A Multi-Scale Ensemble Method for Physical Activity Recognition from Accelerometer Data

Yonglei Zheng* Weng-Keen Wong* Stewart Trost† Xinze Guan*

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. The challenge, however, is in interpreting accelerometer data once it has been collected. In this work, we applied data mining algorithms to the problem of classifying a time series as being one of several possible physical activity types. 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 (SWEM) 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. We applied the SWEM algorithm on three data sets containing accelerometer data from physical activities. We showed that SWEM outperformed several baseline algorithms and its high accuracy suggested the viability of deploying this algorithm.

1 Background and Motivation

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 [2].

*School of EECS, Oregon State University. {zheng,wong,guan}@eecs.oregonstate.edu

†School of Human Movement Studies, University of Queensland. s.trost@uq.edu.au

Self-reports have traditionally been the means of providing information about physical activity [6]. 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 [21].

One of the most promising alternatives to self-reports is the use of accelerometers [6], 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 the triaxial accelerometer data collected from several different activity classes, with the data from each axis shown in a different color. 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 [19] 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. [17]), and are turning to data mining methods to identify physical activity types and estimate energy expenditure more accurately.

The basic data mining task involves classifying a triaxial time series (like the ones in Figure 1) as a single physical activity type. Many approaches to time series classification have been proposed in the data mining literature [24] 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 data mining 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 learning techniques can produce fairly 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 a classifier trained on a set of features computed from subwindows of different sizes from the orignal time series, thereby leveraging discriminative features found at different temporal scales. We evaluate this ensemble method on three different accelerometer data sets.

2 Related Work

Past work has explored activity recognition using data from a single accelerometer, typically placed at the waist or hip [17, 4]. A variety of standard supervised learning algorithms have been employed, but Ravi et al. [16] 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 [1, 14]. 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.

Time series classification has been an area of active research in data mining [24]. Our task specifically involves classifying an *entire* numeric time series as a single activity type. Methods accomplishing this task rely on representing the time series in such a way that captures the discriminative features. For instance, Lin et al. introduced the SAX representation which converts a time series into a symbolic representation [13]. Ye et al. extract discriminative local temporal patterns called shapelets to serve as features [25]. Once a raw time series has been converted into a feature-based 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 even with just Euclidean distance [9]. Dynamic time warping (DTW) [3] handles distortions in time series better than pure Euclidean distance and can be particularly effective when combined with k-NN [22]. 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 time series with repetitive patterns like our accelerometer data.

Hidden Markov Models (HMMs) have also been used for physical activity recognition [12, 15, 14]. Since HMMs can model transition 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.

3 Methods and Technical Solutions

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 [17], decision trees [4], quadratic discriminant analysis [15] and support vector machines [18].

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 [26], which generates a large set of potentially useful features. In our past work [27], 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 statitics as features that can all be extracted in linear time or less (Table 1).

The second issue involves determining the window size W, which is seldom addressed even though it affects the accuracy of the algorithm [20]. 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

