Memory-Aware Active Learning in Mobile Sensing Systems

Zhila Esna Ashari, *Student Member, IEEE*, Naomi S. Chaytor, Diane J. Cook, *Fellow, IEEE*, and Hassan Ghasemzadeh, *Senior Member, IEEE*

Abstract—We propose a novel active learning framework for activity recognition using wearable sensors. Our work is unique in that it takes limitations of the oracle into account when selecting sensor data for annotation by the oracle. Our approach is inspired by human-beings' limited capacity to respond to prompts on their mobile device. This capacity constraint is manifested not only in the number of queries that a person can respond to in a given time-frame but also in the time lag between the query issuance and the oracle response. We introduce the notion of *mindful active learning* and propose a computational framework, called *EMMA*, to maximize the active learning performance taking informativeness of sensor data, query budget, and human memory into account. We formulate this optimization problem, propose an approach to model memory retention, discuss the complexity of the problem, and propose a greedy heuristic to solve the optimization problem. Additionally, we design an approach to perform mindful active learning in batch where multiple sensor observations are selected simultaneously for querying the oracle. We demonstrate the effectiveness of our approach using three publicly available activity datasets and by simulating oracles with various memory strengths. We show that the activity recognition accuracy ranges from 21% to 97% depending on memory strength, query budget, and difficulty of the machine learning task. Our results also indicate that EMMA achieves an accuracy level that is, on average, 13.5% higher than the case when only informativeness of the sensor data is considered for active learning. Moreover, we show that the performance of our approach is at most 20% less than the experimental upper-bound and up to 80% higher than the experimental lower-bound. To evaluate the performance of EMMA for batch active learning, we design two instantiations of EMMA to perform active learning in batch mode. We show that these algorithms improve the algorithm training time at the cost of a reduced accuracy in performance. Another finding in our work is that integrating clustering into the process of selecting sensor observations for batch active learning improves the activity learning performance by 11.1% on average, mainly due to reducing the redundancy among the selected sensor observations. We observe that mindful active learning is most beneficial when the query budget is small and/or the oracle's memory is weak. This observation emphasizes advantages of utilizing mindful active learning strategies in mobile health settings that involve interaction with older adults and other populations with cognitive impairments.

Index Terms—Active learning, wearable computing, machine learning, activity recognition, memory retention, cognitive factors, human-in-the-loop learning.

1 Introduction

With the advent of the Internet-of-Things (IoT) paradigm, applications of sensor-based systems have advanced significantly across many domains from health monitoring and autonomous vehicles to smart building and environmental monitoring [1], [2]. Mobile and wearable devices are being increasingly utilized, along with machine learning algorithms, to monitor physical and mental health, and to improve human well-being through clinical interventions. Most of these applications are human-centered in that they focus on monitoring human health [3] and even interacting with humans to incorporate their feedback for improved performance of the system. The monitoring component often relies on computational algorithms that can detect important health events. For example, wearable sensors are extensively utilized to record human physiological data and then, computational algorithms such as machine learning models are applied for data analysis and to make predictions about events of interest [4], [5], [6].

To train accurate machine learning models for different applications, such as activity recognition, an adequate number of labeled sensor data is required. However, data collections and related experiments are mainly done in laboratory settings where the experiments are highly controlled. Unfortunately, models that are trained based on sensor data collected in controlled environments and laboratory settings perform extremely poorly when utilized in uncontrolled environments and outside clinics [7]. Therefore consideration of real-world and uncontrolled settings has become increasingly important. Specifically, in human-centered applications, various limitations of human-beings, which can affect the performance of the trained models, need to be taken into account.

For an activity recognition classifier to be accurate, one needs to collect and label sensor data in end-user settings. Therefore, active learning is a natural choice for labeling the data where the end-user acts as the oracle agent and we iteratively query the user for correct labels [8], [9]. Throughout this article, the terms 'end-user' and 'oracle' are interchangeably used. In such a human-centered monitor setting, it is critical to design active learning strategies that

• *Z. Esna Ashari, D. Cook, and H. Ghasemzadeh are with the School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA, 99164 USA email:{z.esnaashariesfahan,djcook, hassan.ghasemzadeh, naomic}@wsu.edu*
N. Chaytor is with the Department of Medical Education and Clinical Sciences, Washington State University, Spokane, WA, 99202 USA email: naomic@wsu.edu

Manuscript received XX, XX; revised XX.

are mindful of the user's cognitive and compliance capabilities. We recognize that human-beings are limited in their capacity to respond to, for example, prompts on their mobile devices. This capacity constraint is usually manifested in the number of queries that a person can, or will, respond to in a given time-frame and in the difference between the time that a query is made and when it has been responded to. This issue is critical in wearable-based continuous health monitoring where the amount of sampled data is orders of magnitude more than what the end-user can possibly annotate [10].

In this article, we introduce the notion of *mindful active learning* and propose EMMA[1] to maximize the active learning performance, taking informativeness of sensor data, query budget, and human memory into account [11]. To the best of our knowledge, our work is the first study that combines informativeness of queried data with the oracle's memory strength in a unified framework for active learning.

Our contributions in this paper can be summarized as follows: (i) we introduce mindful active learning as a general approach for budget-aware and delay-tolerant active learning in human-in-the-loop mobile systems. Mindful active learning takes into account the possibility that the oracle may forget the past events, particularly when there is a large time difference between the query and the activity being performed, and at the same time being constrained on the maximum number of queries that can be made; (ii) to account to human memory in the active learning process, we propose an approach to model memory retention based on the Ebbinghaus forgetting curve [12]; (iii) we formulate mindful active learning as an optimization problem and propose an approach to solve this problem; and (iv) we evaluate the performance of our algorithms for activity recognition using several datasets involving wearable and mobile sensors.

