to use regression-based thresholds called cut-points (Bassett Jr., Rowlands, and Trost 2012) which allow researchers to estimate the time spent performing physical activities at different intensity levels. Researchers, however, have found cut-points to be inaccurate (eg. (Staudenmayer et al. 2009; Trost et al. 2012)), and are turning to machine learning methods to identify physical activity types and estimate energy expenditure more accurately.

The basic machine learning task involves classifying a triaxial time series as a single physical activity type. Many approaches to time series classification have been proposed in the machine learning literature (Xing, Pei, and Keogh 2010) and choosing the right approach depends on the nature of the time series. Two key characteristics of physical activity data (such as the data in Figure 1) that we will leverage to improve classification accuracy are its repetitive pattern lasting for the duration of the physical activity and the fact that discriminative features occur at different temporal scales.

In this work, we apply machine learning techniques to the task of predicting physical activity type from data collected from a single body-mounted triaxial accelerometer. We show that a straightforward application of supervised

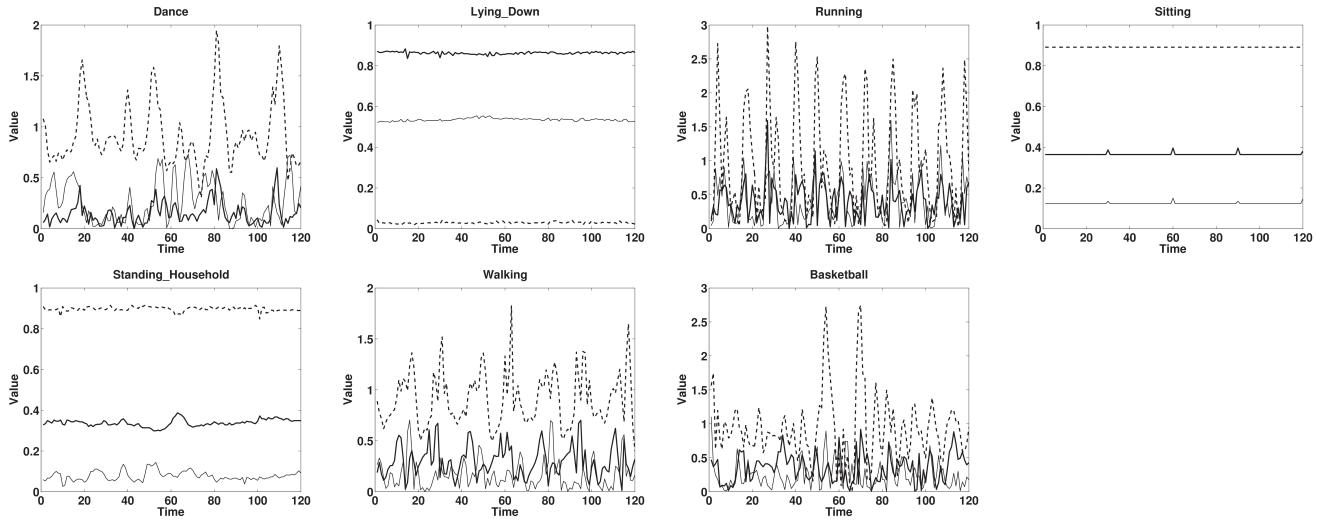


Figure 1: An example of 4 seconds of data from all seven classes in the OSU\_Hip dataset. These plots illustrate triaxial accelerometer data collected at 30 Hz, and plot line in a graph represents one axis. The activity types are, from left to right: (top) dancing, lying down, running, sitting, (bottom) standing and household chores, walking and basketball.

learning techniques can produce highly accurate results. To produce even more accurate results, we develop a new algorithm consisting of an ensemble of multi-scale classifiers in which each ensemble member is trained on a set of features computed from subwindows of different sizes from the original time series, thereby leveraging discriminative features from different temporal scales. We evaluated this ensemble method on three accelerometer data sets and found that it improved over standard supervised learning techniques.

