Leveraging Active Learning and Conditional Mutual Information to Minimize Data Annotation in Human Activity Recognition

REBECCA ADAIMI, University of Texas at Austin, USA
EDISON THOMAZ, University of Texas at Austin, USA

A difficulty in human activity recognition (HAR) with wearable sensors is the acquisition of large amounts of annotated data for training models using supervised learning approaches. While collecting raw sensor data has been made easier with advances in mobile sensing and computing, the process of data annotation remains a time-consuming and onerous process. This paper explores active learning as a way to minimize the labor-intensive task of labeling data. We train models with active learning in both offline and online settings with data from 4 publicly available activity recognition datasets and show that it performs comparably to or better than supervised methods while using around 10% of the training data. Moreover, we introduce a method based on conditional mutual information for determining when to stop the active learning process while maximizing recognition performance. This is an important issue that arises in practice when applying active learning to unlabeled datasets.

CCS Concepts: • **Computing methodologies → Active learning settings**.

Additional Key Words and Phrases: Human Activity Recognition, Active Learning, Data Annotation, Stopping Criterion, Conditional Mutual Information

ACM Reference Format:
Rebecca Adaimi and Edison Thomaz. 2019. Leveraging Active Learning and Conditional Mutual Information to Minimize Data Annotation in Human Activity Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 3, Article 70 (September 2019), 23 pages. https://doi.org/10.1145/3351228

1 INTRODUCTION

With advances in mobile sensing and computing, human activity recognition (HAR) has continued to grow over the last decade, with applications in numerous areas such as abnormal human behavior detection [19], health monitoring and fitness tracking [14, 32, 42], daily routine monitoring [50], and many others [28, 29, 33, 36, 52]. Today, the ubiquity of mobile phones, smartwatches, and wearable sensors [24], make it possible to acquire large amounts of activity data with relative ease [26]. However, to reliably learn and recognize human activities from sensor data, vast amounts of labeled training data is typically required, which remains one of the main challenges of supervised methods in real-world studies. Providing high-quality and consistent ground truth labeling of vast amount of data has proven to be time-consuming and not always feasible or reliable. Several annotation techniques have been used for capturing ground truth in data collected in real-world settings, including ecological momentary assessment [43], and self-reported time-use diaries [20]. While these methods have merits, they are susceptible to biases and recall errors and are known to be disruptive, directly affecting and possibly changing

Authors' addresses: Rebecca Adaimi, University of Texas at Austin, 2501 Speedway, Austin, Texas, 78712, USA, rebecca.adaimi@utexas.edu; Edison Thomaz, University of Texas at Austin, 2501 Speedway, Austin, Texas, 78712, USA, ethomaz@utexas.edu.

2474-9567/2019/9-ART70 $15.00
https://doi.org/10.1145/3351228

the activities being recorded. Other ground truth acquisition methods that rely on the capture and subsequent analysis of audio or video have become more popular [25, 44]. While these methods might capture activities with higher fidelity and in an objective way, they introduce new challenges, such as privacy and ethical concerns.

In this paper, we explored Active Learning (AL) as a way to achieve a highly accurate model while minimizing the annotation effort and amount of labeled data needed. Active Learning is particularly well suited for HAR since sensor data may be abundant but ground truth labels are expensive to obtain. We experimented with existing AL approaches in both stream-based (online) and pool-based (offline) settings using 4 activity recognition datasets of varying tasks, sensors used, and collection strategies with very promising results. In AL applications, determining a stopping criterion is an important issue that arises in practice, since it is a trade-off between labeling cost and effectiveness of the classifier. Analyzing the information gain at every AL iteration, we further propose a stopping criteria based on conditional mutual information that stops the AL process once no additional informative data samples are left to query. The contributions of this work are:

- A study of pool-based and stream-based AL approaches on 4 established HAR datasets. Compared to standard supervised methods, we demonstrate that applying AL in HAR research leads to accurate comparable models using less training data, with some datasets requiring only 8% of data to achieve comparable performance.
- A proposed stopping criterion based on conditional mutual information that measures the information gained from an unlabeled pool of data during the active learning process.

2 ACTIVE LEARNING BACKGROUND

The main idea behind AL is that if a learning algorithm can choose the data it wants to learn, it could potentially perform the same or even better than standard supervised methods with significantly less training data. Thus, AL is a process in which a model only queries data that can add knowledge and improve performance. This concept has been popular in many data mining and machine learning applications because it helps in (i) reducing the annotation cost and manual labeling efforts and (ii) reducing model training computation time.

2.1 Sampling Strategies

There are three sampling strategies that have been considered in the literature in which a learning model can query instances: (i) *Membership Query Synthesis*, (ii) *Pool-based Selective Sampling*, and (iii) *Stream-based Selective Sampling*. Membership Query Synthesis [3] was one of the very first AL methods studied. In this scenario, the learning model generates a data instance from a certain distribution. However, a main limitation of this method is the inability to use human annotators for labeling since the queried samples are not sampled from real-world data, and thus can sometimes be challenging to recognize and label. To address this limitation, pool-based and stream-based selective sampling were proposed [4, 21]. A key assumption for these scenarios is that labelling an unlabeled instance is inexpensive. As such, instances can be selectively sampled from real-world data instead of synthesizing them. For pool-based selective sampling, the model has access to all the unlabeled data from which it selects the best. On the other hand, stream-based selective sampling scans unlabelled data sequentially and decides based on some querying strategy whether to issue a query for a sample or not. This can make finding the very best requests elusive. This scenario is typically suitable for online settings in which decisions are made on the fly and where memory or processing power is limited. In our paper, since we are dealing with real-world data, we evaluate both pool-based and stream-based selective sampling.

2.2 Query Strategies

For all AL scenarios, a query strategy is used to evaluate the informativeness of unlabelled data. The most commonly used query strategy is known as *Uncertainty Sampling* [21]. Using this strategy, the model queries

instances for which it is least confident about its most probable label. This strategy can be used for both pool-based and stream-based settings. However, in a multi-class application, this query strategy only considers information about the most probable label of a sample and discards information about the remaining labels. Therefore, *Margin-based Uncertainty Sampling* was proposed that takes into account the two most probable class labels [34]. The intuition for this strategy is that small margins indicate instances for which the model is uncertain of their label. Therefore, querying those instances for their true label would help the model better distinguish between the different classes. However, for batch-mode approaches that query groups of instances, the queried points might not be representative of the underlying distribution of the unlabeled data. Therefore, researchers studied diversity-based approaches that ensure diversity among instances in the batch of samples queried. Many algorithms have been proposed that consider informativeness as well as diversity when querying instances, such as adopting a Gaussian kernel to measure the similarity between any two instances [51], using a single linkage method used in hierarchical clustering [16], and maximizing the angles within a set of hyperplanes with SVM [8].

In our paper, we applied batch-mode selective sampling in the pool-based setting, and in order to ensure diversity, we employed a k-means cluster-based constraint to preserve the distribution of the unlabelled data. As for the stream-based scenario, Sculley explored several label-efficient schemes for spam filtering, such as *Logistic Margin Sampling*, *b-Sampling*, and *Fixed Margin Sampling* [35]. We applied the logistic margin sampling approach, that maps classification confidence values to sampling probabilities, as a query strategy for our stream-based selective sampling. As can be seen, there are several AL approaches that can be explored, but for the purposes of this paper, we start by exploring two commonly used AL frameworks on an extensive set of HAR datasets. More details about the frameworks are found in Section 4.

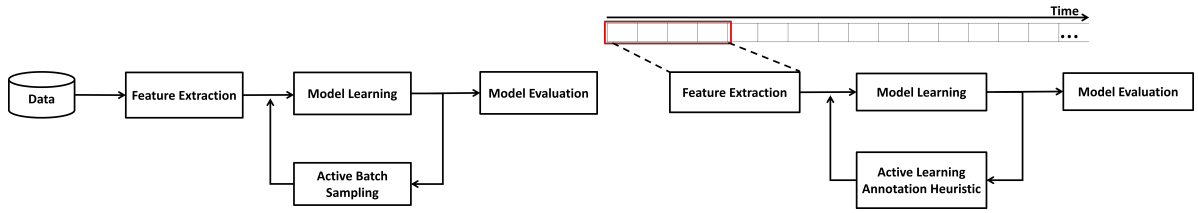
3 RELATED WORK

In HAR research focused on naturalistic settings, data annotation is often a time-consuming and burdensome process, and considered a key challenge [9]. Moreover, inertial sensor data, which is commonly used in HAR applications, is hard to interpret and annotate. Therefore, whether in controlled laboratory settings [5, 25, 38] or *in-the-wild* [44], researchers often employ other sensor modalities to assist the labeling process, such as video recordings. Methods that require direct and regular individual input, including daily self-recall and experience sampling [20, 43, 47] are popular and straightforward to implement, but are subject to biases and memory recollection errors. Generally, supervised learning methods require large annotated datasets for training, which requires human supervision for annotations, a process that is costly, non-objective and error-prone. [48].

In order to facilitate the data annotation process, tools that provide labeling recommendations based on already labeled data [45] or self-annotation tools [15] have been developed. Others have proposed automating the annotation process by implementing a knowledge-driven method using weak labels thus enabling an online supervised training approach [13, 39]. Alternatively, going beyond fully supervised techniques, semi-supervised approaches have been considered as a solution to reducing the need for labelled data. This approach requires only a small set of labelled data and a large set of unlabelled data [40]. However, semi-supervised approaches use the most confident predictions of unlabeled samples and adds them to the training set. One drawback to this is the case when a wrong prediction is added to the training set, which decreases the accuracy of the classifier. As opposed to semi-supervised methods, AL does not use predictions as labels, but rather detects the most informative unlabeled samples and queries the user for a label.

3.1 Active Learning in HAR

A large body of work has investigated the use of AL in HAR applications. Many studies focused on pool-based AL for offline applications. For example, Stikic et al. explored semi-supervised and AL to reduce the amount of labeled training data required [41]. Rebetez et al. proposed a system that learns activities by actively querying users using



(a) Pool-based Active Learning Framework (b) Stream-based Active Learning Framework

Fig. 1. Active Learning Frameworks

a Growing Neural Gas algorithm (GNG) and tested it on the Opportunity dataset [30]. Longstaff et al. explored various active and semi-supervised learning methodologies to improve the performance of an initial classifier [23]. Alemdar et al. and Bagaveyev et al. explored different AL methods on datasets collected in home settings, showing a reduction in the number of labeled data required [2, 6]. Liu et al. presented preliminary exploration of pool-based AL for multi-sensor physical activity recognition on one dataset [22]. Overall, compared to prior work, we present a more in-depth exploration of the use of pool-based AL on a wider set of HAR data.

Stream-based AL has also been used in the context of HAR. Abdallah et al. implemented a clustering-based AL method to predict activities, and allowed for multiple queries to be sent to the user which could lead to user disruption [1]. Shahmohammadi et al. leveraged smartwatches to implement an AL framework that provides continuous activity recognition monitoring [37]. In Zliobaite et al., the authors presented a new AL framework in data streams that deals with concept drift [49]. Miu et al. implemented a smartphone activity recognition framework that included a stream-based segmentation method [27]; the system required a large annotated dataset to generate the hyperparameters for their AL model. Similar to our work, they tested a stream-based approach on the Opportunity dataset [10]. However, their approach included an ideal segmentation procedure for each activity, and data segments that were not annotated were replayed and inputted again for potential querying. We implemented a similar stream-based approach and expanded our evaluation by testing on multiple different datasets.

