Addressing the Problem of Activity Recognition with Experience Sampling and Weak Learning

William Duffy, Kevin Curran, Daniel Kelly, Tom Lunney
School of Computing, Engineering & Intelligent Systems
Ulster University
Derry, United Kingdom
Duffy-w@ulster.ac.uk

***Abstract*—Quantifying individual's levels of activity through smart or proprietary devices is currently an active area of research. Current implementations use subjective methods, for instance, questionnaires or require comprehensively annotated datasets for automated classification. Each method brings its own specific drawbacks. Questionnaires cause recall bias and providing annotations for datasets is difficult and tedious. Weakly supervised methodologies provide methodologies for handling inaccurate or incomplete annotations and literature has shown their effectiveness for classifying activity data. As a key issue of activity recognition is capturing annotations, the aim of this work is to evaluate how classification performance is affected by limiting annotations and to investigate potential solutions. Experience sampling combined with the algorithms in this paper can result in a classifier accuracy of 74% with a 99.8% reduction in annotations, with increased compute overheads. This paper shows that experience sampling combined with a method of populating labels to unlabeled feature vectors can be a viable solution to the annotation problem.**

Keywords—Activity recognition; weak supervision; experience sampling; multi-class classification; ECOC SVM

I. INTRODUCTION

Currently, the practicality of collecting user specific activity data has never been so high. This is due to the global adoption of smart devices across users of all age ranges, providing new opportunities for signal analysis and the application of pattern recognition techniques.

With advances in healthcare leading to extensions in overall lifespan, the elderly are becoming a larger proportion of the overall population, placing a greater strain on the healthcare system and increasing the associated costs [1]. As a result, methods of automated activity detection are receiving interest in research. Methods of detecting the onset of serious medical issues, for example, gait analysis of an individual thought to be suffering from Parkinson's disease has advantages both in patient care and potentially reducing the cost to healthcare providers [2]. Wearable devices can also be useful in the field of biometric security, and sports such as dart throw analysis [3].

A prominent issue and the focus of this research is the problem of annotated data. Having a fully annotated dataset is desirable as it provides the maximum amount of training data for a supervised classifier to learn from. This gives the classifier the best chance at recognizing this activity when new unseen data is introduced. Generally, fully annotated data is expensive to record [4]. This is due to both the effort involved in the capture process but also the equipment and setup [2]. The typical setup of an activity recognition capture session involves connecting multiple sensors and performing the same activity over an extended period to ensure sufficient data is available to both train and validate the model. Many studies are also performed in laboratory settings, further complicating the capture of useful data [1]. Even with a successfully completed data capture experiment, we are limited not only to the activities captured but also to the individuals who performed them. A system which is trained to recognized activities of a specific person will always be more successful than a generic system. This is due to the slight idiosyncrasies displayed by each person in performing the same activity. However, with many movement sensors having high refresh rates, asking users to accurately annotate their own activities is not feasible. For this reason, weakly supervised methodologies are being proposed to solve the issue. These methodologies take considerably weaker labelling information than normal supervised methods [5]. Generally, unlabeled data is available in greater quantities than labelled data. Weak supervision takes advantage of this and tries to gain information from the relationship between labelled and unlabeled data points [4].

If wearables are to be used, and if the models are to be truly tailored to each user, then a method of gathering data from the user is required to allow data to be annotated. One avenue is to use the experience sampling method which is a type of diary study where the user will annotate the data manually. Although this type of data collection could be considered intrusive if a user is requested for data continuously [4]. In order to reduce the number of queries sent to the user and therefore reduce the intrusiveness of the method, weak supervision techniques can be used to gather extra information about the data points which the user annotated.

This paper sets out to show that experience sampling in combination with a method of weak learning could provide a solution to the problem of annotated data for activity recognition.

Section III demonstrates our methodology, Section IV outlines the experiments, Section V discusses results and Section VI presents our conclusions and recommendations for future work.