## Related Work

Past work has explored activity recognition using data from a single accelerometer, typically placed at the waist or hip (Staudenmayer et al. 2009; Bonomi et al. 2009). A variety of standard supervised learning algorithms have been employed, but Ravi et al. (2005) conclude that ensemble methods, especially majority voting, were consistently among the top performing algorithms for activity recognition. An alternative approach is to use data from multiple accelerometers or other sensors for physical activity recognition (Bao and Intille 2004). Although multiple sensors can produce more accurate predictions, they are not practical as they would require an individual to wear multiple devices, which may be considered too cumbersome.

Our task specifically involves classifying a numeric time series as a single activity type. A variety of methods that accomplish this task (Xing, Pei, and Keogh 2010; Wang et al. 2010) do so by transforming the raw time series into a more efficient representation that preserves its key traits but reduces its dimensionality. Examples of these representations include a symbolic representation (Lin et al. 2003) or a feature vector of shapelets (Ye and Keogh 2009). Once a raw time series has been converted into an efficient representation, supervised learning techniques can be applied to it. One of the most frequently used methods is the k-nearest neighbor (k-NN) algorithm, which can be surprisingly accurate using Euclidean distance (Keogh and Kasetty 2003) or dynamic time warping (DTW) (Wang et al. 2010). These techniques work well when the task is to match the overall shape of a time series, but as we will show, they perform poorly on accelerometer data with repetitive patterns.

Hidden Markov Models (HMMs) have also been used for physical activity recognition (Lester et al. 2005). Since HMMs can model transitions between physical activities, they are more suitable for segmenting a time series into a sequence of physical activity types rather than classifying an entire time series as a single activity type.

## Methodology

A simple and effective approach to the classification task is to cut the time series up into non-overlapping windows of some size  $W$ . Then, each window can be converted into a feature vector and each feature vector treated as if it were an independent, identically distributed (i.i.d.) data instance. At this point, supervised learning algorithms can be applied to each feature vector. This approach works especially well for time series with repetitive patterns, provided each window contains at least one “cycle” of the repetitive pattern. Indeed, this approach has been shown to be effective for physical activity recognition by several researchers, who have all used a variety of machine learning techniques such as neural nets (Staudenmayer et al. 2009), decision trees (Bonomi et al. 2009) and support vector machines (Su et al. 2005).

Treating each window as an i.i.d data instance requires addressing two key issues. The first important issue is how to convert raw accelerometer data into a feature vector. Features need to capture important aspects of the data that can discriminate between different activity types and be applicable for different subjects. Knowing what these features will be beforehand is very difficult. Therefore, one approach

for engineering features is to follow the techniques used in computer vision and create a bag of features (Zhang and Sawchuk 2012), which generates a large set of potentially useful features. In our work, we found that the bag of features approach produced accurate predictions if the algorithm can handle a large number of features by guarding against overfitting (eg. through regularization of its parameters). Furthermore, since the ultimate goal is to deploy physical activity recognition algorithms in real time, the step of converting raw accelerometer data to a feature vector needs to be efficient. We use a large number of coarse-grained summary statistics as features that can all be computed in linear time from a window of data (Table 1).

The second issue involves determining the window size  $W$ , which is seldom addressed even though it affects the accuracy of the algorithm (Trost et al. 2012). The larger the window size, the longer it takes to identify the activity. For instance, if the window size were 60 seconds, then one must wait for a full 60 seconds worth of data to be accumulated before the prediction can take place. On the other hand, the smaller the window size, the less information is available to make an informed prediction of the activity type. If the features are summary statistics over the window (eg. the mean), then the window size also has an effect on the quality of the computed features. Having the wrong window size can cause the feature representation to oversmooth/undersmooth aspects of the data that are needed to discriminate between physical activity types.

## Algorithm

We now describe the Subwindow Ensemble Model (SWEM), which is designed to leverage a key aspect of physical activity recognition – that physical activity types have discriminative features at different temporal scales. The SWEM consists of an ensemble of classifiers, with each classifier trained on a different feature representation of the data. Each feature representation corresponds to a set of features generated from different temporal scales of the time series. With this approach, we avoid committing to one particular window size at the expense of having to perform feature generation and classifier training for as many times as we have ensemble members. At the end, the predictions by each ensemble member are combined via majority vote to produce an overall prediction for a time series.