4 ACTIVE LEARNING FRAMEWORKS

In this paper, one of our aims is to validate two AL frameworks, pool-based and stream-based, on HAR datasets. The specific goal is to show their effectiveness at reducing the amount of labeled data needed to reach comparable performance to the standard fully supervised approaches. Both pool-based and stream-based frameworks consist of multiple stages: (1) feature extraction, (2) model learning, (3) active batch sampling or active learning annotation heuristic, and (4) model evaluation, as depicted in Figure 1. In this work, every stage of the framework is applied in a way that is specific to each dataset in order to reliably compare active vs. supervised learning approaches. In this section, we discuss in detail the active batch sampling and the annotation heuristic used in the pool-based and stream-based frameworks respectively. The remaining stages are discussed in more detail in Section 5.

4.1 Pool-based Active Learning

In this form of AL, training is achieved with a small randomly selected labeled subset of the data. The initial batch size is a hyperparameter that is defined by the user. After fitting the model to the labeled data, the AL sampling function is applied to the pool of unlabeled data from which a batch is sampled and added to the training data. The model is then updated and the process repeats. The sampled batch size is also a hyperparameter determined by the user. Figure 2 shows one iteration of the AL process. The active learning approach implemented is an

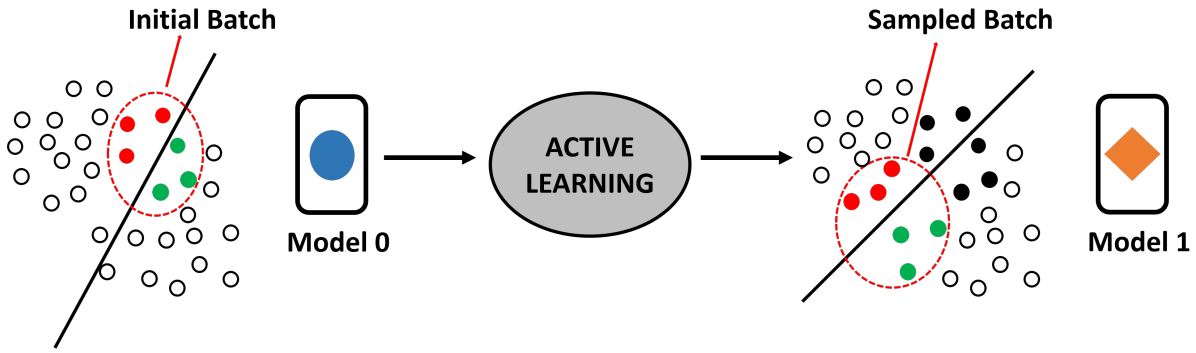


Fig. 2. Illustration of one iteration of pool-based AL. Red and green denote two classes. Model 0 is the initial model trained on the initial labeled batch (colored dots). One iteration of AL queries a batch (colored dots) and adds it to the training data. Model 1 is the updated model trained on the new training data. Black dots are previously labeled data, while blank dots are remaining unlabeled data.

informative and diverse approach [1] that employs margin-based uncertainty sampling [34] as an informative measure while preserving the same data distribution over clusters. This is done using a cluster-based sampling scheme that tries to select the most representative data points according to a data distribution constraint imposed using a clustering algorithm. More formally, assuming we are dealing with a classification problem of two classes c_1 and c_2, let the posterior probability of each class be denoted by P(c_1|x) and P(c_2|x). The first sampling function computes the margin of data points from the pool of unlabeled training data and sorts them in increasing order. Note that, in the case of multiclass classification which is common in complex activity recognition tasks (e.g., ADLs), the two most probable classes are used to compute the margin. The margin sampling function is as follows:

$$\underset{x}{argmin}(P(c_1|x) - P(c_2|x)) \tag{1}$$

Simply selecting the top samples with the least margin can lead to redundancy in the selected batch thus minimizing the diversity in the samples selected. In order to improve diversity and reduce redundancy, a diversity criterion is imposed using a clustering method. A batch is created by sorting unlabeled data samples by increasing the margin and then growing the batch greedily. A sample is added to the batch if the resulting batch maintains the same distribution over clusters as the entire training data. In our evaluation, we assume that the number of activities is known beforehand, which is usually the case with predefined classification problems. Thus, we set the number of clusters according to the number of classes. However, it is important to note that the cluster labels do not necessarily match the externally-supplied class labels. Thus, imposing this clustering constraint does not necessarily ensure that at least one sample from each class is queried, but it ensures that the queried batch does not include redundant samples belonging to one cluster [17]. Algorithm 1 shows the pseudo code of the pool-based active learning algorithm.

4.2 Stream-based Active Learning

The stream-based AL approach we employed follows a pipeline that is similar to the one implemented in [27] but without any activity segmentation. As depicted in Figure 1b, to simulate a continuous stream of data, we first apply a sliding window approach and extract features for each frame. After the feature extraction step, the model

[1] https://github.com/google/active-learning

ALGORITHM 1: Pool-based Active Learning Algorithm

Input: Unlabeled data (X), model ($model$) trained on labeled subset of data, batch size (N), and number of activities ($n\_clusters$).
Output: A batch of size N containing indices of selected samples.
$model\_cluster = KMeans\ clustering\ model\ using\ n\_clusters$;
$cluster\_prob = number\ of\ samples\ per\ cluster\ divided\ by\ total\ number\ of\ samples$;
$predict\_prob = predicted\ class\ probabilities\ of\ X\ using\ model$;
if $classes < 2$ **then**
    $min\_margin = abs(predict\_prob)$
else
    $min\_margin = difference\ between\ the\ two\ most\ probable\ classes$;
end
$rank\_ind = argmin(min\_margin)$;
$new\_batch = []$;
for *each index i in rank_ind* **do**
    if *length of new batch* $== N$ **then**
        break;
    end
    $label_i$ = label extracted from $model\_cluster$ at index i;
    if $\frac{\#\ of\ data\ points\ queried\ with\ label_i}{N} < cluster\_prob[label_i]$ **then**
        append index i to new_batch;
    end
end

(initially trained on a small batch of labeled data) classifies frames generating a predicted class probability p_{pred} for each activity class. Thus, for a multiclass problem, the highest class probability is considered as a measure of classification confidence, following Equation 2. Implementing the *Logistic Margin Sampling* as in [27], we use the classification confidence generated for a frame and compute the probability of requesting a label for that frame (Equation 3).

$$p_{conf} = max(p_{pred}) \tag{2}$$

$$p_{req} = exp(-\gamma \cdot p_{conf}) \tag{3}$$

Algorithm 2 illustrates the active learning algorithm used [27]. The tunable hyperparameter γ controls the querying behavior thus affecting the number of annotation requests. We examine the change in the performance of the stream-based AL approach when varying γ in our experiments. Similar to [27, 35], we implement a randomized method for requesting a label. In order to generate a decision for requesting an annotation, we generate a random threshold from a uniform distribution between [0,1]. The frame label is requested when p_{req} exceeds the threshold.

5 DATASETS

To reiterate, our aim is to study whether AL can reduce the amount of labeled data needed in HAR if compared to supervised learning methods. To that end, we evaluate the pool-based and stream-based frameworks on 4

ALGORITHM 2: Stream-based Active Learning Algorithm

Input: Hyperparameter (γ) used in computing the asking probability, model for classification (*model*), and the features of j^{th} frame (f_j) in the stream of data
Output: boolean decision (h_{req}) to request a label for frame f_j
$p_{pred} = model.predict(f_j)$
$p_{conf} = max(p_{pred})$
$p_{req} = exp(-\gamma \cdot p_{conf})$
$threshold = uniform([0, 1])$ # sampling a random threshold
$h_{req} = threshold < p_{req}$

Table 1. Summary of Datasets

Dataset	Activity	Model	Evaluation Metric
Opportunity Dataset [10]	Locomotion and gestures	k-Nearest Neighbor (k=3)	weighted F-measure
ExtraSensory Dataset [47]	Behavioral activities	Logistic Regression w/ balanced class weights	balanced accuracy
Fluid Intake Dataset [12]	Fluid intake	Random Forest Classifier (n=185)	precision, recall, and F-measure
PAMAP2 Dataset [31]	Physical activities	Random Forest Classifier (n=100)	precision, recall, and F-measure

commonly used activity recognition datasets that characterize different types of behaviors: locomotion and gestures (Opportunity [10]), behavioral context recognition (ExtraSensory [47]), fluid intake detection (Fluid Intake [12]), and physical activities (PAMAP2 [31]). For every dataset, we followed the frameworks in Figure 1 with each stage specific to every dataset. The *active batch sampling* and the *active learning annotation heuristic* processes detailed in Section 4 are common for every dataset. Table 1 summarizes the key elements of every dataset. We were successful in replicating the original evaluation pipeline for all datasets except for PAMAP2, explained in more detail in the following subsections. As for the evaluation metrics, we used the same approach employed in the original evaluation of every dataset. In order to evaluate the performance of each AL technique, we compared a fully supervised approach using the whole training datasets against the AL approaches.

5.1 Opportunity Dataset

The Opportunity dataset [10] is a public state-of-the-art HAR dataset. It comprises the readings of motion sensors recorded while 4 subjects executed typical daily activities. From the Opportunity challenge, we followed the baseline approach and replicated two tasks of the activity recognition problem: (1) classifying the 4 modes of locomotion using the body-worn sensors, and (2) recognizing the 17 different right-arm gestures. We replicated the evaluation setup using the same sliding window for feature extraction process, k-Nearest Neighbours (3-NN) model, evaluation metric, and train/test split used for the baseline performance.

5.2 Fluid Intake Dataset

The fluid intake dataset is a dataset compiled in a laboratory study with 30 participants for fluid intake detection [12]. It includes a variety of realistic everyday activities and gestures captured using accelerometer data from inertial sensors in wearable wristbands. Following the processing pipeline in [12], we replicated the fully supervised Leave-One-Participant-Out (LOPO) evaluation on the lab data where a random forest classifier with 185 trees was used for classifying fluid intake instances.

5.3 PAMAP2 Dataset

The PAMAP2 Physical Activity Monitoring dataset contains 18 different physical activities captured using 3 inertial measurement units and a heart rate monitor from 9 subjects. Our implementation modeled 12 activity classes as defined in [31]. A replication package for this dataset was not provided. In our attempt to replicate the original implementation, we were unable to accurately compute the features used and reach the same performance using a k-NN model. Thus, we implemented our own evaluation by extracting a subset of some of the features extracted in the original work and modeled the data using a random forest. Data from 3 IMUs (placed on the chest, dominant arm, and dominant ankle) and from a heart rate monitor were used. The data was segmented using a sliding window with 5.12 seconds window size and 1 second overlap, similar to the original implementation. From the segmented 3D-acceleration data collected from the IMU placed on the arm, features in both time and frequency domain were calculated. The extracted features included: mean, median, standard deviation, peak acceleration, energy, absolute integral, correlation between each pair of axes, power ratio of the frequency bands 0-2.75 Hz and 0-5 Hz to the total power, peak frequency of the power spectral density, and spectral entropy of the normalized power spectral density. These features were extracted for each axis separately, and for the 3 axes together. Moreover, the features mean, standard deviation, absolute integral, and energy were calculated on each axis of the weighted pairwise combination of the 3 IMUs (ankle + chest, ankle + arm, chest + arm) as well as a weighted sum of all three IMUs. From the heart rate data, we computed the mean of the data and its gradient. Therefore, a total of 85 features were extracted from each data segment. Using k-NN as in the original implementation resulted in very low performance. Therefore, we explored random forest since it has been known to perform well on activity recognition datasets. Thus, by tuning the number of trees, we were able to achieve better performance. We evaluated the model using the LOPO approach, and computed results using common metrics such as precision, recall, and F-measure.