978-1-5386-4185-9/18/$31.00 ©2018 IEEE

II. Related Work

Research into weak supervision for activity recognition has shown potential [4], [6]. Experience sampling, a type of diary study, has been used to gather annotations from the user about the activity they are performing. Multiple methods of annotation request are used, some which ask what activity users have been performing the most over a given time frame and some which ask for all the activities performed within the time window [4]. Instead of polling users at fixed intervals, context aware methods of experience sampling exist but have not been tested for the purpose of activity recognition. These apply a cost benefit approach to asking the user for input [7]. This approach could be useful in reducing the number of intrusive data requests the system makes.

Weak supervision techniques are variations on standard supervised techniques, they attempt to focus on gaining knowledge from the unlabeled data as it is available in far greater quantities than labelled data. One such variation is multiple instance learning which places feature vectors into sets known as bags. Classifiers, for instance, SVM are modified to accept these bags. Previous work applied a multiple instance SVM with experience sampling. Classifier accuracy improved in comparison to an SVM provided only the labels gained through experience sampling for sampling windows of 10, 30 and 120 minutes [4].

Another weak supervision method is graph label propagation which attempts to group labelled and unlabeled data into communities. Each community is then assigned a label based on the known labelled instances inside these communities. This label is then applied to all the unlabeled instances within that community. When applied to the TU Darmstadt dataset, graph label propagation exceeded the baseline accuracies set by an SVM trained with full ground truth [4].

Active learning for activity recognition, which attempts to discover the unlabeled data points which the classifier can learn most from. When applied to accelerometer data which initially has minimal annotations it provides a significant increase in accuracy [6]. However, issues with active learning are noisy annotators and the classifier may focus on difficult to annotate data points [8].

III. Methodology

A. Dataset

The Human Activities and Postural Transitions (HAPT) dataset [9] uses a waist mounted smartphone to collect 6 activities. The signals from the gyroscope and the accelerometer are sampled at 50Hz. In total, 30 participants are included in the dataset with the age range being between 19-48 years.

The dataset contains a mixture of static and dynamic activities as well as the postural transitions which occurred on the static activities. For the purposes of this experiment, the postural transitions are being ignored. The static activities performed are standing, sitting and lying and the dynamic activities are walking, walking upstairs/downstairs.

Validation is achieved through the use of a random train/test split. The participants in the training set are completely independent of the test set, so the system will not be tested on individuals it has been trained on. The training data is made up of 70% of the participants and the test data of the remaining 30%.

B. Feature Extraction

The features used for this dataset are pre-computed. They come from the raw 3-axis accelerometer and gyroscope signals and are a mixture of time domain and frequency domain features. The signals are de-noised using a median and low pass Butterworth filter. Five different time series are produced from these signals. Acceleration values produced from "body" movements and the acceleration values produced from gravity. The signals are split using a low pass Butterworth filter. The jerk of the body signals are then produced by calculating the rate of change of the acceleration values with respect to time. The time series now produced are the acceleration due to gravity, the body acceleration, the body gyroscope acceleration and jerk of the body signals. The magnitude of each of these is calculated using norms. Fast Fourier transforms are calculated from these signals.

The signals are split into windows which are 128 samples in length, approximately 2.6 seconds of data. Several variables are calculated from each of these signals, including mean, standard deviation, max/min values, median absolute deviation, signal magnitude area, energy of signal, interquartile range, signal entropy, correlation coefficients, the auto regression coefficient, angle between the vectors, skewness, kurtosis, mean frequency, frequency component which has the largest magnitude and energy of the frequency interval of the Fourier transform window. The mean of the angle between the vectors is also calculated. This results in a 561 length feature vector.

C. Experience Sampling

Experience sampling is a diary study method for gathering data from a user. We simulate it in this experiment by asking a user for data about each activity they are currently performing. This process is simulated by finding which feature vectors lie on the experience sampling intervals. E.g. for a one minute experience sampling window, the feature vector which lies on exactly one minute will be used. The annotation for this feature vector is then obtained from the original set of annotations.