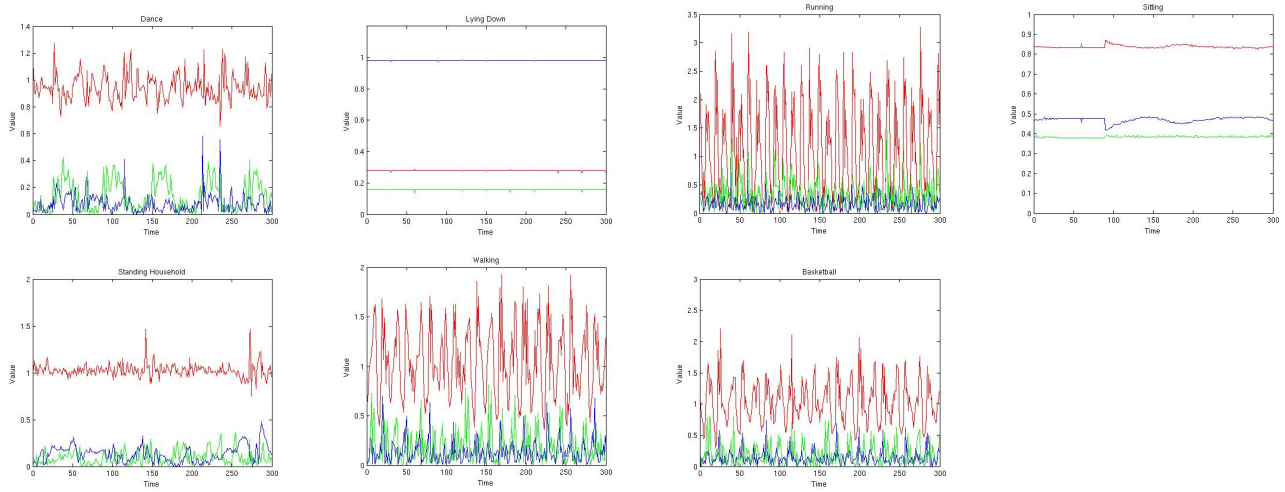

Figure 1: An example of 10 seconds of data from all seven classes in the OSU_Hip dataset. These plots illustrate triaxial accelerometer data collected at 30 Hz, and each color in a plot represents one axis. The activity types are, from left to right: (top) dance, lying down, running, sitting, (bottom) standing and household chores, walking and basketball.

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.

3.1 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 stacking [23] 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

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: $\sigma_s = \sqrt{\frac{1}{T}\sum_{i=1}^{T}(s_i - u_s)}$.
4. Coefficients of variation: $c_v = \frac{\sigma_s}{\mu_s}$.
5. Peak-to-peak amplitude: difference between maximum and minimum signal values: $max(S) - min(S)$.
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 T is the length of the window

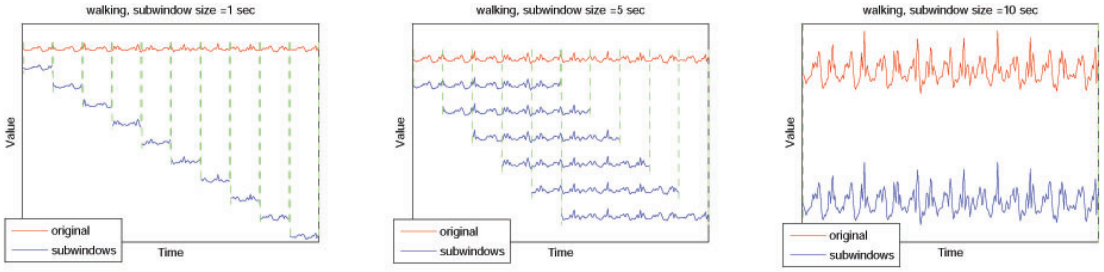

Figure 2: Decomposing a time series of a 10-second walk into 1, 5 and 10-second overlapping subwindows. The subwindows shift by 1 second.

$L = 1, 5, 10$, 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 trains the ensemble member with the training data and adds the trained ensemble member to the set M.

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

The training of the stacking meta-model is shown in Algorithm 2. The for loop in lines 4-11 iterates over ensemble members. Give an ensemble member m, we retrieve its associated subwindow length l in line 5. Then, in lines 7-9, we extract all possible subwindows of length l from the time series t. Each subwindow s is converted into a feature vector through the FEATURIZE function, and we predict its class label using ensemble member m. The predictions over all subwindows of size l are stored in p. In line 10, the overall classification for the time series t using ensemble member m (corresponding to temporal scale l) is obtained by a majority vote on p. Lines 12 and 13 form training instances for the meta-model. The features of these training instances consist of the predictions by each ensemble member (stored in the vector v) and the associated class label of time series t. Finally, in line 15, the meta model is trained[1]

Algorithm 3 describes how SWEM classifies a new time series. In lines 1-8, the SWEM-PREDICT function collects the predictions of each ensemble member. In line 3, we retrieve the length l of the subwindow associated with ensemble member M. The for loop in lines 4-6 converts all subwindows of length l in T into a feature vector, predicts the class of each feature vector, and adds the predictions to p. In line 10, the overall prediction by ensemble member m (at temporal scale l) is obtained by a majority vote over the predictions in p. Finally in line 11, the overall prediction is produced by the meta model using the predictions in v by each ensemble member m.

4 Empirical Evaluation

4.1 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 differ-

[1]Due to a limited amount of training data after splitting the original dataset into training, validation and testing parts, we train both the ensemble members and the meta model on the same training set. This has been known to cause overfitting, but our results show good generalization on holdout test data.

Algorithm 2 SWEMMetaTrain(M, T, c)

Input

M: the ensemble members generated by SWEMMemberTrain.
T: dataset of time series for training.

Output

meta: the meta model.