5.4 ExtraSensory Dataset

The ExtraSensory dataset is a public dataset for behavioral context recognition *in-the-wild* from mobile sensors [46]. Containing data from 60 users, the dataset consists of measurements from five sensors in a smartphone: accelerometer (Acc), gyroscope (Gyro), location (Loc), audio (Aud), and phone-state sensors (PS), as well as accelerometer measurements from a smartwatch. The authors evaluated the dataset by classifying 25 labels from different context domains. In our attempt to replicate their evaluation, our AL framework encountered missing classes in the initial labeled data for 3 of the classes (i.e., On a bus, Shopping, Drinking (alcohol)). We excluded these classes from the analysis and performed our tests with 22 activities in total. The authors addressed different approaches to fuse information from different sensor modalities; we replicated two of these approaches: (1) the single-sensor classifiers approach where, for a given context label, classification is done based on each sensor independently, and (2) the early fusion approach where features from multiple sensors are concatenated before classification (EF). As in the original implementation, we used logistic regression with balanced class weights and evaluated using balanced accuracy by splitting the 60-users data into 48 users for the train set and 12 users for the test set.

6 EXPERIMENTS AND RESULTS

For each dataset, we evaluated the fully supervised (on all available training data) and the AL approaches (both pool-based and stream-based using a subset of the training data). The aim was to study whether the AL methods yielded improved or comparable performance to a fully supervised model. In the following experiments, the initial batch size and the pool-based queried batch size were empirically set to 2%. We show the effect of tuning these hyperparameters on performance in Section 6.1.6. Source code for the analysis is available at https://github.com/radaimi/leveraging-AL-and-CMI-in-HAR.git.

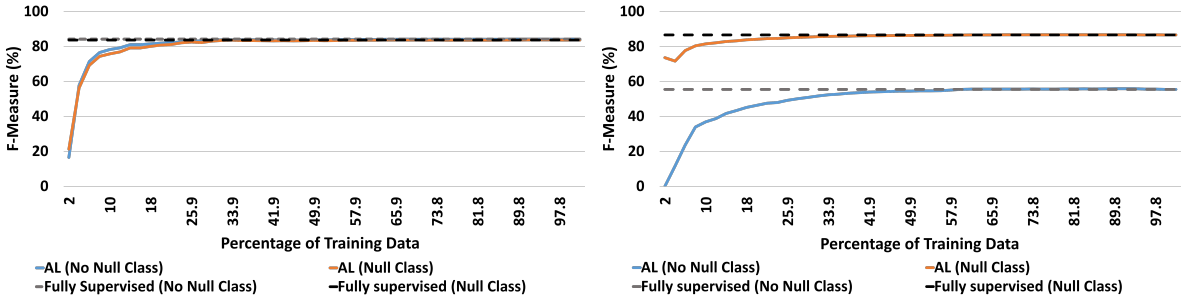


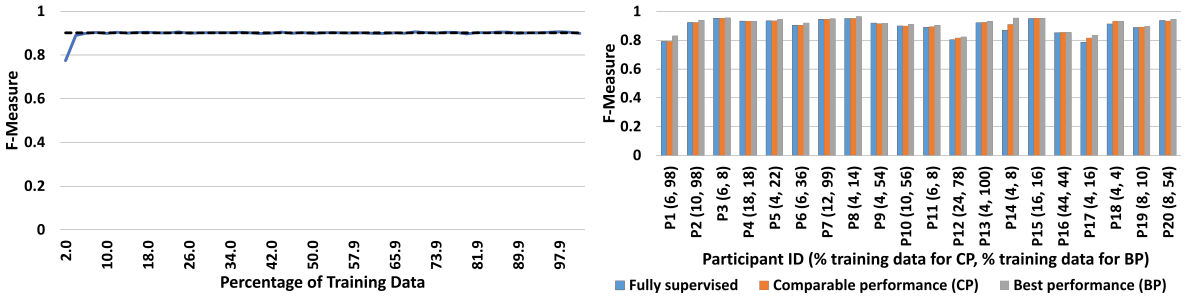
Fig. 3. Learning Curves of average F-measure of Pool-based Active Learning on Opportunity Dataset for (a) Locomotion and (b) Gesture Recognition. Including and not including the Null class in the F-measure, the AL reaches comparable performance to the fully supervised using considerably less training data.

6.1 Pool-based Active Learning

6.1.1 Results for Opportunity Dataset. We trained user-specific classifiers for modes of locomotion and gesture recognition, as explained in Section 5, and measured the weighted F-measure—either including or not including the *Null* class. Figure 3 shows the learning curves of the F-measure averaged over the LOPO evaluation of every subject. Applying pool-based AL, we observe a gradual increase in performance for both locomotion and gesture recognition tasks as data samples are queried and added to the training data. Looking at Figure 3, we observed that comparable performance to the fully supervised approach was reached using only ~10-20% of the data for locomotion recognition for both cases—including and not including *Null* class. As for gesture recognition, performance had a slower increase with added training data. Given the large number of *Null* class samples, not considering the *Null* class in the F-measure, we observed a slower increase in the learning curve, thus requiring ~40-50% of the training data to reach comparable performance as the fully supervised. However, including the *Null* class leads to a significant increase in F-measure and also shows that slightly less data is needed (~20-30%) to reach comparable performance.

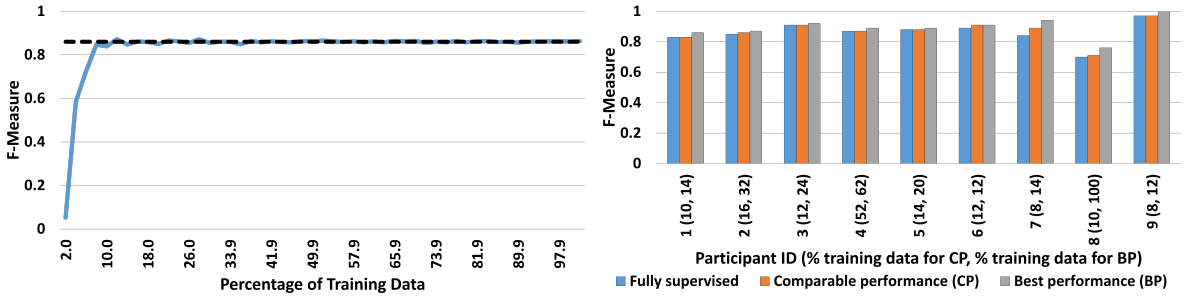
6.1.2 Results for Fluid Intake Dataset. Implementing a fully supervised approach for baseline comparison, we obtained an average precision of 90.21%, average recall of 90.91%, and average F-measure of 90.18% with LOPO cross-validation. Figure 4a shows the average F-measure over all 30 participants when applying the pool-based AL approach for batch sampling. Comparing to the fully supervised baseline performance, we observed that we are able to reach comparable and even slightly better performance (90.35% F-measure) using only 8% of the training data. One important observation is that, when employing pool-based AL, we not only achieved comparable performance to the fully supervised approach but also, in some cases, higher performance. We hypothesized this occurred due to the effectiveness of the AL framework in choosing samples that more optimally discriminate the target classes; we plan to study this finding more deeply in future work. To better visualize the described behavior for the different LOPO evaluations, Figure 4b shows the F-measure of the fully supervised learning for every participant as well as the comparable F-measure reached using the least amount of training data and the maximum F-measure reached during the pool-based AL process. The x-axis included the percentage of training data used for each case for every participant. For legibility, we visualized the results for a subset of 20 participants.

6.1.3 Results for PAMAP2 Dataset. With PAMAP2, the fully supervised performance resulted in an average F-measure of 86%, average precision of 88.2%, and average recall of 86%. Similar to the fluid intake dataset, we



(a) (b)

Fig. 4. Results of Fully Supervised and Pool-based Active Learning Approaches on Fluid Intake Dataset. (a) Learning Curve of pool-based AL (black dotted line is the fully supervised F-measure) and (b) Bar plot of F-measure per participant of pool-based and fully supervised baseline approaches. The x-axis labels include in parenthesis the percentage of the training data used to achieve the corresponding comparable performance to the fully supervised and the best performance.

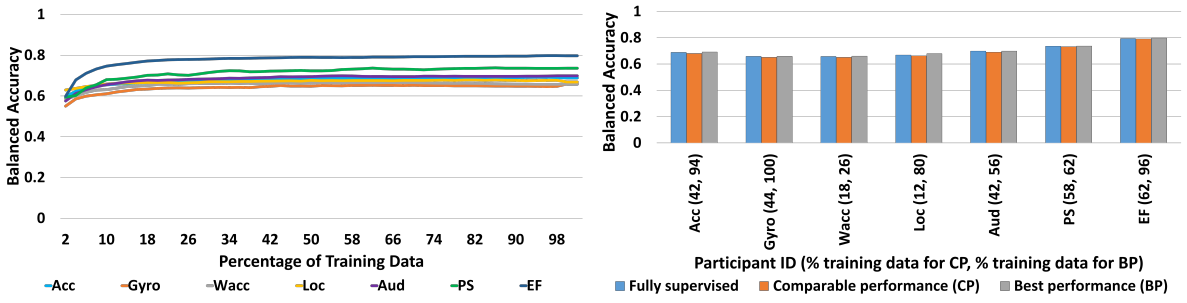


(a) (b)

Fig. 5. Results of Fully Supervised and Pool-based Active Learning Approaches on PAMAP2 Dataset. (a) Learning Curve of pool-based AL (black dotted line is the fully supervised F-measure) and (b) Bar plot of F-measure per participant of pool-based and fully supervised baseline approaches. The x-axis labels include in parenthesis the percentage of the training data used to achieve the corresponding comparable performance to the fully supervised and the best performance.

employ a LOPO evaluation and visualize the average F-measure performance as more training data is sampled using the pool-based AL approach. This generated the learning curve depicted in Figure 5a. We observed that, using only 8% of the training data which is around 300-400 data samples, the model was able to achieve comparable performance to the fully supervised approach with 84.6% F-measure. Training with 12% of the data resulted in slightly better performance than the fully supervised (89.4% precision, 87% recall, and 87.2% F-measure). Applying similar analysis as in the fluid intake case, we observed the performance results of the LOPO evaluation 5b. For almost all subjects, a slightly higher performance than the fully supervised can be achieved using less training data.