Algorithm 1 provides pseudocode for how the ensemble members in the SWEM are trained. The SWEMMEMBERTRAIN function accepts as input labeled time series data and a list of subwindow sizes. If we use Figure 2 as a running example, we have  $L = 1, 5, 10$ , corresponding to subwindows of 1, 5 and 10 seconds length. The for loop in line 2 iterates over the subwindow sizes. In lines 4-10, we decompose each time series  $t$  into overlapping subwindows of size  $l$ ; the overlap occurs because each subwindow of size  $l$  is shifted over by 1 second, as in Figure 2. Line 5 retrieves the class label of time series  $t$ . Lines 6-9 takes each subwindow of size  $l$ , converts it to a feature vector with the FEATURIZE function, and adds the feature vector with the class label to the training data. The BUILD-MODEL function in Line 11

Features
1. Sum of values of a period of time: $\sum_{i=1}^T s_i$ .
2. Mean: $\mu_s = \frac{1}{T} \sum_{i=1}^T s_i$ .
3. Standard deviation: $\sigma_s = \sqrt{\frac{1}{T} \sum_{i=1}^T (s_i - \mu_s)^2}$ .
4. Coefficients of variation: $c_v = \frac{\sigma_s}{\mu_s}$ .
5. Peak-to-peak amplitude: $\max\{s_1, \dots, s_T\} - \min\{s_1, \dots, s_T\}$ .
6-10. Percentiles: $10^{th}, 25^{th}, 50^{th}, 75^{th}, 90^{th}$
11. Interquartile range: difference between the $75^{th}$ and $25^{th}$ percentiles.
12. Lag-one-autocorrelation: $\frac{\sum_{i=1}^{T-1} (s_i - \mu_s)(s_{i+1} - \mu_s)}{\sum_{i=1}^T (s_i - \mu_s)^2}$ .
13. Skewness: $\frac{\frac{1}{T} \sum_{i=1}^T (s_i - \mu_s)^3}{(\frac{1}{T} \sum_{i=1}^T (s_i - \mu_s)^2)^{\frac{3}{2}}}$ , measure of asymmetry of the signal probability distribution.
14. Kurtosis: $\frac{\frac{1}{T} \sum_{i=1}^T (s_i - \mu_s)^4}{(\frac{1}{T} \sum_{i=1}^T (s_i - \mu_s)^2)^3} - 3$ , degree of the peakedness of the signal probability distribution.
15. Signal power: $\sum_{i=1}^T s_i^2$ .
16. Log-energy: $\sum_{i=1}^T \log(s_i^2)$ .
17. Peak intensity: number of signal peak appearances within a certain period of time.
18. Zero crossings: number of times the signal crosses its median.
19. Correlation between each pair of axes: $\frac{\sum_{i=1}^T (s_i - \mu_s)(v_i - \mu_v)}{\sqrt{\sum_{i=1}^T (s_i - \mu_s) \sum_{j=1}^T (v_j - \mu_v)}}$ .

Table 1: Time series features used in our representation of each window where the time series is denoted as  $s_1, \dots, s_T$  and  $T$  is the length of the window

trains the ensemble member with the training data and adds the trained ensemble member to the set  $M$ .

Algorithm 2 describes how SWEM classifies a new time series. In lines 2-9, the SWEM-PREDICT function collects the predictions by an ensemble member over subwindows of size  $l$ . In line 3, we retrieve the length  $l$  of the subwindow associated with ensemble member  $m$ . The for loop in lines 5-7 converts all subwindows of length  $l$  in  $T$  into a feature vector, predicts the class of each subwindow feature vector, and adds the predictions to  $p$ . In line 8, the prediction by a single ensemble member  $m$  for the entire time series  $t$  is obtained by a majority vote over  $p$ , which stores the predicted class labels for each subwindow. Finally, in line 10, the overall prediction by the ensemble is produced by a majority vote over the predictions by each ensemble member.