Procedure

```
training_data ← {}
for each time series t in T do
    label ← Class-Label(t), v ← {}
    for each ensemble member m from M do
        l ← SubwindowLength(m)
        p ← {}
        for each subwindow s of t with length l do
            p ← p ∪ Predict(m, Featurize(s))
        end for
        v[m] ← MajorityVote(p)
    end for
    x ← Feature-Vector(v)
    training_data ← training_data ∪ (x, label)
end for
meta ← Build-Model(training_data)
return meta
```

Algorithm 3 SWEMPredict($M, meta, t$)

Input

M: the ensemble members generated by SWEMMemberTrain.
meta: the meta model trained by SWEMMetaTrain.
t: the time series to be predicted.

Output

prediction: the prediction of time series t.

Procedure

```
v ← {}
for each ensemble member m in M do
    l ← SubwindowLength(m),
    p ← {}
    for each subwindow s of t with length l do
        p ← p ∪ Predict(m, Featurize(s))
    end for
    v[m] ← MajorityVote(p)
end for
prediction ← Predict(meta, v)
return prediction
```

ent 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 53 participants with wrist-mounted accelerometers while the OSU_Hip dataset contained data from 18 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 extremely similar and not shown due to space limitations. The two minutes of data were cut up into 10-second time windows. In our past work [20], we 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[2] [7, 8]. The data were collected from seven subjects with triaxial accelerometers at a 100 Hz sampling rate. Six activities: stay, walk, jog, skip, stUp (stair-up) and stDown (stair-down) were performed by all subjects in a controlled lab environment. Figure 3 shows an example of all six classes in the HASC dataset. As with the OSU data, we divide the HASC time series data into windows of length 10 seconds.

[2]Available at http://hasc.jp/hc2011/download-en.html

4.2 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 axes) was extracted from each pair of axes, resulting in a total of 57 features. The ensemble members and the meta-model for the SWEM were linear support vector machines. Libsvm[5] was used as the SVM implementation in our experiments.

The dataset was randomly split by subjects 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 classification accuracy 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 accuracy was reported. Note that each dataset has an approximately balanced number of time series from each physical activity type.

We compared the performance of SWEM against

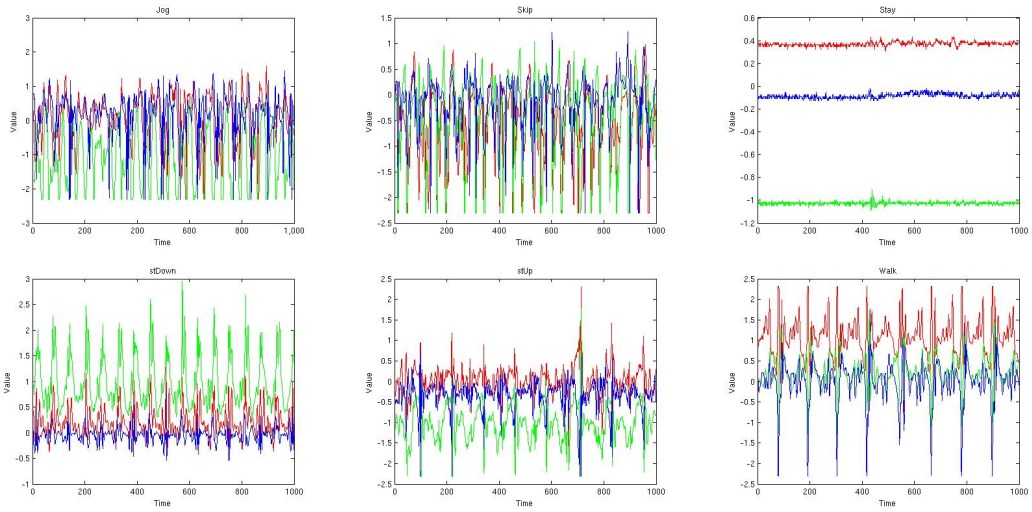

Figure 3: An example of 10 seconds of data from all six classes in the HASC dataset. These plots illustrate triaxial accelerometer data collected at 100 Hz, and each color in a plot represents one axis. The activity types are, from left to right: (top) jog, skip, stay, (bottom) stair-down, stair-up, and walk.

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, 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 an ANN in which 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 [25], but the training phase of the shapelet algorithm, which is computationally expensive, did not finish running on our full dataset[3].

[3]We were able to finish training the shapelet algorithm using a sample of 10% of the OSU training data. Shapelets resulted in accuracies of 0.46 and 0.43 on the sub-sampled OSU_Hip and OSU_Wrist datasets respectively, but were outperformed by SWEM_SVM trained on the same sub-sampled training data (0.94 and 0.88 accuracy on OSU_Hip and OSU_Wrist respectively).