6.1.4 Results for ExtraSensory Dataset. As discussed in Section 5, we implemented two approaches: (1) single-sensor classifiers for each of the 5 sensors—Accelerometer (Acc), Gyroscope (Gyro), Watch Accelerometer (WAcc), Location (Loc), and Audio (Aud), as well as Phone State (PS), and (2) the early fusion (EF) approach where features of different sensors were concatenated. We model each of the 22 context labels separately using a logistic



(a) (b)

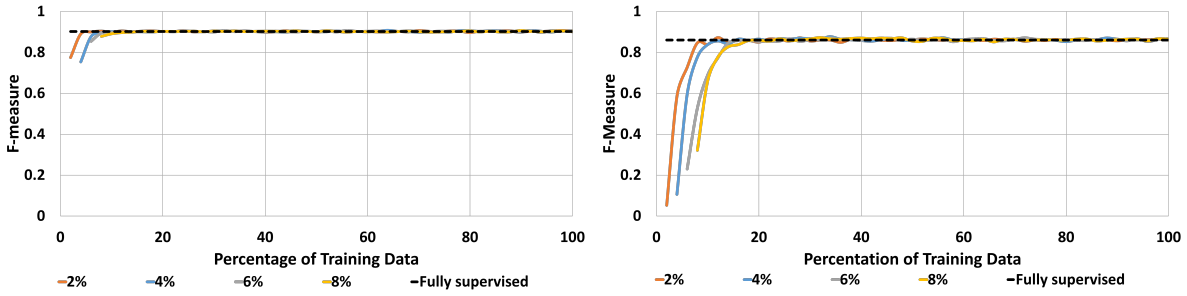
Fig. 6. Results of Fully Supervised and Pool-based Active Learning Approaches on ExtraSensory Dataset. (a) Learning Curve of pool-based AL (black dotted line is the fully supervised F-measure) and (b) Bar plot of balanced accuracy per classifier of pool-based and fully supervised baseline approaches. The x-axis labels include in parenthesis the percentage of the training data used to achieve the corresponding comparable performance to the fully supervised and the best performance.

regression and evaluate over the test set. For each of the single-sensor and early fusion approaches, we first implemented a fully supervised approach by training on the whole training set to get a performance baseline. Figure 6a shows the learning curve of the average balanced accuracy measure over all 22 context labels for every sensor classifier and the EF classifier when applying the pool-based AL. The bar plot in Figure 6b provides the exact amount of training data required to reach the fully supervised performance (comparable performance) or even surpass it (best performance). Despite the fact that the percentage training data recorded is in some cases large, looking at Figure 6a, we can observe that the performance was already approximately equal to the fully supervised (blue bars in Figure 6b) with a difference not more than ~0.02 in balanced accuracy when training on ~10-20%.

6.1.5 Impact of Data Collection Strategy on Performance. As the previous sections showed, utilizing AL for model training led to different results for each dataset. These performance differences likely occurred due to the specific characteristics of each dataset. Both PAMAP2 and Fluid Intake datasets were collected in a controlled environment with participants following a specific script of activities. As for the Opportunity dataset, data collection occurred in a room simulating a studio flat in which participants asked to follow a high-level script of activities while still allowing them some freedom in performing the activities. Therefore, the Opportunity dataset is more diverse if compared to the datasets collected in a more controlled setting, i.e., Fluid Intake and PAMAP2.

The ExtraSensory dataset, which is almost 10× as big as the other datasets, was collected *in-the-wild* with users engaging in regular natural behavior. With only 2% of the data, the pool-based AL method matched the performance of the fully supervised approach. We attribute this finding to the distribution of the dataset; the initial batch of 2% was likely large and diverse enough to produce effective predictive models. This observation can also be confirmed with the stream-based AL results in Section 6.2.5. Starting with 2% initial batch, the stream-based AL framework ended up querying not more than 2% with some context activities and sensors resulting in comparable and even higher performance than the fully supervised (refer to Table 2b).

These observations suggest that AL can be effective in minimizing the amount of labeled data when it originates from controlled or field studies. However, for *in-the-wild* experiments, our findings indicate that the dataset should be at least as large as ExtraSensory so that the pool-based AL method matches the performance of the fully supervised approach. Since it might not be always possible to collect extensive amounts of data in free living conditions, an alternative data collection strategy is to capture data in semi-controlled experiments, which



(a) (b)

Fig. 7. Learning Curves of (a) Fluid Intake and (b) PAMAP2 datasets for varying size of the initial labeled batch. For both datasets, using 2% initial labeled batch leads to a faster increase in learning curve.

combine elements of laboratory and field studies [7, 11]. A hybrid study design, similar to one that produced Opportunity, can yield data that is conducive to AL: more deterministic but also diverse and representative of behaviors expressed in real-world settings.

6.1.6 Hyperparameter Tuning. In order to implement the pool-based AL, two hyperparameters needed to be defined: (1) initial labeled batch size and (2) sampled batch size. As was shown in Figure 2, we started by forming the initial labeled batch by picking the first samples in the dataset to train an initial model. The size of this labeled pool of data was determined by setting the corresponding hyperparameter. We observed the effect of varying this parameter on the performance curve of the pool-based active learning method compared to the fully supervised approach. Note that while varying the initial labeled batch size, we empirically fixed the sampled batch size to 2%. Figure 7 depicts the average F-measure results of the LOPO evaluation for PAMAP2 and fluid intake datasets. Similar behavior was observed for the remaining datasets. It was observed that, using an initial labeled batch of 2% for both Fluid Intake and PAMAP2 datasets, the fully supervised performance was reached with less training data, as opposed to using a larger initial batch. Training the initial learner on a very small set of labeled data resulted in low performance. Intuitively, training on more data improves the model's performance. However, sampling more training data using the AL approach ensures that the most informative data points are being added which should lead to a faster improvement in performance as opposed to random sampling or simply taking the first samples in the data. However, it is worth noting that it could be the case that the initial batch already contains informative data. This behavior is observed in both Fluid Intake and PAMAP2 datasets. For PAMAP2, starting with 2% initial labeled batch and then sampling using pool-based AL with 2% batch, the fully supervised performance of 86% was achieved using only 8% of the data. On the other hand, if the initial learner was trained on 8% initial batch, the performance was only 32%. Similarly for the Fluid Intake dataset, starting with 2% initial batch, the fully supervised performance of 90% was achieved using only 6% of the data, whereas the initial learner trained on 6% resulted in a slightly lower performance of 85%.

As previously mentioned, another hyperparameter is the size of the batch queried by the AL framework. Fixing the initial starting labeled batch at 2%, we vary the size of the batch sampled by the pool-based AL approach and observe the performance curve for every dataset. Figure 8 shows the effect of varying the batch size on the performance curve of the Fluid Intake and PAMAP2 datasets. We can observe that using batch sizes 0.5%, 1%, and 2% result in faster convergence to the fully supervised performance compared to larger batch sizes. However, using a very small batch size (0.5% and 1%) leads to slight fluctuations in the performance as opposed to using 2% which resulted in a smoother curve. Moreover, using a smaller batch size increases the number of AL iterations

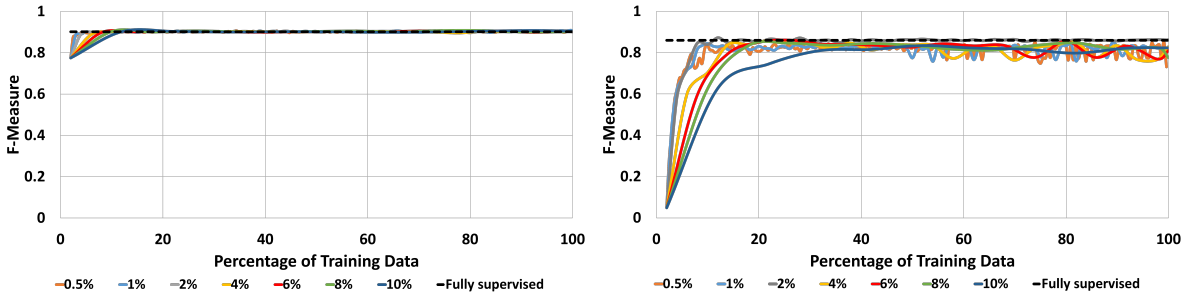


Fig. 8. Learning Curves of (a) Fluid Intake and (b) PAMAP2 datasets for varying size of the sampled batch in pool-based AL. For both datasets, sampling a batch of size 2% leads to a faster and more stable increase in learning curve.

needed to reach comparable performance. Recall that our pool-based AL approach samples data based on the uncertainty of the model in its predictions. Thus, while the algorithm tries to select a batch of samples with the highest uncertainty measure, the number of uncertain data samples could be less than the defined batch size, therefore leading the algorithm to select remaining uninformative samples to fill the batch. This would explain the slower increase in performance when using a large batch size.

6.2 Stream-based Active Learning

6.2.1 Querying Probability. The stream-based method is simulated by inputting every data sample as it is received to the annotation decision method that decides whether to query the sample for annotation or not. As each new data sample is annotated, we evaluated the performance of the model against the test set for the corresponding subject. As described in Section 4.2, the *active learning annotation heuristic* computes a probability p_{req} for which an annotation request is issued. The probability of querying a sample is computed as a function of a tunable parameter γ. γ controls the querying behavior as follows: for fixed confidence probability p_{conf}, when γ increases, the request probability decreases, thus resulting in fewer queries. Setting a fixed γ, if the model's confidence probability is high, with high γ, the samples in question are more likely to be ignored, and so queries will be more directed towards samples with low confidence. Thus, when running our experiments, there was always this tradeoff of setting γ large enough to reject, with high probability, samples that are relatively confidently classified, and small enough to not query samples extremely frequently thus allowing simulations to complete in reasonable time. In the following sections, we observe results of varying γ on some datasets.

6.2.2 Results for Opportunity Dataset. In this experiment, we explored the effect of varying γ on the amount of samples queried and the performance reached. Using $\gamma = 6$, we observed a gradual increase in performance for locomotion recognition as data samples are queried. The process ended up sampling ~3% of the data with performance lower than the fully supervised performance (around an average difference of 12% in F-measure for both including and not including Null Class between AL performance and fully supervised performance). On the other hand, there was very little increase in performance for gesture recognition with a significant gap in performance when not including the Null class. In order to observe whether comparable performance can be reached with more queried samples, we reduced the hyperparameter γ to 4. For locomotion recognition, the learning curve of F-measure performance as data samples were added gradually increased for each subject, showing an improvement in model performance (Figure 9). An interesting result observed in the learning curves for locomotion recognition was a significant increase in performance when one specific data sample was queried.

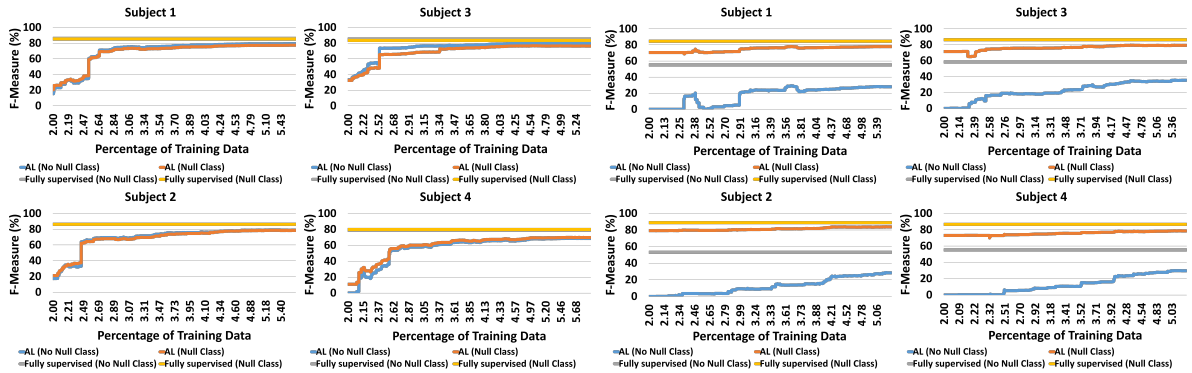


Fig. 9. Learning Curves of Stream-based AL ($\gamma = 4$) on Opportunity Dataset for (a) Locomotion and (b) Gesture Recognition. Around 5% of training data was sampled with locomotion performance using AL almost reaching the fully supervised performance. Gesture recognition had a slower learning curve due to class imbalance, in particular the inclusion of the Null class.