The SWEM algorithm trades off speed for accuracy. If there are  $|M|$  ensemble members in SWEM, then the process of training an ensemble member and predicting with it needs to be repeated  $|M|$  times during the SWEM algorithm. Ensemble members dealing with smaller subwindow sizes will take longer to run because there are more small subwindows within a time series than large subwindows. Fortunately, each ensemble member can be handled independently of the others during training and prediction, allowing us to exploit parallel processing and thereby making

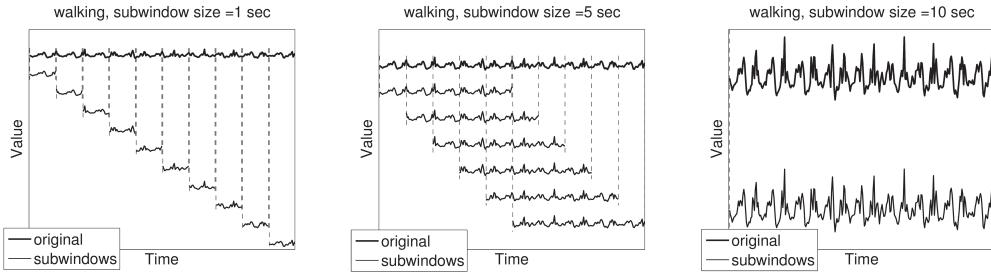


Figure 2: Decomposing a time series of a 10-second walk (shown on top) into 1, 5 and 10-second overlapping subwindows (shown on the bottom). The subwindows shift by 1 second.

---

**Algorithm 1 SWEMMEMBERTRAIN( $T, L$ )**


---

**Input**

- 1:  $T$ : dataset of labeled time series for training.
- 2:  $L$ : subwindow sizes.

**Output**

- 1:  $M$ : the ensemble members.

**Procedure**

```

1:  $M \leftarrow \{\}$ 
2: for each  $l$  in  $L$  do
3:    $training\_data \leftarrow \{\}$ 
4:   for each time series  $t$  in  $T$  do
5:      $label \leftarrow CLASS-LABEL(t)$ 
6:     for each subwindow  $s$  of  $t$  with length  $l$  do
7:        $x \leftarrow FEATURIZE(s)$ 
8:        $training\_data \leftarrow training\_data \cup (x, label)$ 
9:     end for
10:   end for
11:    $M \leftarrow M \cup BUILD-MODEL(training\_data)$ 
12: end for
13: return  $M$ 
```

---

the SWEM algorithm efficient in practice.

## Evaluation

### Datasets

The SWEM was evaluated on three accelerometer-based physical activity datasets – two datasets were from Oregon State University (OSU) and the third was from the Human Activity Sensing Consortium (HASC).

The OSU datasets contained data recorded by triaxial accelerometers at a 30 Hz sampling rate. Participants of ages 5-15 were asked to perform seven different types of physical activities for 2 minutes each in a controlled lab environment. The seven activity classes included: lying down, sitting, standing and household chores, walking, running, basketball and dance. The OSU\_Wrist dataset contains data collected from 18 participants with wrist-mounted accelerometers while the OSU\_Hip dataset contained data from 53 participants with hip-mounted accelerometers. Figure 1 shows an example of all seven classes in the OSU\_Hip dataset. Data from the OSU\_Wrist dataset is similar and not shown due to space limitations. The two minutes of data were cut up

---

**Algorithm 2 SWEMPREDICT( $M, t$ )**


---

**Input**

- 1:  $M$ : the ensemble members generated by SWEMMEMBERTRAIN.
- 2:  $t$ : the time series to be predicted.

**Output**

- 1: *prediction*: the prediction of time series  $t$ .

**Procedure**

```

1:  $v \leftarrow \{\}$ 
2: for each ensemble member  $m$  in  $M$  do
3:    $l \leftarrow SUBWINDOWLENGTH(m),$ 
4:    $p \leftarrow \{\}$ 
5:   for each subwindow  $s$  of  $t$  with length  $l$  do
6:      $p \leftarrow p \cup PREDICT(m, FEATURIZE(s))$ 
7:   end for
8:    $v[m] \leftarrow MAJORITYVOTE(p)$ 
9: end for
10:  $prediction \leftarrow MAJORITYVOTE(v)$ 
11: return  $prediction$ 
```

---

into 10-second time windows. Thus, the classification task involves classifying each 10-second window as an activity type. In past work, Trost et al. (2012) found that 10-second time windows were a good compromise between collecting enough data to predict the activity class and having a fast enough detection time.