2 Related Work

Active learning has been widely used for human-in-loop learning tasks to iteratively interact with an expert user to retrieve necessary information for improved learning performance [13], [14], [15], [16], [17], [18], [19], [20], [21]. Vijayanarasimhan et al. studied the problem of active learning under constrained query budget for image and video recognition [22]. Active learning has also been studied in the context of activity recognition [23], [24]. For instance, Bao et al. used accelerometer data annotated by end-users to detect physical activities [25]. Active learning has been used for many other human-centered prediction problems beyond those that use wearable sensors and images [8], [26]. Murukannaiah et al. presented an active learning framework on personalizing the place-aware application [27] where the approach was based on collecting information from the user on places that the data were recorded with querying according to entropy only. Also, Active learning allows us to collect a small number of labeled observations to personalize machine learning algorithms when the system is deployed in a new setting. This area of personalizing machine learning models is also related to transfer learning research. In recent years there has been a growing body of literature in designing transfer learning techniques for active recognition systems [28], [29], [30], [31].

Physical activity monitoring has various clinical applications such as gait analysis [32], fall detection [33], mobility assessment, and activity recognition. Activity recognition, which is the pilot application in this study, aims at determining types of physical activities that a person or a group of people perform based on sensor and/or video observation data.

In mobile sensor-based systems, these observations include data recorded by various wearable sensors or sensors embedded in smart devices, such as accelerometer and gyroscope sensors. For the purpose of recognizing human activities based on sensor data, an adequate amount of labeled training data is needed in order to achieve a high accuracy. However, for most of the applications, the recorded sensor samples are currently labeled in lab settings. When collecting data in uncontrolled settings using mobile devices, gathering the ground truth labels relies on user's annotation of their activity behaviors. As a result, cognitive factors of the user are often overlooked when gathering the ground truth annotations. Therefore, when sensor-based systems are used in uncontrolled environments, this unconsidered factor can highly and negatively affect the performance of the recognition model. As a result, the performance of activity recognition methods are directly dependent on how the active learning algorithm is designed [34], [35].

Prior research does not take into account important cognitive attributes of human beings, such as memory [2] strength for remembering the events, while designing active learning solutions. Current research makes an implicit assumption that either the oracle has a perfect memory that can precisely remember all the events, or each query is instantaneously responded to. None of these assumptions are realistic in continuous health monitoring settings where the end-user of the system also acts as our oracle. In this article, we attempt to take the first steps at integrating cognitive and compliance/adherence attributes of the oracle with active learning. In particular, we account for (i) the oracle capability to respond to active learning queries as measured by the number of queries that are made; and (ii) the oracle's cognitive capacity to respond to the queries as measured by memory retention, a function that combines memory strength of the oracle agent with the amount of delay in responding to the issued queries.

3 Mindful Active Learning Framework

The proposed framework for mindful active learning is shown in Fig. 1. Users employ wearable devices to collect sensor data about activity behavior. The process of data collection (i.e., sensing) is continuous and transparent to the user. The gathered sensor data form an unlabeled dataset, which also forms the input to our active learning algorithm. The proposed active learning algorithm, referred to as EMMA, also process the user's memory strength and a maximum available query budget, B, as input. The EMMA algorithm selects an optimal set of sensor data based on

1. Software for EMMA (*Entropy-Memory Maximization*) is available at https://github.com/zhesna/EMMA.

2. In this article, 'memory' refers to human memory.

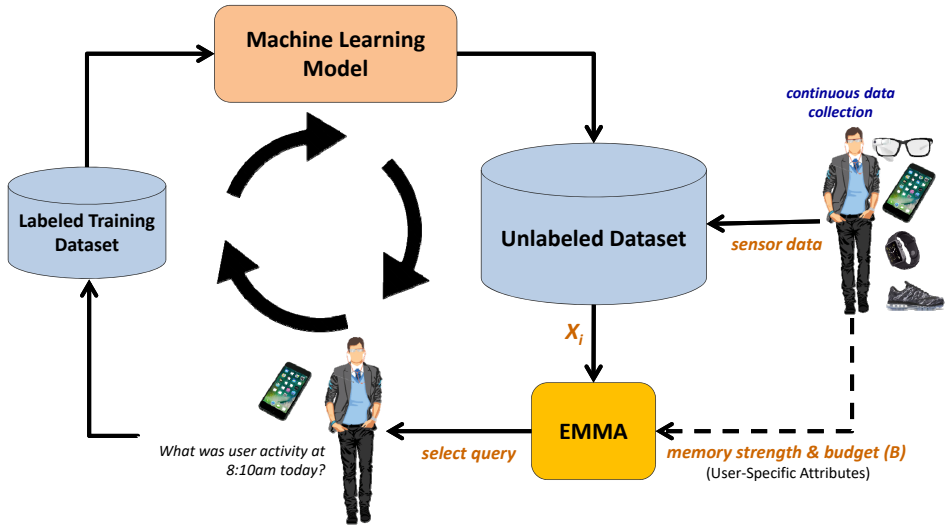

Fig. 1. Mindful Active Learning Framework. X_i represents the observations recorded by the sensors. The sensor observations along with the available budget B and memory strength of the user are the inputs to the algorithm in EMMA, the proposed active learning method. The optimal set of samples selected by EMMA, based on memory strength, time difference between occurrence of the activity and query time, and informativeness of the sensor data are selected. The user is queried the samples at each iteration and the labeled samples are used to update the machine learning model. This iterative approach continues until the available budget is exhausted.

memory strength, time difference between occurrence of the activity and query time, and informativeness of the sensor data. The user is then queried to label each selected data point. After each data point is labeled by the user, it is added to the training dataset and the activity recognition model is retrained for improved performance. The process repeats until the query budget is exhausted.