A consequence of this method is that we only get back what we ask from the user. Requesting more data from the user will be beneficial to the experiment as it could provide more data for training but such a system could be considered intrusive to the user. With this in mind, designing the questions the system will ask the user is important. Previous implementations have asked for data like what activity they have been performing the most since the previous data request and what activity are they currently performing. This experiment will only simulate asking what they are currently performing and will label the current feature vector with the data collected from the user.

A potential issue is that when a user is about to input the data they will move the device, and the movement of using the device will be incorrectly labelled as the locomotive activity

978-1-5386-4185-9/18/$31.00 ©2018 IEEE

they are performing. Since this data has been pre-collected and since we are simulating the user entering a label, this is not an issue but if this were to be used on an online system it would require data collection to be halted until the user has completed the request and resume once they have stopped using the device.

The experience sampling will be performed using several different time intervals. The simulated annotation requests will be performed every 1, 3, 5, 10, 15, 20, 30 and 60 minutes.

To ensure that the experience sampling methodology is providing an unbiased view of the data, the sampling times are being randomly moved back and forward by a few seconds. The entire experiment is then repeated with these new experience sampling labels. This process is only performed on the training data, the test set does not change.

1) Short experience sampling windows (1-5 minutes)

It can be expected that the short experience sampling windows will perform like the fully supervised approach. At one minute intervals, even with 95% of labels removed, the system should still have sufficient data to classify 6 activities with reasonable accuracy.

2) Medium experience sampling windows (10-20 minutes)

We postulate that this could be the optimal window size as it provides an acceptable break between data collection requests, while potentially providing enough data for classification.

3) Long experience sampling windows (30+ minutes)

It is suspected that this length of sampling windows will show the limitations of the dataset, with potentially insufficient information being captured to provide accurate classification. It will be possible for entire activities to be missed.

D. Weak Supervision

Two algorithms are presented which attempt to populate labels to unlabeled feature vectors using pairwise Euclidean distance. The initial setup of both algorithms is the same, with a 561-width matrix containing all feature vectors stored in X and labels stored in Y with $Yi \in \{0,1, \dots 6\}$. All labels in Y will initially be 0 except those collected using experience sampling.

The variable K is used by algorithm 1, it controls the number of times the algorithm repeats. As the amount of processing is exponentially increased for each incrementally higher value of K so the values will be limited to between 1 and 3.

Algorithm 1 presents the methodology for the single label based propagation. After the experience sampling has gathered labels, the algorithm runs through each of the experience sampling labelled feature vectors and finds the single closest unlabeled feature vector to each. Each of these single vectors is given the same label as the experience sampling feature vector, effectively doubling the number of labelled feature vectors. This process is then repeated K times, however, for each successive run, the original labelled feature vectors are removed. For example, the system is looking for the closest variables to the experience sampling points on the first run, these experience sampling points are then removed for the second run. Without this step, it will just re-find the original points.

Algorithm 1 Single label propagation

$xPoints = index\ of\ each\ position\ Y\ which\ is\ not\ zero$
$for\ each\ x\ in\ xPoints$
    $cLabel = value\ of\ Y\ at\ position\ x$
    $cFeature = feature\ vector\ of\ X\ at\ position\ x$
    $For\ each\ z\ in\ X$
        $distances(z) = pairwise\ distance\ of\ cFeature\ and\ X(z)$
    end
    $lowest = \min(distances)\ which\ is\ not\ index\ x$
    $Y(lowest) = cLabel$
end
$For\ x = 1{:}K$
    temp = xPoints
    $xPoints = index\ of\ each\ position\ Y\ which\ is\ not\ zero$
    $Remove\ temp\ from\ xPoints$
    $Repeat\ lines\ 2\ to\ 10\ with\ new\ xPoints$
end

For the second algorithm, the variable M is used to control the number of similar feature vectors to look at for comparison. The algorithm aggregates the labels gathered by experience sampling into groups, for example, all feature vectors labelled as walking will be grouped together. The average of each of feature vectors in each group is taken, these averages are then compared against the unlabeled data points looking for the M closest. These M data points are then given the label of the closest group.