The fully supervised performance value, where the model was trained using all training data, acted as an upper performance baseline. The system ended up querying ~5-6% of the data. Compared to the initial experiment using $\gamma = 6$, the performance reached was higher but still was not able to reach the fully supervised. The average difference between the AL and fully supervised performance was ~7% in F-measure for both including and not including the Null class. On the other hand, there was a slower increase in the model performance for gesture recognition, indicating that the data samples considered informative by the annotation heuristic method did not cause a significant improvement in the model learning process. Including the Null Class, the AL learning curve showed slight improvements in F-measure as samples were queried, eventually reaching a performance gap of around 5-6% between AL and fully supervised. However, when not including the Null class, the learning curve showed slow but visible improvements in performance which indicated that, due to class imbalance, actively querying samples was improving the model's classification of the gesture classes. Further reducing γ should result in more queried data samples and the possibility of reaching the fully supervised performance but at a high computational cost due to the need to retrain the k-NN model whenever a sample is queried.

6.2.3 Results for Fluid Intake Dataset. We simulated a data stream by inputting the training data of 29 participants as one data stream, and applied the annotation heuristic on each incoming sample. When a data sample was added for labelling, we evaluated the model against the test data corresponding to the remaining participant. This process was repeated for every participant. We varied γ, observed the total percentage of data queried at the end of the AL process, and evaluated on the test data. Figure 10a compares the fully supervised F-measure to the reached performance when applying AL with varying γ. The data labels above each bar corresponds to the percentage of training data queried. For visualization purposes, only 9 randomly selected participants from all 30 participants were plotted. We can clearly see the effect of γ on the percentage of data queried and the F-measure performance. For some participants, performance using $\gamma = 2$ was comparable to and even higher than the fully supervised (e.g., participants 1, 5, and 9) using only ~15-16%. For participants 1 and 6, using $\gamma = 4$ resulted in higher performance than fully supervised using only ~4%. One interesting observation is that performance does not always exhibit a monotonic increase as samples are queried. Thus, even though the AL framework ends up

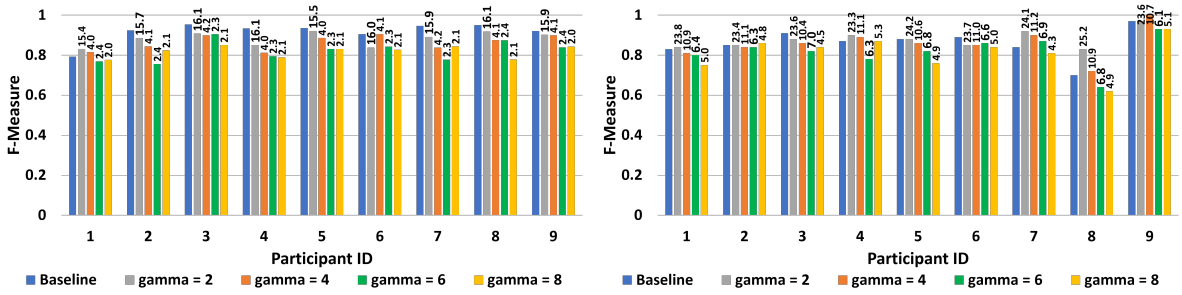


Fig. 10. Bar plot showing the end performance reached per participant using stream-based AL on (a) Fluid Intake and (b) PAMAP2 datasets for different γ values. Data labels on each bar shows the final percentage of training data that resulted in the plotted performance. Note that Fluid intake dataset included 30 participants, but for visualization purposes, we plotted performance of 9 random participants.

querying a certain number of data points, this does not mean the best performance was reached at the end. In some cases, higher performance was achieved using less data.

6.2.4 Results for PAMAP2 Dataset. Applying a similar approach as the Fluid Intake dataset, we evaluated the model against each subject data while training on the remaining subjects as training samples were annotated based on the annotation heuristic. For most participants, using $\gamma = 2$ resulted in comparable and even higher performance than the fully supervised using only ~23% of the training data. Even increasing γ to 4 still resulted in comparable performance indicating that less data (~11%) can still achieve good performance. As we kept increasing γ, performance for most participants dropped either slightly or significantly, with some participants still achieving the fully supervised performance using only ~4-5% of the data (e.g., participants 2,4, and 7).

6.2.5 Results for ExtraSensory Dataset. For this dataset, stream-based AL was implemented for each classifier corresponding to a specific sensor and a context label. Thus, for each sensor-label pair, a certain number of training samples were queried, which led to different results for each case. In order to observe the performance of the stream-based approach, we had to observe the results for each context label and each sensor system (single-sensors and EF). To that end, we evaluated the performance for 22 labels separately for each of the single-sensor classifiers and the EF classifier in the fully supervised case (Table 2a) and the stream-based case (Table 2b). For all labels and sensors, γ was set to 6 as a reasonable tradeoff between being able to conduct the experiment and the high computational cost incurred due to the size of the dataset. This resulted in sampling around 2-5% of the training data for each case. For some context labels and classifiers, comparable and even better performance was achieved using $< 5\%$ of the training data. In some cases, performance fluctuated as data samples were added, indicating that queried data samples do not always improve the model performance. The tabulated results represent the last model performance reached after all queried samples were added.

6.2.6 Computational Cost of Stream-based AL. An important criteria for stream-based selection is that the decision to request and receive a label be made in real-time as data unfolds sequentially as a stream. In this paper, we conducted simulations of online AL to demonstrate the capabilities of the stream-based AL approach. At every iteration, the algorithm should be capable of deciding to request a label, receive a label, update the model, and evaluate the model on a holdout test set in real-time. Thus, there are multiple factors to consider that can affect the runtime. The logistic margin sampling is fast: it simply consists of computing the querying probability

Table 2. Results on ExtraSensory Dataset

Labels	Acc	Gyro	WAcc	Loc	Aud	PS	EF
Lying Down	73.15 %	68.54 %	66 .50 %	64.71 %	76.29 %	84.74 %	88.68 %
Sitting	64.34 %	61.40 %	62.55 %	61.19 %	63.66 %	71.44 %	77.88 %
Walking	77.12 %	79.68 %	70.46 %	57.25 %	65.84 %	68.97 %	80.12 %
Running	73.54 %	77.22 %	74.45 %	70.65 %	68.31 %	54.88 %	69.13 %
Bicycling	81.54 %	79.77 %	61.39 %	77.73 %	75.11 %	78.34 %	81.97 %
Sleeping	74.22 %	70.37 %	66.88 %	60.59 %	76.65 %	86.21 %	89.08 %
Lab work	64.75 %	70.09 %	64.65 %	81.36 %	67.14 %	79.15 %	83.84 %
In class	60.92 %	52.87 %	60.18 %	70.93 %	74.37 %	75.96 %	83.80 %
In a meeting	62.38 %	60.05 %	54.54 %	70.82 %	81.52 %	68.44 %	80.23 %
At main workplace	61.29 %	52.99 %	51.49 %	75.89 %	63.52 %	80.63 %	80.55 %
Indoors	79.69 %	78.29 %	71.88 %	57.72 %	66.28 %	75.05 %	84.23 %
Outside	78.99 %	77.96 %	69.38 %	58.84 %	65.39 %	74.65 %	83.49 %
In a car	85.65 %	62.18 %	71.90 %	86.11 %	79.64 %	87.01 %	88.05 %
Drive (I'm the driver)	86.20 %	61.79 %	77.85 %	88.33 %	78.23 %	84.87 %	90.26 %
Drive (I'm a passenger)	70.74 %	68.66 %	64.83 %	73.70 %	70.28 %	73.25 %	73.60 %
At Home	64.15 %	57.44 %	57.62 %	69.61 %	72.68 %	71.60 %	77.79 %
At a Restaurant	64.13 %	63.32 %	66.02 %	52.69 %	76.99 %	74.80 %	81.49 %
Phone in pocket	68.72 %	66.04 %	64.21 %	63.75 %	70.19 %	77.13 %	79.52%
Exercise	61.33 %	53.02 %	55.88 %	64.74 %	60.49 %	66.42 %	78.80 %
Cooking	54.36 %	66.71 %	70.34 %	50.97 %	63.66 %	63.60 %	63.20 %
Strolling	52.87 %	66.03 %	70.62 %	62.31 %	57.15 %	68.73 %	65.85 %
Bathing-Shower	55.00 %	55.53 %	73.12 %	51.06 %	63.75 %	53.14 %	72.97 %

(a) Balanced Accuracy per label of Fully Supervised Approach on ExtraSensory Dataset