The third dataset used in our experiments consists of the “Sample Data” from the Human Activity Sensing Consortium (HASC) 2011 challenge<sup>1</sup> (Kawaguchi et al. 2011). The data were collected from seven subjects with triaxial accelerometers at a 100 Hz sampling rate. Six activities, namely stay, walk, jog, skip, stUp (stair-up) and stDown (stair-down), were performed by all subjects in a controlled lab environment. As with the OSU data, we divide the HASC time series data into windows of length 10 seconds.

## Experiments

The SWEM consisted of 10 ensemble members, with each ensemble member corresponding to a subwindow of length 1, 2, 3, etc. up to 10 seconds. Features 1-18 were extracted from each axis, and Feature 19 (the correlation between

<sup>1</sup> Available at <http://hasc.jp/hc2011/download-en.html>

axes) was extracted from each pair of axes, resulting in a total of 57 features. The ensemble members for the SWEM were linear support vector machines, implemented using LibSVM (Chang and Lin 2011).

The dataset was randomly split by subject into three non-overlapping subsets for training, validation and testing. Each algorithm in our experiments was trained on the training set, each trained model was tuned on the validation set, and parameter settings achieving the highest performance on the validation set were chosen as the parameters for the final model evaluated on the test set. For the SVMs used in SWEM, the  $C$  parameter was tuned over values 0.01, 0.1, 1, 10, 100 and 1000. Each algorithm was evaluated using 30 training-validation-testing splits and the average macro-F1 was reported.

We compared the performance of SWEM against the following algorithms. First, 1-nearest neighbor is a commonly used baseline for time series classification algorithms. We applied a 1-nearest neighbor algorithm on the raw time series with both a Euclidean distance metric (1NN\_EUC) and with Dynamic Time Warping (1NN\_DTW). In addition, we applied a 1-nearest neighbor algorithm on the SAX representation of the time series with both a Euclidean distance metric (1NN\_EUC\_SAX) and with Dynamic Time Warping (1NN\_DTW\_SAX). For SAX, we tuned the number of segments (10, 50, 100) and the number of symbols (5, 10, 20) but found that the results did not change much. Finally, we also applied a linear SVM, implemented with LibSVM (Chang and Lin 2011), with the  $C$  parameter tuned on the validation set. We chose an SVM because it was one of the best performing algorithms on the physical activity data as compared to other supervised learning techniques. Since artificial neural networks (ANNs) are commonly used in the exercise science literature, we include results from a feed-forward neural network with a single hidden layer. The *nnet* (Venables and Ripley 2002) package in R was used as the ANN implementation in our experiment. We tuned the number of hidden units (1-30) and decay weights (0, 0.5, and 1). We also attempted to represent the data with shapelets (Ye and Keogh 2009), but the training phase of the shapelet algorithm, which is computationally expensive, did not finish running on our full dataset.

## Discussion

Table 2 illustrates the results of the experiments. The SWEM\_SVM was the best performing model on the OSU\_Hip (0.942), OSU\_Wrist (0.896) and HASC (0.820) datasets. SVMs and ANNs performed reasonably well, with SVMs being superior to ANNs, especially on the OSU\_Wrist dataset. Although SWEM\_SVM resulted in a slight improvement over SVM, both SWMs and ANNs were provided with an informed value of  $W = 10$ , which helped their performance. In general, finding an appropriate value for this window size is difficult to do. The SWEM\_SVM algorithm, in contrast, removes the need to commit to a particular window size and it can exploit features of the data from different subwindow sizes. The nearest neighbor methods performed poorly because they tried to match the overall shape of the time series. In addition, the SAX representation

Algorithm	OSU_Hip	OSU_Wrist	HASC
SWEM_SVM	<b>0.942†</b>	<b>0.896†</b>	<b>0.820†</b>
SVM	0.937	0.886	0.794
ANN	0.919	0.787	0.738
1NN_EUC	0.572	0.456	0.648
1NN_DTW	0.561	0.494	0.648
1NN_EUC_SAX	0.142	0.147	0.169
1NN_DTW_SAX	0.143	0.143	0.169

Table 2: Average macro-F1 scores of the various algorithms on the three datasets. The bold font marks the model with the highest average macro-F1 and the symbol † indicates that the improvement by SWEM is statistically significant (Wilcoxon signed rank test, p-value < 0.05) above all the other algorithms.

also performed poorly because the aggregation caused many of the discriminative details to be smoothed out.