E. Classification

As Support Vector Machines have been shown to perform well on activity data, classification will be achieved through the use of an error-correcting output codes support vector machine (ECOC SVM). An ECOC SVM as it allows an SVM to act as a multi-class classifier. This is advantageous for activity data as generally there are multiple activities to classify.

This is achieved by providing an ensemble of binary one vs one classifiers [10]. The error-correcting codes output by the ECOC SVM means that although there are extra data overhead, these codes can help the system recover from errors such as poor input features or a flawed training algorithm, something which may be appropriate for weak supervision techniques since noisy labels are inevitable.

Decision tree based ensemble methods have also been used successfully for activity recognition, again the ensemble method has the capability for multi-class classification. In this case, a Tree Bagger will be used.

Each of these classifiers will be tested on the fully supervised data and the more successful of these two selected.

978-1-5386-4185-9/18/$31.00 ©2018 IEEE

Algorithm 2 Unique aggregate label propagation

$xLabels = 2 - width\ matrix\ containing\ each\ known\ label\ and\ its\ index$
$uLabels = unique\ labels\ in\ xLabels$
$for\ each\ x\ in\ uLabels$
  $tLabel = label\ in\ xLabels\ at\ position\ x$
  $temp = indexes\ of\ Y\ with\ label\ tLabel$
  $temp2 = feature\ vectors\ in\ X\ with\ indexes\ in\ temp$
  $featuresAvg = mean\ of\ features\ in\ temp2$
  $for\ each\ z\ in\ X$
    $distances(z) = pairwise\ distance\ between\ featuresAvg\ and\ X(z)$
  end
  $counter = 0$
  $while(counter \leq M)$
    $smallest(counter) = \min(distances)\ which\ is\ not\ index\ x$
  end
  $Y(smallest) = tLabel$
end

IV. Experiments

The following experiments will first evaluate the performance of standard supervised classification with full ground truth. The amount of data available to the classifier will then be restricted via experience sampling and finally new labels introduced through weak supervision methods.

A. Supervised

Standard supervised learning was performed by using the pre-computed features with all the provided activity labels.

The results shown in Table I provide a baseline for weakly supervised performance to be compared against, with an overall average of 96.4% the ECOC SVM performed significantly better than the Tree Baggers average of 90.1%. As it performs better then ECOC SVM will be used for any future experiments.

TABLE I. Supervised Classifier Comparison

	Accuracy	
Activity	**ECOC SVM**	**Tree Bagger**
Walking	99.4%	94.0%
Walking upstairs	96.8%	86.8%
Walking downstairs	96.4%	82.1%
Sitting	89.2%	88.6%
Standing	96.9%	89.2%
Laying	99.8%	99.8%

TABLE II. Experience Sampling Window Length

Sampling window (min)	0	1	3	5	10	15	20	30	60
Number of labels	7415	322	107	64	32	21	16	10	5

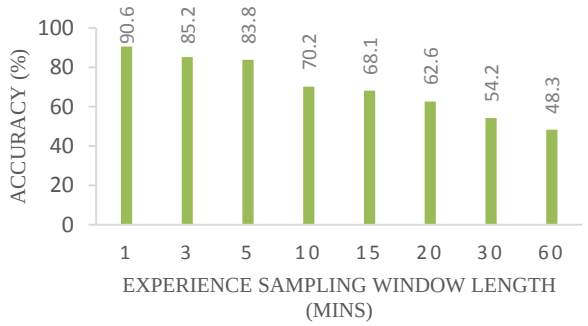

Fig. 1. Experience sampling performance.

B. Weakly Supervised

1) Randomisation of experience sampling requests

To provide as unbiased view of the data as possible, each of the experience sampling requests will be randomly moved forward or backwards by a random number of seconds.

A seed value of 1 is being used to ensure that these results are repeatable and that on each experiment is being provided with the same annotations.