Labels (Total # Samples)	Acc	Gyro	WAcc	Loc	Aud	PS	EF
Lying Down (250906)	**73.35 % (3.6%)**	**68.43 % (3.8%)**	**69.62 % (4.1%)**	60.26 % (2.8%)	70.68 % (2.8%)	82.54 % (2.6%)	86.47 % (2.5%)
Sitting (251590)	**65.38 % (4.8%)**	59.93 % (5.2%)	**63.06 % (4.0%)**	60.91 % (3.7%)	61.82 % (3.3%)	**71.79 % (3.4%)**	72.70 % (2.9%)
Walking (251590)	74.08 % (3.2%)	78.72 % (4.0%)	68.76 % (3.6%)	56.60 % (4.3%)	53.37 % (3.3%)	64.71 % (3.1%)	72.12 % (2.7%)
Running (119463)	41.01 % (2.5%)	62.40 % (2.2%)	50.37 % (2.3%)	61.59 % (2.6%)	48.82 % (2.4%)	**70.14 % (2.5%)**	**72.72 % (2.4%)**
Bicycling (118795)	71.75 % (2.5%)	72.53 % (4.0%)	55.44 % (3.5%)	75.26 % (3.4%)	68.46 % (2.7%)	61.85 % (2.7%)	64.07 % (2.4%)
Sleeping (232327)	73.76 % (3.3%)	**70.81 % (3.5%)**	**66.50 % (2.6%)**	58.25 % (3.0%)	71.72 % (2.7%)	**86.16 % (2.7%)**	85.69 % (2.5%)
Lab work (37940)	60.23 % (2.2%)	55.52 % (4.1%)	62.81 % (3.4%)	78.59 % (2.8%)	52.64 % (2.6%)	77.09 % (2.6%)	77.11 % (2.5%)
In class (76957)	56.58 % (2.8%)	**52.53 % (2.4%)**	**59.96 % (2.7%)**	62.27 % (4.0%)	72.93 % (2.7%)	71.94 % (2.7%)	73.22 % (2.4%)
In a meeting (196373)	**65.93 % (2.9%)**	56.05 % (3.5%)	53.98 % (2.9%)	**75.44 % (2.6%)**	78.68 % (2.6%)	57.32 % (2.4%)	57.80 % (2.3%)
At main workplace (168805)	60.50 % (3.8%)	48.35 % (3.5%)	**52.21 % (3.7%)**	**76.43 % (2.6%)**	58.95 % (3.1%)	77.95 % (2.7%)	78.64 % (2.5%)
Indoors (160830)	74.14 % (2.7%)	77.54 % (2.4%)	66.82 % (2.7%)	**67.26 % (3.4%)**	59.48 % (2.6%)	72.33 % (2.7%)	71.47 % (2.5%)
Outside (121673)	75.11 % (2.1%)	76.08 % (3.1%)	67.01 % (2.9%)	**64.43 % (3.3%)**	59.00 % (2.8%)	70.37 % (2.7%)	64.29 % (2.5%)
In a car (140315)	78.89 % (2.7%)	60.13 % (3.5%)	67.36 % (2.3%)	83.22 % (3.4%)	76.54 % (2.7%)	84.12 % (2.7%)	81.96 % (2.4%)
Drive (I'm the driver) (143121)	**86.40 % (2.7%)**	60.57 % (4.0%)	**77.06 % (2.9%)**	87.09 % (3.2%)	74.39 % (2.8%)	80.43 % (2.7%)	83.85 % (2.5%)
Drive (I'm a passenger) (108144)	69.08 % (2.6%)	64.83 % (4.0%)	53.23 % (2.8%)	66.82 % (3.2%)	59.55 % (2.6%)	**79.82 % (2.4%)**	61.71 % (2.5%)
At Home (290275)	**65.92 % (3.0%)**	**58.92 % (2.4%)**	**58.51 % (4.0%)**	65.88 % (2.8%)	68.37 % (3.0%)	70.59 % (3.1%)	73.96 % (2.7%)
At a Restaurant (131761)	59.03 % (2.4%)	62.82 % (3.8%)	53.36 % (2.7%)	**53.15 % (2.8%)**	66.36 % (2.6%)	64.95 % (2.7%)	60.50 % (2.4%)
Phone in pocket (121489)	**68.06 % (2.8%)**	60.46 % (3.0%)	65.58 % (2.9%)	60.19 % (3.8%)	60.49 % (2.8%)	68.78 % (2.8%)	72.99 % (2.5%)
Exercise (204176)	60.09 % (3.1%)	61.21 % (2.5%)	53.07 % (4.0%)	**66.99 % (3.7%)**	**61.02 % (3.0%)**	**71.37 % (2.8%)**	56.65 % (2.4%)
Cooking (174621)	**54.35 % (2.3%)**	62.90 % (3.7%)	54.43 % (3.2%)	45.36 % (4.1%)	45.81 % (2.7%)	51.80 % (2.4%)	52.93 % (2.4%)
Strolling (43888)	**64.42 % (2.9%)**	54.24 % (2.7%)	63.37 % (4.3%)	**63.54 % (4.1%)**	51.52 % (2.9%)	49.99 % (2.6%)	49.24 % (2.4%)
Bathing-Shower (159634)	51.65 % (2.9%)	52.85 % (4.2%)	66.62 % (3.5%)	50.03 % (2.5%)	50.04 % (2.5%)	49.21 % (2.3%)	49.47 % (2.3%)

(b) Balanced Accuracy per label of Stream-based Active Learning on ExtraSensory Dataset (% of training data in parenthesis). Values in **bold** are values comparable or higher than the fully supervised values in 2a.

and deciding whether to request a label or not. After that, the runtime will depend on the type of model and model parameters. Runtimes of implementations running on an Intel Xeon CPU @ 2.10GHz for every dataset are given in Table 3. For PAMAP2, Fluid Intake, and ExtraSensory datasets, the average time of one iteration was fast since retraining a random forest and a logistic regression model is not computationally expensive. On the

Table 3. Average Time in seconds for one iteration of stream-based AL for every dataset.

Dataset	Model	Runtime (seconds)
PAMAP2	Random Forest (100 trees)	0.22
Fluid Intake	Random Forest (185 trees)	0.58
ExtraSensory	Logistic Regression	0.11
Opportunity	k-Nearest Neighbor (k=3)	53.04

other hand, for the Opportunity dataset, a k-NN model with k=3 was used which requires computing distances between every data point. Thus, retraining the model at every iteration increased the runtime complexity. We should note that in our simulations, we did not optimize our code for fast real-time operation. Moreover, the code used was from the replication package and was not optimized for computational efficiency.

7 CONDITIONAL MUTUAL INFORMATION

Our analysis has provided empirical evidence that AL can help reduce the amount of labeled data needed to train HAR models. However, an important problem that arises in practice is determining when to stop the AL process when starting with a fully unlabeled dataset. With pool-based AL, we observed varying rates of performance increase as data samples were queried. In some cases, models leveraging AL and built with less than 10% of the total amount of data showed comparable or better performance than a supervised model trained with 100% of the data. One way to further understand and measure this behavior is to compute the amount of information gained from querying additional samples with Conditional Mutual Information (CMI) at every iteration of the pool-based AL framework [18]. More specifically, we measured the amount of information gained about the predicted labels on the remaining unlabeled data when a certain batch is queried. Let $X = \{x_i\}_{i=1}^{n+m}$ denote the set of input features, $Y = \{y_i\}_{i=1}^{n+m}$ be the corresponding class labels where $y_i \in \{1, ..., c\}$ with c denoting the number of classes, $L = \{(x_1, y_1), ..., (x_n, y_n)\}$ denotes the set of labeled instances with $|L| = n$, and U denotes the index set corresponding to the unlabeled data $\{x_1, ..., x_m\}$ with $|U| = m$.

We define CMI as follows:

$$CMI = H(Y_U|X_U, L) - H(Y_U|X_U, L, (x_{i,...,k}, y_{i,...,k})) \tag{4}$$

where

$$\begin{aligned} H(Y_U|X_U, L) &= \sum_{i \in U} H(Y_i|x_i, L) \\ H(Y_i|x_i, L) &= -\sum_{y_i} P(y_i|x_i, L) \log P(y_i|x_i, L) \end{aligned} \tag{5}$$

$H(Y_i|x_i, L)$ represents the conditional entropy of the unknown label Y_i with respect to the instance x_i given the labeled data L. $H(Y_U|X_U, L)$ is the sum of individual marginal entropies. Using a parametric probabilistic conditional model, we get $P(y|x, L)$ for the classification task when testing on the unlabeled data. This is used to measure the CMI gained at every AL iteration. Since the performance curves for several datasets showed that prediction performance stabilizes as more training data was added, we hypothesized this behavior is attributed to the decrease of information gain. In other words, if we were to observe the CMI at every iteration, we should observe a convergence of CMI to small values (close to zero) which would indicate little to no information added with more data.

We measured CMI using the pool-based AL strategy for all the datasets in our analysis. For each dataset in Figure 11, there are multiple plots corresponding to their respective LOPO evaluations. As batches of data are

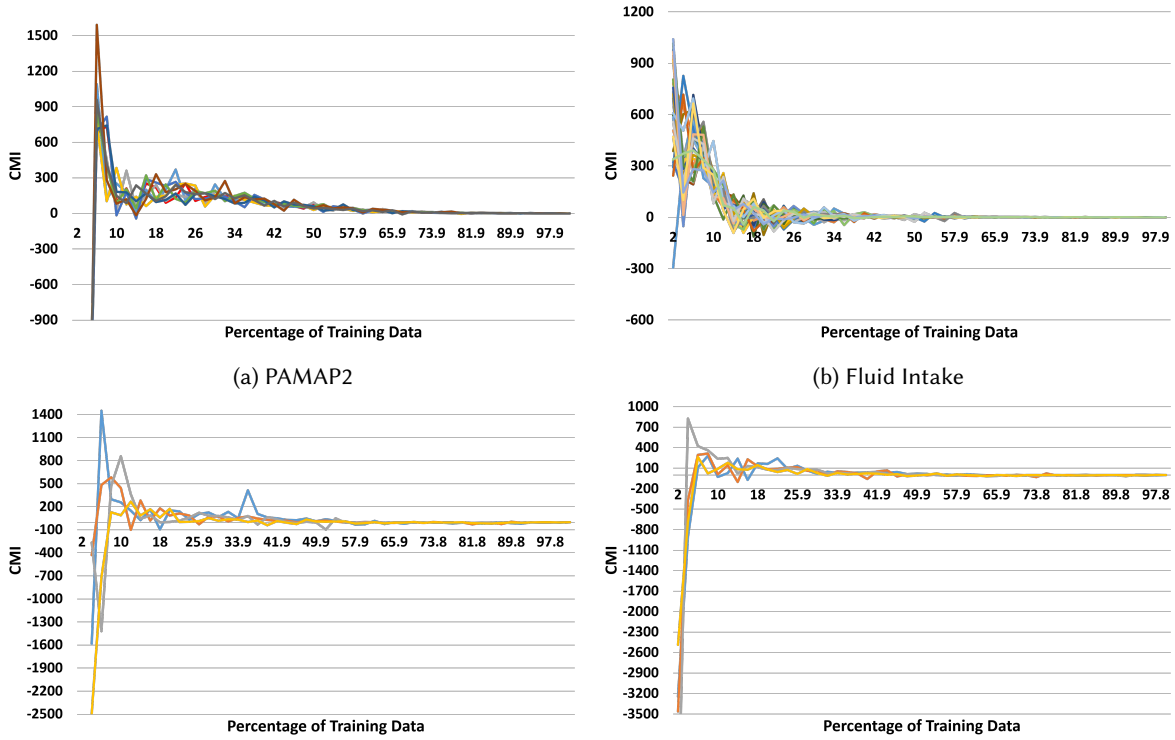


Fig. 11. Conditional mutual information (CMI) results for (a) PAMAP2, (b) Fluid Intake, and (c)-(d) Opportunity datasets. For all, CMI stabilizes and converges to near-zero values indicating little to no information gain as is queried using pool-based AL.

queried and added for training, data entropy, defined using Equation 5, can increase or decrease based on whether the queried batch reduces entropy or not. It is possible to observe that the CMI measure approaches zero as more training data is added. This indicates that the queried batch does not add any new information to the model. In other words, the model has already learned the best decision surface it can learn. This could be noted with the Fluid Intake, PAMAP2, and Opportunity datasets as shown in Figure 11. We attribute this behavior to the fact that the data was captured in more controlled, and thus less diverse settings. It is important to note, however, that CMI convergence is not always observed; it highly depends on the data and how diverse the data is. This CMI convergence was not observed for all activities in the ExtraSensory dataset, for example. Additionally, the CMI measure cannot be applied to the stream-based approach as it requires computing the entropy of a pool of unlabeled data.

7.1 Stopping Criteria

With any AL approach, the fundamental algorithmic principle involves repeatedly querying informative samples for annotation until a predefined stopping criterion is met. Since the goal of AL is to reduce the labeling efforts required to reach a good performance, a stopping criterion can be defined based on the model reaching its maximum effectiveness [53]. Typically, an AL process can end when the labeled training data reaches a predefined size. However, this does not always guarantee the best model with the best performance. Another possible

stopping criterion is when the model achieves a targeted performance. But, this is not always achievable as it depends on the problem setting. Examining the performance curve when testing the model on the test set, a stopping criterion could be determined when performance stops improving. Since our pool-based approach relies on the model's uncertainty in the unlabeled data, an intuitive stopping criterion would be if no more informative data samples are left to query. One way to measure this is by looking at the information gain in the predicted labels of the remaining unlabeled data as samples are repeatedly queried by the AL algorithm, formally defined as the conditional mutual information (Section 7). Thus, a reasonable stopping criterion can be the point at which CMI falls and stabilizes between a predefined range. It is based on the assumption and observation of the convergence and stabilization of the CMI curves after a certain number of queried data, as shown in Figure 11. However, it is possible that CMI does not converge for some datasets, which would indicate a high dataset entropy. Also, the CMI-based stopping criterion does not apply to the stream-based approach as it requires computing the probability estimates of the model on the pool of unlabeled data.