3.1 Problem Statement

Assume that we are given a collection of sensor measurements recorded while wearable sensors are carried by the user during daily activities. Without loss of generality, we assume that the sensor measurements, referred to as sensor observations henceforth, are represented in feature space. These observations need to be used to train a classifier for activity recognition. Because the sensor observations are unlabeled, we use active learning to construct a labeled training set by querying the end-user to annotate a subset of the observations. Because of the time lag between querying a sensor observation and performing the activity associated with that observation, it is possible that the oracle is incapable of remembering the correct label. We postulate that the likelihood of such a mislabeling is a function of the oracle's memory strength and the time lag. Memory strength determines how drastically the ability of human to recall the events falls over time. Moreover, we assume that the number of allowed queries is constrained to a given value, referred to as budget, in order to minimize the burden of the oracle. Therefore, we need to select a subset of sensor observations, upper-bounded by a given budget, such that the probability of obtaining correct and informative labels is maximized. This problem can be formally defined as follows.

Problem 1 (Mindful Active Learning). *Let $\mathcal{A} = \{A_1, A_2, ..., A_n\}$ represent the set of activities that need to be recognized by the wearable system. We refer to this set as the activity vocabulary. Furthermore, let $\mathcal{X} = \{X_1, X_2, ..., X_m\}$ be a set of m observations made by the sensors at times $\mathcal{T} = \{t_1, t_2, ..., t_m\}$. The task of active learning is to query the user, with a predefined memory strength of s, at time $t_q \geq t_m$ to label sensor observations in $\mathcal{X}$ and train an activity recognition model using the labeled observations. The active learning process is constrained by limiting the number of queries to a given upper-bound budget, B.*

3.2 Problem Formulation

The task of selecting a subset of observations to be labeled by the oracle can be viewed as finding a set $\mathcal{Z} = \{z_1, z_2, ..., z_k\}$ of k observations in $\mathcal{X}$ (i.e., $\mathcal{Z} \subseteq \mathcal{X}$ and $k \leq B$) such that the misclassification error due to a model trained over $\mathcal{Z}$ is minimized. The goal is to select the best subset of unlabeled observations to form a candidate query set (Z) that results in an accurate classifier. In order to maximize the performance of the final classifier, we consider two criteria including informativeness of the candidate observation and the oracle's ability to remember the correct label at time t_q. We use I_i and M_i to refer to informativeness and memory measures for a given observation X_i captured at time t_i.

This optimization problem can be formulated as

$$\text{Maximize} \quad \sum_{i=1}^{m} a_i \mathcal{E}(I_i, M_i) \tag{1}$$

Subject to:

$$X_i \in \mathcal{X} \tag{2}$$

$$\sum_i a_i \leq B \tag{3}$$

$$a_i \in \{0, 1\} \tag{4}$$

where X_i denotes the sensor observation captured at time t_i, and B represents the budget. The binary decision variable a_i determines whether or not observation X_i is selected for inclusion in $\mathcal{Z}$. The objective function in (1) attempts to maximize the total amount of expected gain ($\mathcal{E}$) given informativeness and memory for individual observations.

We propose to use uncertainty of the model with respect to a given observation as a measure of informativeness of that observation for inclusion in $\mathcal{Z}$. We also propose to measure memory by combining the memory strength of the user with the expected error due to difference in time between query issuance and activity occurrence.

To obtain the expected gain ($\mathcal{E}(I_i, M_i)$), which involves two random variables, one can aim to model the expected gain experimentally based on vast amounts of data (i.e., both sensor and human memory). However, in the absence of such data or prior research in this area, we assume that informativeness of an observation and the reliability of its prospective label are independent of each other. Therefore, we model the expected gain as a multiplicative function of these two variables. Intuitively, because I_i determines how informative a sample X_i is and M_i determines how likely it is to get the correct label for observation X_i, their multiplication is a reasonable proxy to the overall expected gain of the observation. Therefore, the expected gain is computed as

$$\mathcal{E}(I_i, M_i) = \mathcal{E}(I_i)\mathcal{E}(M_i) \tag{5}$$

We note that (5) assumes an even contribution of the informativeness and memory to the overall expected gain. Investigating a more complex formulation of the expected gain where the importance of each random variable is taken into consideration is out of the scope of this paper.

To quantify the informativeness (I_i) of observation X_i, we propose to use entropy, as shown in (6), to measure how certain the model is about its predicted label for X_i.

$$\mathcal{E}(I_i) = E_i = -\sum_{j=1}^{n} P_{ij} \log P_{ij} \tag{6}$$

The term E_i in (6) refers to entropy for observation X_i, and P_{ij} represents the probability of X_i being classified as activity A_j. Because the classifier is less certain to classify observations that carry a higher entropy, such observations will naturally be more informative if labeled and used for classifier retraining. Therefore, E_i is a reasonable proxy for $\mathcal{E}(I_i)$. To quantify memory, M_i, for observation X_i, we define memory retention as follows.