2) Experience Sampling Only

Table II shows the number of labels found through experience sampling for each length of sampling window. Even a one minute interval significantly reduces the number of labels available in the training set, any feature vector which has not been assigned a label is discarded as it is not usable by a supervised classifier.

Fig. 1 shows that the experience sampling method alone provides excellent results at low window lengths but once the sampling window exceeds 5 minutes performance is significantly reduced.

3) K and M values

Experiments were performed to find the most appropriate values for the K and M values, these will be performed on the 1, 10 and 20 minute experience sampling windows. These times have been selected as they provide data about both short and medium length sample windows. The longer windows are not being used due to dataset limitations.

The bigger the values the more labels will be generated, however, this could also result in an increased number of incorrect labels compared to ground truth.

a) Algorithm 1

We can see from Table III that increasing the K-value does increase the number of new labels generated, however, the bigger the K-value used, the higher chance that incorrect labels will be created.

A K-value of three will be used for the remainder of the experiment. This is due to it having significantly more labels than the other K-values, while only having a slight reduction in quality. K-values of above 3 will likely generate more labels, however, the performance impact is unacceptably high.

978-1-5386-4185-9/18/$31.00 ©2018 IEEE

TABLE III. K-VALUE TUNING ALGORITHM 1

1 Minute Sampling Window			
K value	1	2	3
New labels	1274	2547	5094
Percent Correct	96.8%	94.8%	91.7%
10 Minute Sampling Window			
K value	1	2	3
New labels	128	256	512
Percent Correct	97.8%	95.3%	93.4%
20 Minute Sampling Window			
K value	1	2	3
New labels	64	128	256
Percent Correct	98.4%	96.9%	95.2%

TABLE IV. M-VALUE TUNING ALGORITHM 2

1 Minute Sampling Window			
M value	10	20	30
New labels	378	434	491
Percent Correct	98.8%	98.1%	97.1%
10 Minute Sampling Window			
M value	10	20	30
New labels	85	144	169
Percent Correct	97.0%	94.7%	90.4%
20 Minute Sampling Window			
M value	10	20	30
New labels	69	127	186
Percent Correct	96.2%	95.9%	94.8%

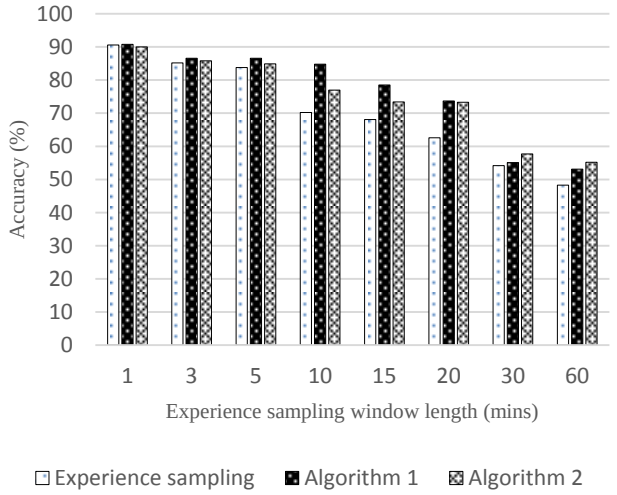

Fig. 2. Classifier performance.

b) Algorithm 2

Based on the results in Table IV, an M value of 20 is selected for Algorithm 2. The reduction in label quality between the values of 10 and 20 is insignificant compared to between 20 and 30. The value of 20 also provides substantially more labels than 10, which will be useful to the classifier.

4) Classifier Results

Fig. 2 shows the overall classification accuracy for each experience sampling window length. The graph shows that as the sampling window length increases above 5 minutes the classifier accuracy of experience sampling starts to decrease quickly whereas, the two weakly supervised algorithms decrease much slower until the 30 minute mark. Both algorithms allow the experience sampling window to be increased to 20 minutes while maintaining an accuracy of above 70%. Algorithm 1 performs better overall, but not significantly. After the 20 minute window accuracy sharply decreases for both Algorithm 1 and 2. This is possibly due to the experience sampling window being so long that entire activities are being missed.