We can gain further insight into the benefits of the multi-scale approach of SWEM\_SVM by comparing its performance to that of its individual ensemble members by removing the meta-layer performing a majority vote. In doing so, we report results as if we had made a prediction by using only a single temporal scale. Tables 3 to 5 compare the classification accuracies of SWEM\_SVM against each individual ensemble member on the OSU\_Hip, OSU\_Wrist and HASC datasets. We refer to each ensemble member as SWEM\_SVM followed by the size of the subwindow eg. SWEM\_SVM1 for a 1 second subwindow. Note that SWEM\_SVM10 is the largest possible subwindow size and these results are identical to applying an SVM to each window of data (ie. the results in Table 2).

Our results show that ensemble members of different subwindow sizes performed better for certain activities than others. These results confirmed our hypothesis that discriminative features existed at different temporal scales for the various activity types. For example, on the OSU\_Hip dataset, subwindows of size 8 and 9 produced the best results for lying, sitting, and walking while subwindows of size 1 produced the best results for standing and running. Similarly, on the HASC dataset, the best results for walking were produced with a subwindow size of 9 while the smaller subwindow sizes produced more accurate predictions for stay, jog, skip, stUp and stDown. Some ensemble members performed better than SWEM\_SVM for specific activities, but over all activities, SWEM\_SVM always had a higher average accuracy than individual ensemble members.

Finally, we also experimented with using stacking (Wolpert 1992) instead of majority vote to combine the predictions made by individual ensemble members into an overall prediction. However, we found that stacking produced similar results to majority vote. An important advantage that majority vote has over stacking is that no additional training is needed for the meta-layer. In order to train a stacking algorithm properly, the meta-layer needs to be trained on training data that is separate from the data used to train the individual ensemble members. If training data is limited, creating a second training set may not be possible.

Model	Macro-F1	Classification Accuracy of Each Physical Activity						
		lying	sitting	standing	walking	running	basketball	dance
SWEM_SVM	0.9424	0.9806	0.9423	0.9678	0.9541	0.9823	0.9419	0.8041
SWEM_SVM1	0.9090	0.9709	0.9294	<b>0.9893</b>	0.9488	<b>0.9876</b>	0.7398	0.6931
SWEM_SVM2	0.9339	0.9735	0.9271	0.9836	0.9543	0.9844	0.8931	0.7648
SWEM_SVM3	0.9357	0.9719	0.9365	0.9727	0.9502	0.9870	0.9283	0.7756
SWEM_SVM4	0.9355	0.9800	0.9265	0.9709	0.9533	0.9810	0.9178	0.7861
SWEM_SVM5	0.9345	0.9780	0.9357	0.9564	0.9494	0.9811	<b>0.9407</b>	0.7931
SWEM_SVM6	0.9361	0.9787	0.9299	0.9609	0.9572	0.9798	0.9306	0.7911
SWEM_SVM7	<b>0.9373</b>	0.9802	0.9353	0.9519	0.9565	0.9798	0.9378	<b>0.8131</b>
SWEM_SVM8	0.9371	<b>0.9819</b>	0.9296	0.9608	<b>0.9615</b>	0.9776	0.9206	0.7991
SWEM_SVM9	0.9383	0.9817	<b>0.9374</b>	0.9572	0.9567	0.9789	0.9359	0.8104
SWEM_SVM10	0.9369	0.9772	0.9318	0.9666	0.9599	0.9776	0.9161	0.7978

Table 3: Macro-F1 and classification accuracies of the overall SWEM\_SVM algorithm and of each ensemble member on the OSU\_Hip dataset. The bold font marks the best performing subwindow size overall and for each activity.