Definition 1 (Memory Retention)**.** *Memory retention, R, is defined as the probability of a human subject with a memory strength of s being able to remember an event correctly after a given time, t, has elapsed.*

We use the Ebbinghaus forgetting curve [12], [36], [37] to quantify memory retention. To this end, memory retention for observation X_i is given by

$$\mathcal{E}(M_i) = R_i = e^{-\Delta t_i / s} \tag{7}$$

where Δt_i denotes the difference in time between occurrence of the event represented by observation X_i and issuance of the query (i.e., t_q). Furthermore, the term s represents memory strength, which is specific to each user. The memory retention in (7) is naturally a measure for receiving correct labels from the oracle. Therefore, the memory retention R_i can be used to quantify the memory for observation X_i (i.e., $\mathcal{E}(M_i) = R_i$).

Problem 2 (Entropy-Memory Maximization)**.** *By replacing $\mathcal{E}(I_i)$ and $\mathcal{E}(M_i)$ in* (5) *with the entropy measure in* (6) *and memory retention in* (7)*, we can rewrite the objective function in* (1) *as follows.*

$$\textit{Maximize} \quad \sum_{i=1}^{m}\sum_{j=1}^{n} a_i e^{-\Delta t_i / s}(-P_{ij} \log P_{ij}) \tag{8}$$

Subject to:

$$X_i \in \mathcal{X} \tag{9}$$

$$\sum_{i} a_i \leq B \tag{10}$$

$$a_i \in \{0, 1\} \tag{11}$$

We refer to this formulation of mindful active learning as Entropy-Memory Maximization (EMMA). In the next section, we discuss the complexity of EMMA analysis and present a solution to this discrete constrained multi-variable maximization problem.

3.3 Problem Complexity

In this section, we discuss the complexity of the entropy-memory maximization discussed in Problem 2.

The complexity of the EMMA problem arises from its exponential input size. To provide some insight into the input size of the EMMA problem, consider a naive approach (i.e., brute-force solution), in which we cycle through all subsets of the unlabeled dataset with the number of elements less than or equal to B. For each subset, we can generate all possible orderings of the sensor observations within that subset. For each ordering, we draw one observation from the subset at a time, compute entropy and memory retention values, query the oracle, and train a classifier with the observations labeled so far. By repeating this process for all subsets and all ordering of the observations within each subset, we can finally choose the ordering and subset that has the maximum value for the objective function in (8). Although this brute-force approach is impractical for real-world deployment, an analysis of the complexity of such an approach provides insight into the relationship between the complexity function and various problem parameters.

Theorem 1. *The time complexity of the brute-force approach for solving EMMA (as described in Problem 2) is exponential in the query budget, B.*

Proof. It is straightforward to see that the overall time complexity of the brute-force solution is $O(|S|)$ where $|S|$ refers to the total number of orderings of all subsets of unlabeled set $\mathcal{X}$ with a subset size less than or equal to B. For each subset of size b, there exist $b!$ orderings of the observations that reside within the subset. Therefore, $|S|$ is given by

$$|S| = \sum_{b=1}^{B} \binom{m}{b} \times b! = \sum_{b=1}^{B} \frac{m!}{(m-b)!} \tag{12}$$

where m denotes the size of the unlabeled set $\mathcal{X}$. The equation in (12) can be presented as

$$|S| = \sum_{b=1}^{B}(((m-b)+1) \times ((m-b)+2) \times ... \times ((m-b)+(b-1)) \times ((m-b)+b)) \tag{13}$$

and therefore:

$$|S| = \sum_{b=1}^{B} \prod_{k=1}^{b} (m-b+k) \tag{14}$$

Because in real-world scenarios, the number of recorded sensor observations is orders of magnitude higher than the number of queries that the oracle can possibly respond to (i.e., $B \ll m$), we can write

$$\prod_{k=1}^{b} (m-b+k) \approx m^b \tag{15}$$

Therefore, (12) can be rewritten as

$$|S| \approx m + m^2 + ... + m^B \tag{16}$$

Using geometric series we can conclude that

$$|S| \approx \frac{m^{(B+1)} - 1}{m-1} \approx m^B \tag{17}$$

Therefore, the time complexity of the naive approach is $O(m^B)$. □

3.4 Greedy Approach for EMMA

Because the solution space is exponential, here we design a greedy heuristic algorithm, shown in Algorithm 1, to solve the EMMA problem. Recall that the goal of the optimization problem in EMMA is to select a subset of observations of high-entropy such that their labels are likely to be remembered by the oracle. So this greedy approach works in this way that, it first evaluates the expected gain for all the samples based on their informativeness, memory strength of the oracle and the amount of time that has passed from doing the corresponding activity by the user and recording that sample. Then, the algorithm finds and selects the sample with the highest value of expected gain and queries the oracle for its label. Then collects the label from the oracle and adds the labeled sample to the training set it is building which will be used for predictions. Based on the updated training set, an updated model will calculate the new values of entropy for the other unlabeled recorded samples. This process will repeat until we have budget available.

This greedy algorithm iteratively chooses the best candidate observation from the set of unlabeled observations $\mathcal{X}$ with the highest values of the expected gain in (5) assuming that the informativeness and memory are measured as given in (6) and (7), respectively. Note that after moving an observation from $\mathcal{X}$ to $\mathcal{Z}$, the model $\mathcal{M}$ is retrained using the labeled data in $\mathcal{Z}$. This procedure is repeated until the entire budget is consumed.