Table V shows the classifier accuracy for individual activities in three different sampling window lengths. The one minute sampling length provides the closest comparison to fully supervised, while lengths greater than 20 minutes run into the limitations of the dataset. For each of these window lengths we can see that certain activities appear to be affected differently based on the length of the sampling window. Walking, standing and laying do not appear to be significantly impacted even with a sampling length of 20 minutes. However, walking upstairs, walking downstairs and sitting are all reduced. Walking upstairs is affected the worst with a significant reduction from 1 to 20 minute sampling length. It could be that certain activities need more data for accurate classification or the dataset could be unbalanced.

TABLE V. PER ACTIVITY CLASSIFIER PERFORMANCE

1 Minute Sampling Window						
Activity	**Walking**	**Walking Upstairs**	**Walking Downstairs**	**Sitting**	**Standing**	**Laying**
Experience sampling	98.2%	91.5%	84.2%	81.9%	87.1%	99.8%
Algorithm 1	95.8%	94.4%	88.2%	79.9%	88.2%	99.8%
Algorithm 2	97.9%	90.4%	84.8%	79.0%	87.4%	99.8%
10 Minute Sampling Window						
Activity	**Walking**	**Walking Upstairs**	**Walking Downstairs**	**Sitting**	**Standing**	**Laying**
Experience sampling	91.6%	52.8%	55.4%	19.3%	95.0%	96.6%
Algorithm 1	94.4%	70.2%	70.8%	30.2%	97.2%	95.6%
Algorithm 2	87.4%	58.9%	76.3%	46.5%	91.6%	97.2%
20 Minute Sampling Window						
Activity	**Walking**	**Walking Upstairs**	**Walking Downstairs**	**Sitting**	**Standing**	**Laying**
Experience sampling only	95.5%	5.3%	41.9%	44.1%	84.9%	94.9%
Algorithm 1	93.3%	38.9%	71.1%	55.8%	83.3%	94.9%
Algorithm 2	92.8%	18.8%	47.3%	75.3%	68.4%	88.8%

Overall, the two algorithms appear to be effective at increasing the classifier accuracy of the worst performing activities. With algorithm 1 increasing the accuracy of walking upstairs and walking downstairs by 33.6% and 29.2% respectively at the 20 minute experience sampling window.

5) Performance results

Although Algorithm 2 has not achieved the classification results that Algorithm 1 did, it is significantly faster as shown

978-1-5386-4185-9/18/$31.00 ©2018 IEEE

in Fig. 3. Although in recent years smart devices have greatly improved in terms of battery life and compute performance [11], they would be the ideal platform for an activity recognition system and anything which reduces the power impact of these systems would be advantageous.

V. Discussion

While these methods do not provide a completely unobtrusive system, they do maintain reasonable levels of classification accuracy even with significant reductions in the number of labelled feature vectors.

Limitations of this work include entire activities can be missed when the experience sampling window becomes too long. A potential solution is to provide a method of moving the request timings to detect more valuable data. These methods could also be used to increase the experience sampling request timings if the data to be collected provides limited new information. This could result in a system with fewer requests and a higher level of user acceptance.

Another limitation of this work is the more activities in the time series there is an increased likelihood that noisy labels will be produced. A potential solution is to provide multiple methods of validating populated labels rather than just their distance in the feature space.

VI. Conclusion and Future Work

As shown, experience sampling is a potentially viable method for collecting activity data. Classifiers perform surprisingly well when presented with minimal training samples, allowing a 10-minute experience sampling window to provide 70% classifier accuracy.