Model	Macro-F1	Classification Accuracy of Each Physical Activity						
		lying	sitting	standing	walking	running	basketball	dance
SWEM_SVM	0.8961	0.7993	0.9522	0.9389	0.9562	0.8345	0.9072	0.7722
SWEM_SVM1	0.8538	0.7368	<b>0.9608</b>	<b>0.9767</b>	0.9639	<b>0.8565</b>	0.4300	0.7257
SWEM_SVM2	0.8808	0.7708	0.9571	0.9618	<b>0.9700</b>	0.8547	0.7206	0.7243
SWEM_SVM3	0.8902	0.7924	0.9481	0.9464	0.9583	0.8507	0.8461	0.7694
SWEM_SVM4	0.8897	0.7778	0.9559	0.9455	0.9563	0.8426	0.8828	0.7403
SWEM_SVM5	<b>0.8911</b>	<b>0.8021</b>	0.9438	0.9368	0.9514	0.8281	<b>0.9194</b>	<b>0.7701</b>
SWEM_SVM6	0.8903	0.7875	0.9549	0.9382	0.9534	0.8299	0.9056	0.7535
SWEM_SVM7	0.8889	0.8007	0.9549	0.9239	0.9522	0.8218	0.9150	0.7611
SWEM_SVM8	0.8887	0.7764	0.9605	0.9378	0.9567	0.8235	0.8933	0.7410
SWEM_SVM9	0.8872	0.7903	<b>0.9608</b>	0.9192	0.9518	0.8241	0.9050	0.7576
SWEM_SVM10	0.8859	0.7819	0.9590	0.9340	0.9575	0.8148	0.8822	0.7312

Table 4: Macro-F1 and activity classification accuracies of the overall SWEM\_SVM algorithm and of each ensemble member on the OSU\_Wrist dataset. The bold font marks the highest accuracy of a single subwindow model for each activity.

Model	Macro-F1	Classification Accuracy of Each Physical Activity					
		stay	walk	jog	skip	stUp	stDown
SWEM_SVM	0.8200	0.9956	0.7656	0.7989	0.8400	0.7111	0.7878
SWEM_SVM1	0.8134	<b>1.0000</b>	0.7456	<b>0.8122</b>	0.8267	0.6800	0.8033
SWEM_SVM2	<b>0.8173</b>	0.9989	0.7367	0.7956	<b>0.8456</b>	0.7067	<b>0.8067</b>
SWEM_SVM3	0.8151	<b>1.0000</b>	0.7389	0.8044	0.8178	<b>0.7211</b>	0.7922
SWEM_SVM4	0.8076	0.9989	0.7378	0.7911	0.8267	0.6978	0.7800
SWEM_SVM5	0.8056	0.9944	0.7489	0.8000	0.8244	0.7000	0.7489
SWEM_SVM6	0.8025	0.9933	0.7322	0.7889	0.8322	0.7000	0.7556
SWEM_SVM7	0.7934	0.9944	0.7389	0.8033	0.8156	0.6567	0.7367
SWEM_SVM8	0.7935	0.9867	0.7233	0.7878	0.8344	0.6689	0.7444
SWEM_SVM9	0.7996	0.9878	<b>0.7767</b>	0.8078	0.8111	0.6767	0.7167
SWEM_SVM10	0.7940	0.9811	0.7311	0.7944	0.8256	0.6689	0.7500

Table 5: Macro-F1 scores and activity classification accuracies of the overall SWEM\_SVM algorithm and of each ensemble member on the HASC dataset. The bold font marks the highest classification accuracy of a single subwindow model for each activity.

## Conclusion

We proposed the Subwindow Ensemble Model which used an ensemble of classifiers trained on features made up of coarse summary statistics computed from different temporal scales. The SWEM outperformed other baseline algorithms and it had the additional benefit of not needing to commit to a single window size  $W$ . The SWEM algorithm achieved very accurate results ( $\geq 90\%$  for OSU data,  $> 80\%$  for HASC), which suggests that the algorithm could be viable for deployment. For future work, we will investigate the challenge of deploying physical activity recognition algorithms in real time on free-living data. Since free-living data consists of a mixture of different activities performed throughout an individual's day, we will explore algorithms for segmenting the data and classifying these segments into physical activity types.

## Acknowledgements

This study was supported by funding by the Eunice Kennedy Shriver National Institute of Child Health and Human Development (NICHD R01 55400).