Algorithm 1 Greedy algorithm for EMMA.

Input: $\mathcal{X}$ (unlabeled observations), B (budget)
Output: $\mathcal{Z}$ (labeled observations)
for b = 1 to B **do**
  Compute $\mathcal{E}(I_i, M_i)$ for all $X_i \in \mathcal{X}$ using (5)–(7)
  Find $X_i \in \mathcal{X}$ with highest value of $\mathcal{E}(I_i, M_i)$
  Remove X_i from $\mathcal{X}$
  Query oracle to annotate X_i, and add labeled X_i to $\mathcal{Z}$
  Retrain model $\mathcal{M}$ using labeled items in $\mathcal{Z}$
end for

Lemma 1. *The time complexity of Algorithm 1 is linear in m, the number of original unlabeled observations in $\mathcal{X}$.*

Proof. The 'for' loop iterates B times. During each iteration, we need to (i) compute $\mathcal{E}(I_i, M_i)$ for the remaining elements in $\mathcal{X}$ and (ii) find $X_i \in \mathcal{X}$ with the highest value of $\mathcal{E}(I_i, M_i)$. Both of these operations require $O(|\mathcal{X}|)$ to complete assuming a constant time complexity for computing $\mathcal{E}(I_i, M_i)$ for a given X_i. Therefore, the time complexity of Algorithm 1 is $B \times O(|\mathcal{X}|)$. Because $|\mathcal{X}|$ is initially m and decreases by one at each iteration, the total number of times to compute $\mathcal{E}(I_i, M_i)$ is given by

$$\sum_{b=1}^{B} (m-b-1) = O(m.B) \tag{18}$$

Therefore, the time complexity of Algorithm 1 is $O(m.B)$ □

It is straightforward to see that Algorithm 1 terminates. Specifically, Algorithm 1 converges all the time because the algorithm is based on a repeating process and each step terminates all the time. For a closer look, the first step of the repeating process, which involves computing $\mathcal{E}(I_i, M_i)$ for all X_i, is a deterministic calculation of a finite set of elements (i.e., 'm') using a given formula. The second step, makes use of a one step algorithm which goes over the whole sample set to find the one with the highest expected gain, which obviously converges because the number of input items in the set is limited to 'm'. The third step is a deterministic removal step. The most challenging step is the forth step within the repeating loop, which relies on the oracle to respond to the active learning queries. In this work, we assume that the user is completely engaged in the algorithm training process and responds to the queries instantaneously. However, in real world, the oracle may not respond to a query and the algorithm may keep waiting. To overcome this situation, one solution may be to define a expiration time for each query. As a result, if a user does not respond to a query and a sufficient amount of time is passed, the query expires and the algorithm moves on. In this way, we assure convergence of the algorithm. The fifth step of the 'for' loop in Algorithm 1 is focused on training a classification model. The training process depends on the underlying classification algorithms. However, each classification algorithm is controlled by parameters to stop

the training [38] and therefore, the classification training terminates in a constant time given the problem parameters.

3.5 Greedy Approach for Batch-EMMA

In this section, we propose our active learning approach for operation in batch mode. We refer to this approach as batch-EMMA. In batch mode active learning, instead of selecting one sample in each iteration of the EMMA algorithm, multiple samples are selected [13], [15], [18], [22]. There are different approaches to select these samples and based on this, we propose two versions for batch-EMMA.

The first version is referred to as EMMA-M in this article. In EMMA-M, the algorithm chooses the best candidate observations from the set of unlabeled observations $\mathcal{X}$ with the highest values of the expected gain $\mathcal{E}(I_i, M_i)$ in (5) in each iteration. After querying the end-user for the labels of these selected observations, they will be moved from $\mathcal{X}$ to $\mathcal{Z}$, and the model $\mathcal{M}$ is retrained using the labeled data in $\mathcal{Z}$. This procedure is repeated until the entire budget is consumed.

The second version of our batch active learning is referred to as EMMA-C in this article. In EMMA-C, in each iteration of active learning, first the unlabeled sensor observations are clustered into multiple groups. Then the best observation with the highest value of expected gain $\mathcal{E}(I_i, M_i)$ is selected from each cluster and then the end-user is queried to label the selected observations. The main goal of EMMA-C is to reduce redundancy among the selected observations, as its advantage over EMMA-M.

In order to explain the differences between EMMA-M and EMMA-C algorithms, Fig. 2 is presented for a synthetic dataset with two features, f_1 and f_2, where the plots show the observations distributions in a 2D feature space. The selected samples by each algorithm are shown in "red".

As shown in Fig. 2, EMMA-M selects the high ranked samples based on their expected gain, which may result in selecting the ones that are close to each other in space. However, EMMA-C first clusters the points. Therefore, the samples that are more similar are grouped together. Then from each cluster, EMMA-C selects the one with the highest expected gain. As shown, EMMA-C will result in selecting the points that are further from each other and less similar. Therefore, EMMA-C selects more diverse samples compared to EMMA-M, resulting in less redundancy of information in the selected set of samples.

4 Validation

This section presents our validation approach for assessing the performance of our mindful active learning algorithm, comparing our approach to other methods, and evaluating the performance of our active learning in batch mode. Throughout this section, we use the term EMMA to refer to the greedy algorithm presented in Section 3.4. We note that the naive approach discussed in Section 3.3 has an exponential time complexity, which makes it impractical given the large amounts of data collected during continuous health monitoring using wearable sensors.