5 Significance and Impact

Algorithm	OSU_Hip	OSU_Wrist	HASC
SWEM_SVM	**0.947**†	**0.911**†	**0.817**†
SVM	0.943	0.903	0.792
ANN	0.934	0.829	0.754
1NN_EUC	0.467	0.509	0.548
1NN_DTW	0.450	0.551	0.549
1NN_EUC_SAX	0.182	0.184	0.168
1NN_DTW_SAX	0.182	0.180	0.169

Table 2: Average classification accuracies of the various algorithms on the three datasets. The bold font marks the model with the highest average accuracy 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.

Table 2 illustrates the results of the experiments. The SWEM_SVM was the best performing model on the OSU_Hip (0.9465), OSU_Wrist (0.9106) and HASC (0.8165) 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 also performed poorly because the aggregation caused many of the discriminative details to be smoothed out.

Table 2 reports the average accuracy over all the activity types. In order to determine which *individual* activity types were more accurately predicted by the multi-scale ensemble, we compare the confusion matrices of SWEM_SVM vs SVM (see supplementary material). The majority of activity types were more accurately predicted by SWEM_SVM than SVM, with gains in accuracy by SWEM_SVM over SVM for basketball in OSU_Hip (0.027), dance in OSU_Wrist (0.038) and stair up in HASC (0.051). Dance in OSU_Wrist was a particularly difficult activity to recognize as it had the lowest accuracy out of all the activity types. There were only four activity types, however, where SVM was more accurate than SWEM_SVM, and these improvements were extremely minor (0.001-0.009 in accuracy).

We can gain further insight into the benefits of the multi-scale approach of SWEM_SVM by comparing its performance to that of its ensemble members individually by removing the stacking layer. 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 ensemble member individually on the three 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 SVM results listed 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 do not recommend using SWEM_SVM on other types of time series data without the repetitive patterns seen in accelerometer data capturing physical activity. For instance, the UCR time series data repository [10] contains many datasets with time series that capture a single complex activity, such as the Cricket dataset [11] which contains wrist-mounted accelerometer signals of cricket umpires performing twelve different signals. On such time series, 1NN_DTW is much more successful than SWEM_SVM because it can align a test instance to the overall shape of its nearest neighbor. SWEM_SVM, on the other hand, cannot fully represent the overall shape of the time series with the features generated from its subwindows.

6 Conclusion and Future Work

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 ($> 90\%$ for OSU data, $> 80\%$ for HASC), which suggest 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-lving data consists of a mixture of different activities performed throughout an individual's day, we explore algorithms for segmenting the data and classifying these segments into physical activity types.

7 Acknowledgements

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

References

[1] L. Bao and S. S. Intille, *Activity Recognition from User-Annotated Acceleration Data Pervasive Computing*, Pervasive Computing, 3001 (2004), pp. 1–17.

[2] A. Bauman, P. Phongsavan, S. Schoeppe, and N. Owen, *Physical activity measurement - a primer for health promotion*, IUHPE - Promotion and Education, 8 (2006), pp. 92–103.

[3] D. Berndt and J. Clifford, *Using dynamic time warping to find patterns in time series*, in AAAI-94 workshop on knowledge discovery in databases, vol. 2, 1994.

[4] A.G. Bonomi, A.H.C. Goris, B. Yin, and K.R. Westerterp, *Detection of type, duration, and intensity of physical activity using an accelerometer*,

Medicine & Science in Sports & Exercise, 41 (2009), p. 1770.

[5] C.C. Chang and C.J. Lin, *Libsvm: a library for support vector machines*, ACM Transactions on Intelligent Systems and Technology (TIST), 2 (2011), p. 27.

[6] S.I. de Vries, F.G. Garre, L.H. Engbers, V.H. Hildebrandt, and S. van Buuren, *Evaluation of neural networks to identify types of activity using accelerometers*, Medicine & Science in Sports & Exercise, 43 (2011), p. 101.

[7] N. Kawaguchi, N. Ogawa, Y. Iwasaki, K. Kaji, T. Terada, K. Murao, S. Inoue, Y. Kawahara, Y. Sumi, and N. Nishio, *Hasc challenge: gathering large scale human activity corpus for the real-world activity understandings*, in Proceedings of the 2nd Augmented Human International Conference, ACM, 2011, p. 27.