## References

- Bao, L., and Intille, S. S. 2004. Activity Recognition from User-Annotated Acceleration Data Pervasive Computing. *Pervasive Computing* 3001:1–17.
- Bassett Jr., D. R.; Rowlands, A.; and Trost, S. G. 2012. Calibration and validation of wearable monitors. *Med Sci Sports Exerc.* 1(1 Suppl 1):S32–38.
- Bauman, A.; Phongsavan, P.; Schoeppe, S.; and Owen, N. 2006. Physical activity measurement - a primer for health promotion. *IUHPE - Promotion and Education* 8:92–103.
- Bonomi, A.; Goris, A.; Yin, B.; and Westerterp, K. 2009. Detection of type, duration, and intensity of physical activity using an accelerometer. *Medicine & Science in Sports & Exercise* 41(9):1770.
- Chang, C., and Lin, C. 2011. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2(3):27.
- de Vries, S.; Garre, F.; Engbers, L.; Hildebrandt, V.; and van Buuren, S. 2011. Evaluation of neural networks to identify types of activity using accelerometers. *Medicine & Science in Sports & Exercise* 43(1):101.
- Kawaguchi, N.; Ogawa, N.; Iwasaki, Y.; Kaji, K.; Terada, T.; Murao, K.; Inoue, S.; Kawahara, Y.; Sumi, Y.; and Nishio, N. 2011. Hasc challenge: gathering large scale human activity corpus for the real-world activity understandings. In *Proceedings of the 2nd Augmented Human International Conference*, 27. ACM.
- Keogh, E., and Kasetty, S. 2003. On the need for time series data mining benchmarks: a survey and empirical demonstration. *Data Mining and Knowledge Discovery* 7(4):349–371.
- Lester, J.; Choudhury, T.; Kern, N.; Borriello, G.; and Hannaford, B. 2005. A hybrid discriminative/generative approach for modeling human activities. In *Proceedings of the 19th international joint conference on Artificial intelligence*, IJCAI'05, 766–772. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Lin, J.; Keogh, E.; Lonardi, S.; and Chiu, B. 2003. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, 2–11. ACM.
- Ravi, N.; Dandekar, N.; Mysore, P.; and Littman, M. L. 2005. Activity recognition from accelerometer data. In *Proceedings of the 17th conference on Innovative applications of artificial intelligence - Volume 3*, IAAI'05, 1541–1546. AAAI Press.
- Staudenmayer, J.; Pober, D.; Crouter, S.; Bassett, D.; and Freedson, P. 2009. An artificial neural network to estimate physical activity energy expenditure and identify physical activity type from an accelerometer. *Journal of Applied Physiology* 107(4):1300–1307.
- Su, S. W.; Wang, L.; Celler, B. G.; Ambikairajah, E.; and Savkin, A. V. 2005. Estimation of walking energy expenditure by using support vector regression. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, 3526–3529. IEEE.
- Trost, S.; Wong, W.; Pfeiffer, K.; and Zheng, Y. 2012. Artificial neural networks to predict activity type and energy expenditure in youth. *Medicine and science in sports and exercise*.
- Trost, S. G.; McIver, K. L.; and Pate, R. R. 2005. Conducting accelerometer-based activity assessments in field-based research. *Medicine and Science in Sports and Exercise* 37(11 suppl):S531–543.
- Tudor-Locke, C., and Myers, A. 2001. Challenges and opportunities for measuring physical activity in sedentary adults. *Sports Medicine* 31(2):91–100.
- Venables, W. N., and Ripley, B. D. 2002. *Modern Applied Statistics with S*. New York: Springer, fourth edition. ISBN 0-387-95457-0.
- Wang, X.; Mueen, A.; Ding, H.; Trajcevski, G.; Scheuermann, P.; and Keogh, E. 2010. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery* 1–35.
- Wolpert, D. 1992. Stacked generalization. *Neural Networks* 5(2):241–259.
- Xing, Z.; Pei, J.; and Keogh, E. 2010. A brief survey on sequence classification. *SIGKDD Explor. Newsl.* 12(1):40–48.
- Ye, L., and Keogh, E. 2009. Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 947–956. ACM.
- Zhang, M., and Sawchuk, A. 2012. Motion primitive-based human activity recognition using a bag-of-features approach. In *Proceedings of the 2nd ACM SIGHIT symposium on International health informatics*, 631–640. ACM.