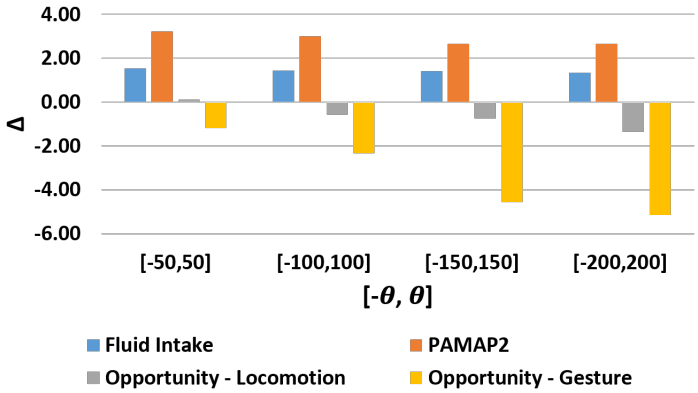
We conducted experiments with the Fluid Intake, PAMAP2, and Opportunity datasets and devised the following heuristic to determine when the AL algorithm should stop running:

(1) Define a fixed threshold θ and a stabilization wait time T (number of iterations) after CMI falls below the threshold.
(2) Stop when $CMI \in [-\theta, \theta]$ and remains in the range for at least T iterations.
(3) Extract best performance reached for some % of training data.

The parameters θ and T are determined by the user. The value of constant θ represents a trade-off between the number of annotations and the effectiveness of the resulting model. A larger θ would result in querying more unlabeled samples for annotation which could improve performance. However, a smaller θ results in fewer queries which could result in lower model performance. In order to observe this tradeoff, we varied $\theta \in \{50, 100, 150, 200\}$ and computed $\Delta = f_{AL} - f_{FS}$ where f_{FS} denotes the performance of the fully supervised approach and f_{AL} denotes the maximum performance reached when using the pool-based AL and stopping the process using the CMI-based stopping criterion. We set $T = 5$ iterations as the stabilization wait time. Implementing the LOPO evaluation, we averaged the performance results and computed Δ.

Figure 12a shows the effect of varying θ on Δ. A negative Δ value indicates that the AL approach resulted in a lower performance than the fully supervised method, while a positive value indicates that the AL approach outperformed the fully supervised method. Examining Figure 12a and Table 12b, we observed the tradeoff of increasing or decreasing θ. With a smaller threshold, it takes longer for the stopping point to be reached. As for performance, we observe a decrease in Δ for the Opportunity dataset which indicates higher performance compared to larger values of θ. However, for PAMAP2 and Fluid Intake datasets, there was little to no change in performance. This is due to the rapid early increase in performance (around 8-12% of training data) when applying the AL process. Thus, for this case, we were still able to achieve best performance using a small threshold, but we could have stopped the AL sampling earlier which would have reduced the cost of annotation.

As already mentioned, the proposed CMI-based stopping criterion does not apply to the stream-based approach since it requires a pool of unlabeled data for computing the entropy. Since for the pool-based scenario we deal with a fixed pool of unlabeled samples, the decision surface which the model will try to learn is represented by a fixed feature space. Thus, for this case, it is possible to track the change in the model's confidence in its predictions of the unlabeled samples since the model is repeatedly querying data from a fixed pool. However, for the stream-based scenario, once a sample is not queried by the AL framework, it is discarded. One possible way for deciding when to stop querying is when the model stops changing as queried samples are added. Model change can be measured by the change in the model's parameters after adding a sample to the training data. Typically, model parameters are found by minimizing a loss function using gradient descent (e.g., stochastic gradient descent). Thus, by using the gradient information to approximate the model change, we can possibly



(a)

	$\theta = 50$		$\theta = 100$		$\theta = 150$		$\theta = 200$	
	Train size (%)	**Stop size (%)**	**Train size (%)**	**Stop size (%)**	**Train Size (%)**	**Stop size (%)**	**Train Size (%)**	**Stop size (%)**
Fluid Intake	17.1	34.3	13.9	26.7	13.7	24.5	13.3	23.6
PAMAP2	22.4	64.7	21.3	52.2	16.4	43.3	16.4	34.4
Opportunity-Loco	39	51	31	35	29	29.5	22	24
Opportunity-Gest	43.5	46.5	37.5	37.5	29.5	29.5	28	28

(b)

Fig. 12. Results of CMI-based stopping criterion with varying thresholds: (a) bar plot of Δ and (b) table showing percentage of training data when process is stopped (stop size) and percentage of training data corresponding to the maximum performance reached (train size). $\Delta = f_{AL} - f_{FS}$ denotes the difference between the pool-based AL performance and the fully supervised performance. θ is a threshold used to determine when to stop when CMI falls in the range $[-\theta, \theta]$.

propose a stopping criterion that stops the AL process once the model stops changing as data samples are added. The key assumption underlying this approach is that the model is updated regularly in the initial iterations of the AL algorithm. In later iterations, the rate of updates decreases and eventually the model converges, i.e., it is no longer updated.

8 CONCLUSION

In this paper, we explored Active Learning (AL) as a way to achieve a highly accurate model while minimizing the annotation effort and amount of labeled data needed. We studied pool-based and stream-based AL approaches on four public datasets that are commonly used for human activity recognition (HAR) research, i.e., Opportunity [10], PAMAP2 [31], Fluid Intake [12], and ExtraSensory [47]. In both approaches, we found that it was possible to achieve a level of performance with models trained with AL that was comparable or even better than with models trained with supervised learning approaches. More significantly, model training with AL required much less annotated data, a very significant advantage since obtaining labeled data in HAR is often a difficult, costly and time-consuming endeavor. As an example, models trained on both PAMAP2 and Fluid Intake datasets showed a rapid increase in predictive performance as data samples were queried with AL, reaching and even surpassing fully supervised performance but using only 8% and 12% of the total amount of data, respectively. In this analysis, we also investigated the impact of hyperparameters (e.g., initial and sample batch sizes) on performance, and the effect of varying γ on the querying probability for the stream-based AL approach, which in turn affects the percentage of queried data.

Finally, in this work we also address a key practical problem of applying AL to unlabeled datasets: determining when to stop querying for labels. We propose a stopping criterion based on a Conditional Mutual Information (CMI) measure. Our results showed that for some datasets, CMI converges to small values (near zero), which indicates that after a certain number of AL iterations, there is little to no information gain with more queried data. Based on experimental results, we put forward a heuristic based on CMI that can serve as a guide to determine when additional sample querying is unlikely to result in substantial gains in performance. While we do not claim this heuristic is generalizable to all datasets, we believe it represents a valuable practical step towards applying AL and minimizing the effort of data annotations in HAR applications.

REFERENCES

[1] Zahraa Said Abdallah, Mohamed Medhat Gaber, Bala Srinivasan, and Shonali Krishnaswamy. 2015. Adaptive Mobile Activity Recognition System with Evolving Data Streams. *Neurocomput.* 150, PA (Feb. 2015), 304–317. https://doi.org/10.1016/j.neucom.2014.09.074

[2] Hande Alemdar, Tim L. M. van Kasteren, and Cem Ersoy. 2011. Using Active Learning to Allow Activity Recognition on a Large Scale. In *Proceedings of the Second International Conference on Ambient Intelligence (AmI'11)*. Springer-Verlag, Berlin, Heidelberg, 105–114. https://doi.org/10.1007/978-3-642-25167-2_12

[3] Dana Angluin. 1988. Queries and Concept Learning. *Mach. Learn.* 2, 4 (April 1988), 319–342. https://doi.org/10.1023/A:1022821128753

[4] Les Atlas, David Cohn, Richard Ladner, M. A. El-Sharkawi, and R. J. Marks, II. 1990. Advances in Neural Information Processing Systems 2. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, Chapter Training Connectionist Networks with Queries and Selective Sampling, 566–573. http://dl.acm.org/citation.cfm?id=109230.109294

[5] M. Bachlin, D. Roggen, G. Troster, M. Plotnik, N. Inbar, I. Meidan, T. Herman, M. Brozgol, E. Shaviv, N. Giladi, and J. M. Hausdorff. 2009. Potentials of Enhanced Context Awareness in Wearable Assistants for Parkinson's Disease Patients with the Freezing of Gait Syndrome. In *2009 International Symposium on Wearable Computers*. 123–130. https://doi.org/10.1109/ISWC.2009.14

[6] Salikh Bagaveyev and Diane J. Cook. 2014. Designing and Evaluating Active Learning Methods for Activity Recognition. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication (UbiComp '14 Adjunct)*. ACM, New York, NY, USA, 469–478. https://doi.org/10.1145/2638728.2641674

[7] Abdelkareem Bedri, Richard Li, Malcolm Haynes, Raj Prateek Kosaraju, Ishaan Grover, Temiloluwa Prioleau, Min Yan Beh, Mayank Goel, Thad Starner, and Gregory Abowd. 2017. EarBit: using wearable sensors to detect eating episodes in unconstrained environments. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies* 1, 3 (2017), 37.

[8] Klaus Brinker. 2003. Incorporating Diversity in Active Learning with Support Vector Machines. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning (ICML'03)*. AAAI Press, 59–66. http://dl.acm.org/citation.cfm?id=3041838.3041846

[9] Andreas Bulling, Ulf Blanke, and Bernt Schiele. 2014. A Tutorial on Human Activity Recognition Using Body-worn Inertial Sensors. *ACM Comput. Surv.* 46, 3, Article 33 (Jan. 2014), 33 pages. https://doi.org/10.1145/2499621

[10] Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digumarti, Gerhard Tröster, José Del R. Millán, and Daniel Roggen. 2013. The Opportunity Challenge: A Benchmark Database for On-body Sensor-based Activity Recognition. *Pattern Recogn. Lett.* 34, 15 (Nov. 2013), 2033–2042. https://doi.org/10.1016/j.patrec.2012.12.014

[11] Keum San Chun, Sarnab Bhattacharya, and Edison Thomaz. 2018. Detecting eating episodes by tracking jawbone movements with a non-contact wearable sensor. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 1 (2018), 4.