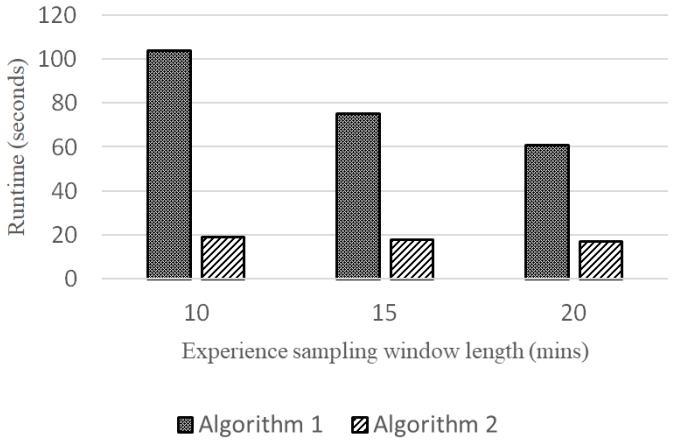

Fig. 3. Algorithm compute performance.

The weak supervision techniques introduced in this paper allows this window to be effectively doubled, with classifier accuracy of 74% even with a 20-minute sampling window. However, above this window length generally results in problems as entire activities are missed and the limitations of the dataset are pushed. The problem is shown in both the experience sampling labels and the weak supervision labels.

A potential issue with the experience sampling method is that when the user is polled for data they will be required to make movements in order to put the annotation into a device. As this experiment is mocked up on already collected data there is no way to simulate this, however a potential solution is to stop collecting data from sensors at a fixed time before alerting the user and begin collecting after a period has expired. Another solution to this issue could be to ask what activity a user was performing a predefined number of seconds ago, e.g. what activity did you perform 30 seconds ago. Another issue is that people will be performing movements with the device in different orientations, for example, walking while changing a song.

Future work will be on determining the ideal time for asking the user for input other than just timing, for example, an online classifier which attempts to discover when an activity that the system does not recognize is being performed.

References

[1] C. Wong, Z. Q. Zhang, B. Lo, and G. Z. Yang, “Wearable Sensing for Solid Biomechanics: A Review,” IEEE Sensors Journal, vol. 15, no. 5. pp. 2747–2760, 2015.

[2] W. Y. Wong, M. S. Wong, and K. H. Lo, “Clinical applications of sensors for human posture and movement analysis: a review.,” Prosthet. Orthot. Int., vol. 31, no. 1, pp. 62–75, 2007.

[3] Y. L. Zheng et al., “Unobtrusive sensing and wearable devices for health informatics,” IEEE Trans. Biomed. Eng., vol. 61, no. 5, pp. 1538–1554, 2014.

[4] M. Stikic, D. Larlus, S. Ebert, and B. Schiele, “Weakly supervised recognition of daily life activities with wearable sensors,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 12, pp. 2521–2537, 2011.

[5] J. Hernández-González, I. Inza, and J. A. Lozano, “Weak supervision and other non-standard classification problems: A taxonomy,” Pattern Recognit. Lett., vol. 69, pp. 49–55, 2016.

[6] M. Stikic, K. Van Laerhoven, and B. Schiele, “Exploring semi-supervised and active learning for activity recognition,” 2008 12th IEEE Int. Symp. Wearable Comput., pp. 81–88, 2008.

[7] Kapoor and E. Horvitz, “Experience sampling for building predictive user models,” Proceeding twenty-sixth Annu. CHI Conf. Hum. factors Comput. Syst. - CHI ’08, pp. 657–666, 2008.

[8] Settles, “Active Learning Literature Survey,” Mach. Learn., vol. 15, no. 2, pp. 201–221, 2010.

[9] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, “Transition-Aware Human Activity Recognition Using Smartphones,” Neurocomputing, vol. 171, pp. 754–767, 2016.

[10] T. G. Dietterich and G. Bakiri, “Solving Multiclass Learning Problems via Error-Correcting Output Codes,” Jouranal Artifical Intell. Res., vol. 2, pp. 263–286, 1995.

[11] M. Shoaib, S. Bosch, O. Incel, H. Scholten, and P. Havinga, “A Survey of Online Activity Recognition Using Mobile Phones,” Sensors, vol. 15, no. 1, pp. 2059–2085, 2015.

978-1-5386-4185-9/18/$31.00 ©2018 IEEE