4.1 Datasets

To assess the performance of EMMA, we used three real-world, publicly-available, sensor-based datasets including HART [39] containing sensor data collected from 30 human subjects during six activities, DAS [40], [41], [42] containing sensor data from 8 human subjects and 10 activities, and AReM [43] featuring 6 activities by one human subject. These datasets are available at the UCI machine learning repository and are prepared for the goal of human activity recognition.

The first dataset, HART, is gathered using a smartphone [39] and 30 subjects have participated in this experiment while performing 6 different activities including walking, walking_upstairs, walking_downstairs, sitting, standing, lying. Overall, 561 features in time and frequency domains were calculated for 7352 data points in the training set and 2948 data points in the test set.

The second dataset is referred to as DAS (the daily and sports activities dataset) [40], [41], [42], recorded with 8 subjects, performing several daily and sport activities using three different sensors mounted on five different units recording 9120 data samples. We used the data from the 3-axis accelerometer mounted on the right leg and considered 10 different classes of activities. These activities include sitting, standing, lying on back, ascending stairs, descending stairs, running on a treadmill with a speed of 8 km/h, exercising on a stepper, cycling on an exercise bike in horizontal position, rowing, and jumping. We extracted 10 commonly-used features from the raw sensor data for further analysis. These features include max, min, amplitude, median, mean, peak to peak amplitude, variance, and RMS (Root Mean Square) power.

The third dataset that we used is referred to as AReM [43]. The data in this dataset were recorded with one subject while performing 6 different activities. The activities include bending, cycling, lying, sitting, standing and walking. The data were recorded using three sensors placed on chest, left ankle, and right ankle. The features extracted from the sensor data include mean value and standard deviation for each reciprocal RSS (Received Signal Strength) reading from the sensors, resulting in 6 features with a total of 2880 sensor observations or data samples.

A summary of the statistics for these three datasets is shown in Table 1.

TABLE 1
Datasets Characteristics

	# features	# activities	# subjects	# sensors
HART	**561**	**6**	**30**	**6**
DAS	**10**	**10**	**8**	**3**
AReM	**6**	**6**	**1**	**3**

4.2 Choice of Classifier

Throughout our experiments, we used the SVM (Support Vector Machine) as our underlying activity recognition classifier. This choice was made based on extensive experiments on our datasets explained in Section 4.1 to select the classifier that has the best performance on average over our data.

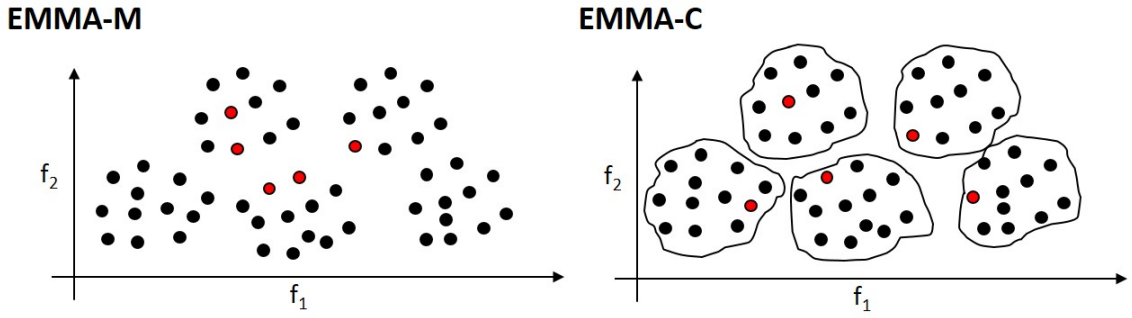

Fig. 2. Comparing EMMA-M and EMMA-C algorithms for a sample dataset with two features. Red points present the samples selected by each algorithm.

We considered four types of machine learning algorithms including SVM with linear kernel, SVM with RBF kernel, logistic regression, and decision tree. The average accuracy values achieved when testing different classifiers over three datasets are presented in Table 2.

TABLE 2
Classification accuracy achieved by four machine learning algorithms.

	SVM_Lin.	SVM_RBF	LR	DT
HART	**0.90**	**0.81**	**0.86**	**0.85**
DAS	**0.94**	**0.85**	**0.94**	**0.92**
AReM	**0.73**	**0.67**	**0.70**	**0.69**

As shown in Table 2, the SVM with linear kernel, SVM with RBF kernel, logistic regression, and decision tree algorithms achieved an overall accuracy of 92%, 83%, 90%, and 88%, respectively. The accuracy numbers were computed over all subjects in each dataset. We chose linear SVM as our activity recognition classifier. However, note that the methodologies presented in this article are independent of the choice of an activity recognition classifier.

4.3 Experimental Setup

For validation, the sensor data collected for each subject was considered separately. The data associated with each subject was split into two sets with 70% as the training set and 30% as the test set. Moreover, the training set was divided into two sets of labeled samples $\mathcal{Z}$ and unlabeled ones $\mathcal{X}$. At the beginning of each experiment we assumed that the labeled set has two labeled samples from two different activities so that an initial activity recognition model can be generated for assigning entropy values to unlabeled set members.

Furthermore, we added another column to our datasets simulating the time passed after the activity at the time of querying. The time was presented as a fraction of a day in Equation 7. This value was used to calculate Memory Retention (R) as the probability of remembering the activity by the subject at the query time.