[12] Keum San Chun, Ashley B. Sanders, Rebecca Adaimi, Necole Streeper, David E. Conroy, and Edison Thomaz. 2019. Towards a Generalizable Method for Detecting Fluid Intake with Wrist-mounted Sensors and Adaptive Segmentation. In *Proceedings of the 24th International Conference on Intelligent User Interfaces (IUI '19)*. ACM, New York, NY, USA, 80–85. https://doi.org/10.1145/3301275.3302315

[13] Federico Cruciani, Ian Cleland, Chris Nugent, Paul McCullagh, Kåre Synnes, and Josef Hallberg. 2018. Automatic Annotation for Human Activity Recognition in Free Living Using a Smartphone. *Sensors* 18, 7 (2018). https://doi.org/10.3390/s18072203

[14] Toon De Pessemier and Luc Martens. 2018. Heart rate monitoring, activity recognition, and recommendation for e-coaching. *Multimedia Tools and Applications* (26 Jan 2018). https://doi.org/10.1007/s11042-018-5640-2

[15] A. Diete, T. Sztyler, and H. Stuckenschmidt. 2017. A smart data annotation tool for multi-sensor activity recognition. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. 111–116. https://doi.org/10.1109/PERCOMW.2017.7917542

[16] Y. Gu, Z. Jin, and S. C. Chiu. 2015. Active learning combining uncertainty and diversity for multi-class image classification. *IET Computer Vision* 9, 3 (2015), 400–407. https://doi.org/10.1049/iet-cvi.2014.0140

[17] Tianxu He, Zhang Kui, Jie Xin, Pengpeng Zhao, Jian Wu, Xuefeng Xian, Chunhua Li, and Zhiming Cui. 2014. An Active Learning Approach with Uncertainty, Representativeness, and Diversity. *TheScientificWorldJournal* 2014 (08 2014), 827586. https://doi.org/10.1155/2014/827586

[18] A. Holub, P. Perona, and M. C. Burl. 2008. Entropy-based active learning for object recognition. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. 1–8. https://doi.org/10.1109/CVPRW.2008.4563068

[19] S. C. Hsu, C. H. Chuang, C. L. Huang, P. R. Teng, and M. J. Lin. 2018. A video-based abnormal human behavior detection for psychiatric patient monitoring. In *2018 International Workshop on Advanced Image Technology (IWAIT)*. 1–4. https://doi.org/10.1109/IWAIT.2018.8369749

[20] Jie Jiang, Riccardo Pozza, Kristrún Gunnarsdóttir, G. Nigel Gilbert, and Klaus Moessner. 2017. Using Sensors to Study Home Activities. *J. Sensor and Actuator Networks* 6 (2017), 32.

[21] David D. Lewis and William A. Gale. 1994. A Sequential Algorithm for Training Text Classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '94)*. Springer-Verlag New York, Inc., New York, NY, USA, 3–12. http://dl.acm.org/citation.cfm?id=188490.188495

[22] R. Liu, T. Chen, and L. Huang. 2010. Research on human activity recognition based on active learning. In *2010 International Conference on Machine Learning and Cybernetics*, Vol. 1. 285–290. https://doi.org/10.1109/ICMLC.2010.5581050

[23] B. Longstaff, S. Reddy, and D. Estrin. 2010. Improving activity classification for health applications on mobile devices using active and semi-supervised learning. In *2010 4th International Conference on Pervasive Computing Technologies for Healthcare*. 1–7. https://doi.org/10.4108/ICST.PERVASIVEHEALTH2010.8851

[24] Donald McMillan, Barry Brown, Airi Lampinen, Moira McGregor, Eve Hoggan, and Stefania Pizza. 2017. Situating Wearables: Smartwatch Use in Context. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3582–3594. https://doi.org/10.1145/3025453.3025993

[25] Christopher Merck, Christina Maher, Mark Mirtchouk, Min Zheng, Yuxiao Huang, and Samantha Kleinberg. 2016. Multimodality Sensing for Eating Recognition. In *Proceedings of the 10th EAI International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth '16)*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 130–137. http://dl.acm.org/citation.cfm?id=3021319.3021339

[26] Daniela Micucci, Marco Mobilio, and Paolo Napoletano. 2016. UniMiB SHAR: a new dataset for human activity recognition using acceleration data from smartphones. *CoRR* abs/1611.07688 (2016). arXiv:1611.07688 http://arxiv.org/abs/1611.07688

[27] T. Miu, P. Missier, and T. Plötz. 2015. Bootstrapping Personalised Human Activity Recognition Models Using Online Active Learning. In *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*. 1138–1147. https://doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.170

[28] Georg Ogris, Paul Lukowicz, Thomas Stiefmeier, and Gerhard Tröster. 2012. Continuous activity recognition in a maintenance scenario: combining motion sensors and ultrasonic hands tracking. *Pattern Analysis and Applications* 15, 1 (01 Feb 2012), 87–111. https://doi.org/10.1007/s10044-011-0216-z

[29] J. Qi, P. Yang, M. Hanneghan, S. Tang, and B. Zhou. 2018. A Hybrid Hierarchical Framework for Gym Physical Activity Recognition and Measurement Using Wearable Sensors. *IEEE Internet of Things Journal* (2018), 1–1. https://doi.org/10.1109/JIOT.2018.2846359

[30] Julien Rebetez, Héctor F. Satizábal, and Andres Perez-Uribe. 2013. Reducing User Intervention in Incremental Activityrecognition for Assistive Technologies. In *Proceedings of the 2013 International Symposium on Wearable Computers (ISWC '13)*. ACM, New York, NY, USA, 29–32. https://doi.org/10.1145/2493988.2494350

[31] Attila Reiss and Didier Stricker. 2012. Creating and Benchmarking a New Dataset for Physical Activity Monitoring. In *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments (PETRA '12)*. ACM, New York, NY, USA, Article 40, 8 pages. https://doi.org/10.1145/2413097.2413148

[32] J. Ryder, B. Longstaff, S. Reddy, and D. Estrin. 2009. Ambulation: A Tool for Monitoring Mobility Patterns over Time Using Mobile Phones. In *2009 International Conference on Computational Science and Engineering*, Vol. 4. 927–931. https://doi.org/10.1109/CSE.2009.312

[33] A. Sathyanarayana, F. Ofli, L. Fernandez-Luque, J. Srivastava, A. Elmagarmid, T. Arora, and S. Taheri. 2016. Robust Automated Human Activity Recognition and Its Application to Sleep Research. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. 495–502. https://doi.org/10.1109/ICDMW.2016.0077

[34] Tobias Scheffer, Christian Decomain, and Stefan Wrobel. 2001. Active Hidden Markov Models for Information Extraction. In *Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis (IDA '01)*. Springer-Verlag, London, UK, UK, 309–318. http://dl.acm.org/citation.cfm?id=647967.741626

[35] D. Sculley. 2007. Online Active Learning Methods for Fast Label-Efficient Spam Filtering.

[36] R. Serra, D. Knittel, P. Di Croce, and R. Peres. 2016. Activity Recognition With Smart Polymer Floor Sensor: Application to Human Footstep Recognition. *IEEE Sensors Journal* 16, 14 (July 2016), 5757–5775. https://doi.org/10.1109/JSEN.2016.2554360

[37] Farhad Shahmohammadi, Anahita Hosseini, Christine E. King, and Majid Sarrafzadeh. 2017. Smartwatch Based Activity Recognition Using Active Learning. In *Proceedings of the Second IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE '17)*. IEEE Press, Piscataway, NJ, USA, 321–329. https://doi.org/10.1109/CHASE.2017.115

[38] M. Stikic, T. Huynh, K. Van Laerhoven, and B. Schiele. 2008. ADL recognition based on the combination of RFID and accelerometer sensing. In *2008 Second International Conference on Pervasive Computing Technologies for Healthcare*. 258–263. https://doi.org/10.1109/

PCTHEALTH.2008.4571084

[39] Maja Stikic, Diane Larlus, Sandra Ebert, and Bernt Schiele. 2011. Weakly Supervised Recognition of Daily Life Activities with Wearable Sensors. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 12 (Dec. 2011), 2521–2537. https://doi.org/10.1109/TPAMI.2011.36

[40] Maja Stikic and Bernt Schiele. 2009. Activity Recognition from Sparsely Labeled Data Using Multi-Instance Learning. In *Location and Context Awareness*, Tanzeem Choudhury, Aaron Quigley, Thomas Strang, and Koji Suginuma (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 156–173.

[41] M. Stikic, K. Van Laerhoven, and B. Schiele. 2008. Exploring semi-supervised and active learning for activity recognition. In *2008 12th IEEE International Symposium on Wearable Computers*. 81–88. https://doi.org/10.1109/ISWC.2008.4911590

[42] A. Subasi, M. Radhwan, R. Kurdi, and K. Khateeb. 2018. IoT based mobile healthcare system for human activity recognition. In *2018 15th Learning and Technology Conference (L T)*. 29–34. https://doi.org/10.1109/LT.2018.8368507

[43] Emmanuel Munguia Tapia, Stephen S. Intille, and Kent Larson. 2004. Activity Recognition in the Home Using Simple and Ubiquitous Sensors. In *Pervasive Computing*, Alois Ferscha and Friedemann Mattern (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 158–175.

[44] Edison Thomaz, Irfan Essa, and Gregory D. Abowd. 2015. A Practical Approach for Recognizing Eating Moments with Wrist-mounted Inertial Sensing. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15)*. ACM, New York, NY, USA, 1029–1040. https://doi.org/10.1145/2750858.2807545

[45] Emma L. Tonkin, Alison Burrows, Przemyslaw R. Woznowski, Pawel Laskowski, Kristina Y. Yordanova, Niall Twomey, and Ian J. Craddock. 2018. Talk, Text, Tag? Understanding Self-Annotation of Smart Home Data from a User's Perspective. *Sensors* 18, 7 (2018). https://doi.org/10.3390/s18072365

[46] Y. Vaizman, K. Ellis, and G. Lanckriet. 2017. Recognizing Detailed Human Context in the Wild from Smartphones and Smartwatches. *IEEE Pervasive Computing* 16, 4 (October 2017), 62–74. https://doi.org/10.1109/MPRV.2017.3971131

[47] Yonatan Vaizman, Katherine Ellis, Gert Lanckriet, and Nadir Weibel. 2018. ExtraSensory App: Data Collection In-the-Wild with Rich User Interface to Self-Report Behavior. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 554, 12 pages. https://doi.org/10.1145/3173574.3174128

[48] Niels van Berkel, Denzil Ferreira, and Vassilis Kostakos. 2017. The Experience Sampling Method on Mobile Devices. *ACM Comput. Surv.* 50, 6, Article 93 (Dec. 2017), 40 pages. https://doi.org/10.1145/3123988

[49] I. Žliobaitė, A. Bifet, B. Pfahringer, and G. Holmes. 2014. Active Learning With Drifting Streaming Data. *IEEE Transactions on Neural Networks and Learning Systems* 25, 1 (Jan 2014), 27–39. https://doi.org/10.1109/TNNLS.2012.2236570

[50] Y. Wang, S. Cang, and H. Yu. 2018. A Data Fusion based Hybrid Sensory System for Older People's Daily Activity and Daily Routine Recognition. *IEEE Sensors Journal* (2018), 1–1. https://doi.org/10.1109/JSEN.2018.2833745

[51] Zuobing Xu, Ram Akella, and Yi Zhang. 2007. Incorporating Diversity and Density in Active Learning for Relevance Feedback. In *Proceedings of the 29th European Conference on IR Research (ECIR'07)*. Springer-Verlag, Berlin, Heidelberg, 246–257. http://dl.acm.org/citation.cfm?id=1763653.1763684

[52] Aras Yurtman and Billur Barshan. 2016. Human Activity Recognition Using Tag-Based Radio Frequency Localization. *Applied Artificial Intelligence* 30, 2 (2016), 153–179. https://doi.org/10.1080/08839514.2016.1138787 arXiv:https://doi.org/10.1080/08839514.2016.1138787

[53] Jingbo Zhu, Huizhen Wang, Eduard Hovy, and Matthew Ma. 2010. Confidence-based Stopping Criteria for Active Learning for Data Annotation. *ACM Trans. Speech Lang. Process.* 6, 3, Article 3 (April 2010), 24 pages. https://doi.org/10.1145/1753783.1753784