[8] N. Kawaguchi, H. Watanabe, T. Yang, N. Ogawa, Y. Iwasaki, K. Kaji, T. Terada, K. Murao, H. Hada, S. Inoue, Y. Sumi, Y. Kawahara, and N. Nishio, *Hasc2012corpus: Large scale human activity corpus and its application*, in Proceedings of the 2nd International Workshop on Mobile Sensing, 2012.

[9] E. Keogh and S. Kasetty, *On the need for time series data mining benchmarks: a survey and empirical demonstration*, Data Mining and Knowledge Discovery, 7 (2003), pp. 349–371.

[10] E. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, and C.A. Ratanamahatana, *The ucr time series classification/clustering homepage (2011)* `http://www.cs.ucr.edu/~eamonn/time_series_data`.

[11] M. H. Ko, G. West, S. Venkatesh, and M. Kumar, *Using dynamic time warping for online temporal fusion in multisensor systems*, Inf. Fusion, 9 (2008), pp. 370–388.

[12] Jonathan Lester, Tanzeem Choudhury, Nicky Kern, Gaetano Borriello, and Blake Hannaford, *A hybrid discriminative/generative approach for modeling human activities*, in Proceedings of the 19th international joint conference on Artificial intelligence, IJCAI'05, San Francisco, CA, USA, 2005, Morgan Kaufmann Publishers Inc., pp. 766–772.

[13] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, *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, ACM, 2003, pp. 2–11.

[14] A. Mannini and A. M Sabatini, *Machine learning methods for classifying human physical activity from on-body accelerometers*, Sensors 2010, 10 (2010), pp. 1154–1175.

[15] D.M. Pober, J. Staudenmayer, C. Raphael, and P.S. FREEDSON, *Development of novel techniques to classify physical activity mode using accelerometers*, Medicine & Science in Sports & Exercise, 38 (2006), p. 1626.

[16] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, *Activity recognition from accelerometer data*, in Proceedings of the 17th conference on Innovative applications of artificial intelligence - Volume 3, IAAI'05, AAAI Press, 2005, pp. 1541–1546.

[17] J. Staudenmayer, D. Pober, S. Crouter, D. Bassett, and P. Freedson, *An artificial neural network to estimate physical activity energy expenditure and identify physical activity type from an accelerometer*, Journal of Applied Physiology, 107 (2009), pp. 1300–1307.

[18] S. W. Su, L. Wang, B. G. Celler, E. Ambikairajah, and A. V. Savkin, *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, IEEE, 2005, pp. 3526–3529.

[19] M.S. Treuth, K. Schmitz, D. J. Catellier, R. G. McMurray, D. M. Murray, M. J. Almedia, S. Going, J. E. Norman, and R. Pate, *Defining accelerometer thresholds for activity intensities in adolescent girls*, Med. Sci. Sports Exerc., 36 (2004), pp. 1259–1266.

[20] S.G. Trost, W.K. Wong, K.A. Pfeiffer, and Y. Zheng, *Artificial neural networks to predict activity type and energy expenditure in youth.*, Medicine and science in sports and exercise, (2012).

[21] C.E. Tudor-Locke and A.M. Myers, *Challenges and opportunities for measuring physical activity in sedentary adults*, Sports Medicine, 31 (2001), pp. 91–100.

[22] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, *Experimental comparison of representation methods and distance measures for time series data*, Data Mining and Knowledge Discovery, (2010), pp. 1–35.

[23] D. Wolpert, *Stacked generalization*, Neural Networks, 5 (1992), pp. 241–259.

[24] Z. Xing, J. Pei, and E. Keogh, *A brief survey on sequence classification*, SIGKDD Explor. Newsl., 12 (2010), pp. 40–48.

[25] L. Ye and E. Keogh, *Time series shapelets: a new primitive for data mining*, in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2009, pp. 947–956.

[26] M. Zhang and A.A. Sawchuk, *Motion primitive-based human activity recognition using a bag-of-features approach*, in Proceedings of the 2nd ACM SIGHIT symposium on International health informatics, ACM, 2012, pp. 631–640.

[27] Y. Zheng, *Predicting activity type from accelerometer data*, master's thesis, Oregon State University, 2012.