We simulated oracles with various memory strengths. The goal was to measure the likelihood of an incorrect label provided by the oracle given the memory strength and the time that had passed from capturing the queried sensor observation. To simulate the oracle's remembering of the event associated with a queried sensor observation X_i based on a given s value for the oracle, we first computed memory retention R_i for X_i using Equation (7). We then assigned the correct label with the probability R and incorrect label with the probability $(1 - R)$. An incorrect label is selected randomly from the set of available activity labels. To alleviate the effect of randomness in our simulation procedure, we repeated each experiment 30 times and report the average results.

4.4 Performance Metrics

We first focused on measuring the performance of the EMMA algorithm. For this purpose, we used the algorithm for creating the labeled training set as we change the maximum available querying budget B, to observe how the performance changes when the amount of the available budget varies. Moreover, we simulated users with various memory strengths to measure the changes in the performance when the probability of having noisy labels in the querying responses ranges from weak memories to strong memories. All three datasets were used for these experiments. The performance metrics used to evaluate the EMMA algorithm include accuracy, precision, recall, and F1-score [7].

4.5 Comparative Algorithms

The second step in our validation process was to compare the performance of EMMA with other methods. For comparative analysis, two more variations of EMMA, in addition to the upper-bound and lower-bound cases, were considered. We assessed the performance of EMMA against the following active learning approaches:

- **EMA** (Entropy Maximization): this approach solves the optimization problem only based on informativeness of the queried observations. It however considers the memory of the user when responding to queries with the goal of testing if considering only entropy is sufficient when user is limited in memory capacity. In this case, $\mathcal{E}(I_i)$=E_i and $\mathcal{E}(M_i) = 1$.
- **MMA** (Memory Maximization): this algorithm aims to maximize the objective function for memory retention only. This means, the algorithm tends to choose observations whose probability of remembering the

label by the oracle is higher than others. In this case, $\mathcal{E}(I_i)$=1 and $\mathcal{E}(M_i)$=R_i.

- **Upper-bound (UB)**: this refers to the case where no erroneous labels exist as a result of memory deficiency. It means that, we assume that the oracle's memory is perfect, as a result of which the optimization problem aims to maximize for entropy only.
- **Lower-bound (LB)**: this refers to the case where the oracle's memory is low and the observations are chosen randomly. Therefore, the informativeness of the queried observation is not considered as a parameter.

For comparative analysis, EMMA and the four above-mentioned algorithms were examined when varying budget numbers and by users with different memory strength values.

4.6 Batch Active Learning

The third step in our validation process was to observe how our greedy algorithm for EMMA performs when used in batch mode for active learning. In batch mode active learning, instead of selecting one sample in each iteration of the EMMA algorithm (referred to as EMMA-1 in this section of our validation), multiple samples are selected. As explained in Section 3.5, for this purpose, we propose two versions for batch-EMMA as follows.

- **EMMA-M**: this refers to the case where in each iteration of the active learning, multiple sensor observations (equal to batch-size) are selected to query the user. To this end, we select the sensor observations with the highest values of $\mathcal{E}(I_i, M_i)$.
- **EMMA-C**: this approach refers to the case where in each iteration in EMMA we first perform a clustering of the unlabeled sensor observations and select a number of the observations (equal to batch-size) from these clusters. The clustering algorithm groups the unlabeled sensor observations into several clusters where the number of clusters indictated by the batch-size. In the next step, one sample with the highest value of $\mathcal{E}(I_i, M_i)$ within each cluster is selected for querying the user. The advantage of this method over the EMMA-M method is that it intends to select the samples that are less similar to each other by clustering them in advance. Therefore, it is likely that the observations chosen from different clusters are more informative in terms of the machine learning task.

For this phase of our validation experiments, multiple batch-size values were used with a fix available budget value, when simulating two end-user, with weaker and stronger memory strengths.

5 Results and Discussions

In this section, we present results on the performance of EMMA for active learning, comparative analysis, and the performance of batch-EMMA active learning.

5.1 Performance of EMMA

As a first analysis, we evaluated the performance of EMMA using different metrics (i.e., accuracy, precision, recall, F1-score) and examined how these metrics change as various parameters of the algorithm change. To this end, we conducted multiple experiments by changing the algorithm parameters including query budget, B, and memory strength, s. After EMMA constructed a labeled dataset, we trained an activity recognition classifier using the labeled dataset and utilized the trained model over the test set to measure performance metrics for each of the three datasets discussed previously. The details of the platform which is used for running our analysis is as follows: Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz

Figure 3 shows the accuracy of the trained classifier on the three datasets with query budget ranging from 5 to 200. The graphs show the accuracy values for five different memory retention levels including R_1 (10%–99%), R_2 (20%–99%), R_3 (30%–99%), R_4 (50%–99%), and R_5 (70%–99%). Note that, as R refers to the probability of remembering a label for a given sensor observation, the R levels represent a range of possible memory retention values for all observations in the dataset. Note that different samples have different time stamps and thus, different time delays. Also, note that, as the time window size to capture each sensor observation from raw sensor readings varies across the datasets, we used different memory strength values on the three datasets to obtain the same memory retention intervals to provide comparable results. Furthermore, as stated before, we repeated each experiments 30 times and their averaged results are presented.

As Figure 3 suggests, the activity recognition accuracy improves for each memory retention value as the query allowance (i.e., budget) increases. Furthermore, as the memory strength grows, the accuracy improves for each budget value. The minimum accuracy, achieved with the least budget and weakest memory is 44%, 29%, and 21% for HART, DAS, and AReM datasets, respectively. The accuracy reaches its maximum value of 85.3%, 97.5%, and 70% with greatest budget and strongest memory for HART, DAS, and AReM datasets, respectively.