Model	Accuracy	Classification Accuracy of Each Physical Activity						
		lying	sitting	standing	walking	running	basketball	dance
SWEM_SVM	0.9465	0.9806	0.9423	0.9678	0.9541	0.9823	0.9419	0.8041
SWEM_SVM1	0.9219	0.9709	0.9294	**0.9893**	0.9488	**0.9876**	0.7398	0.6931
SWEM_SVM2	0.9402	0.9735	0.9271	0.9836	0.9543	0.9844	0.8931	0.7648
SWEM_SVM3	0.9419	0.9719	0.9365	0.9727	0.9502	0.9870	0.9283	0.7756
SWEM_SVM4	0.9408	0.9800	0.9265	0.9709	0.9533	0.9810	0.9178	0.7861
SWEM_SVM5	0.9401	0.9780	0.9357	0.9564	0.9494	0.9811	**0.9407**	0.7931
SWEM_SVM6	0.9411	0.9787	0.9299	0.9609	0.9572	0.9798	0.9306	0.7911
SWEM_SVM7	0.9422	0.9802	0.9353	0.9519	0.9565	0.9798	0.9378	**0.8131**
SWEM_SVM8	0.9420	**0.9819**	0.9296	0.9608	**0.9615**	0.9776	0.9206	0.7991
SWEM_SVM9	**0.9436**	0.9817	**0.9374**	0.9572	0.9567	0.9789	0.9359	0.8104
SWEM_SVM10	0.9426	0.9772	0.9318	0.9666	0.9599	0.9776	0.9161	0.7978

Table 3: Activity classification accuracies of the overall SWEM_SVM algorithm and of each ensemble member on the OSU_Hip dataset. The bold font marks the highest accuracy of a single subwindow model for each activity.

Model	Accuracy	Classification Accuracy of Each Physical Activity						
		lying	sitting	standing	walking	running	basketball	dance
SWEM_SVM	0.9106	0.7993	0.9522	0.9389	0.9562	0.8345	0.9072	0.7722
SWEM_SVM1	0.8727	0.7368	**0.9608**	**0.9767**	0.9639	**0.8565**	0.4300	0.7257
SWEM_SVM2	0.8993	0.7708	0.9571	0.9618	**0.9700**	0.8547	0.7206	0.7243
SWEM_SVM3	0.9073	0.7924	0.9481	0.9464	0.9583	0.8507	0.8461	0.7694
SWEM_SVM4	0.9073	0.7778	0.9559	0.9455	0.9563	0.8426	0.8828	0.7403
SWEM_SVM5	**0.9080**	**0.8021**	0.9438	0.9368	0.9514	0.8281	**0.9194**	**0.7701**
SWEM_SVM6	0.9073	0.7875	0.9549	0.9382	0.9534	0.8299	0.9056	0.7535
SWEM_SVM7	0.9052	0.8007	0.9549	0.9239	0.9522	0.8218	0.9150	0.7611
SWEM_SVM8	0.9055	0.7764	0.9605	0.9378	0.9567	0.8235	0.8933	0.7410
SWEM_SVM9	0.9032	0.7903	**0.9608**	0.9192	0.9518	0.8241	0.9050	0.7576
SWEM10_SVM10	0.9025	0.7819	0.9590	0.9340	0.9575	0.8148	0.8822	0.7312

Table 4: 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	Accuracy	Classification Accuracy of Each Physical Activity					
		stay	walk	jog	skip	stUp	stDown
SWEM_SVM	0.8165	0.9956	0.7656	0.7989	0.8400	0.7111	0.7878
SWEM_SVM1	0.8113	**1.0000**	0.7456	**0.8122**	0.8267	0.6800	0.8033
SWEM_SVM2	**0.8150**	0.9989	0.7367	0.7956	**0.8456**	0.7067	**0.8067**
SWEM_SVM3	0.8124	**1.0000**	0.7389	0.8044	0.8178	**0.7211**	0.7922
SWEM_SVM4	0.8054	0.9989	0.7378	0.7911	0.8267	0.6978	0.7800
SWEM_SVM5	0.8028	0.9944	0.7489	0.8000	0.8244	0.7000	0.7489
SWEM_SVM6	0.8004	0.9933	0.7322	0.7889	0.8322	0.7000	0.7556
SWEM_SVM7	0.7909	0.9944	0.7389	0.8033	0.8156	0.6567	0.7367
SWEM_SVM8	0.7909	0.9867	0.7233	0.7878	0.8344	0.6689	0.7444
SWEM_SVM9	0.7961	0.9878	**0.7767**	0.8078	0.8111	0.6767	0.7167
SWEM10_SVM10	0.7919	0.9811	0.7311	0.7944	0.8256	0.6689	0.7500

Table 5: 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.