An interesting observation from Figure 3 is the performance decline in weak memory cases. It is generally known that increasing the number of labeled observations using active learning improves the classifier performance. However, we can see that adding a large number of labeled observations may reduce the accuracy if the memory retention is low due to a weak memory. An example of such cases is shown in Figure 3a for retention levels R_1, R_2, and R_3. In all these cases, the activity recognition performance improves up to a certain point (e.g., B=60) and it starts dropping after that point, mainly because more incorrect labels are added to the dataset resulting in a less accurate model. For instance, the accuracy for R_1, as the weakest memory, starts from 44% for B=5, reaches up to 63% for B=60, and again drops to 52.5% with B=200.

Another interesting observation is that the accuracy saturates after a certain point with strong memories. This can be observed, for example, in Figure 3a for memory retention levels R_4 and R_5 where the accuracy plateaus

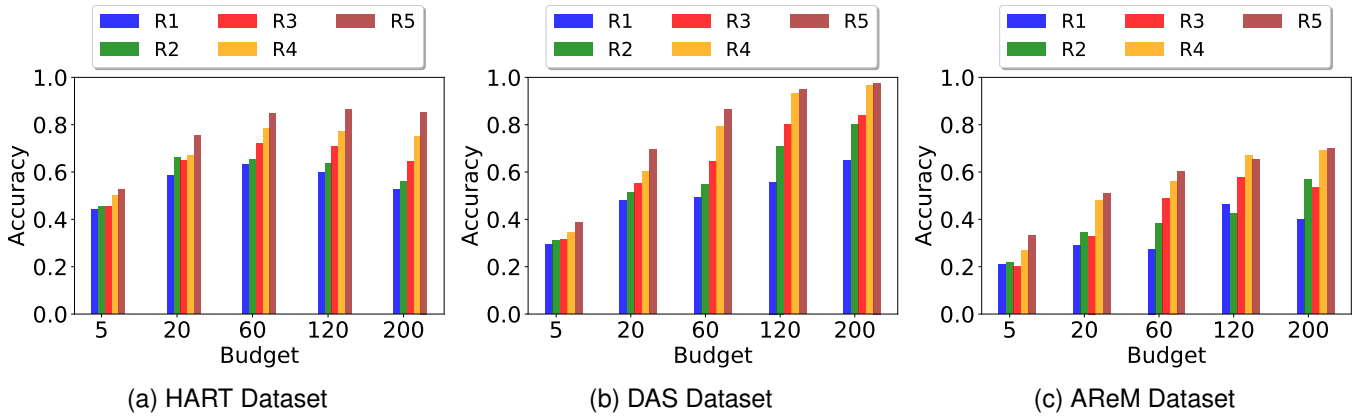

Fig. 3. Accuracy of activity recognition model trained using EMMA on three datasets (HART, DAS, and AReM) as a function of query budget for five different memory retention levels (R_1–R_5).

after acquiring 60 labeled observations and stays around 77% and 85% for higher budget values. The reason is that, in this dataset, each added observation is very informative and the classifier learns faster and needs a smaller number of observations to achieve its maximum capability.

The difference in accuracy among different datasets in Figure 3 can be explained by the quality of the data and the difficulty of the recognition task. For example, the DAS dataset (i.e., Figure 3b) contains highly discriminative features over different activities. As a result of this quality, adding a few correctly labeled observations has a significant impact on improving the classifier performance. For instance, when considering R_5, the change in accuracy when B grows from 5 to 20 is 23% for HART and 30% for DAS. In contrast, the AReM dataset (i.e., Figure 3c) contains less discriminative features and the window size over which the features are calculated is small, leading to needing a lot more labeled sensor observations to learn new activities. Therefore, the growth in the performance of the classifier in Figure 3c is slow compared to other datasets. Specifically, the amount of improvement in the accuracy is 17% for R_5 when B increases from 5 to 20 in the AReM dataset, while this improvement is 23% and 30% for the HART and DAS datasets, respectively.

We also computed the performances of EMMA in terms of precision, recall, and F1-score. The results are presented in Table 3 to Table 5.

Table 3, Table 4, and Table 5 suggest that the performance of EMMA in terms of precision, recall, and F1-score for HART, DAS, and AReM datasets follow similar trends to the accuracy measures discussed previously. In general, as the memory and budget values increase, the performance of EMMA improves. As can be seen from these tables, the performance of EMMA with the HART dataset starts to decrease with a budget value around 60, when the user's memory is weak, and the performance saturates when the user's memory is strong.

The precision, recall, and F1-score metrics for the HART dataset achieve their minimum values of 0.46, 0.31 and 0.35, respectively, when the smallest budget and weakest memory value are used. Similarly, the maximum values of precision, recall, and F1-score, which are 0.88, 0.88 and 0.87, are achieved with the greatest budget and strongest memory.

As for the DAS dataset, as shown in Table 4, the minimum values for precision, recall, and F1-score are 0.29, 0.10 and 0.14 and the maximum of these values are 0.98, 0.98 and 0.98. Furthermore, the minimum values of precision, recall, and F1-score for the AReM dataset are 0.21, 0.08 and 0.11, respectively, with smallest budget and weakest memory. As shown in Table 5, the maximum values for precision, recall, and F1-score for the AReM dataset are 0.64, 0.61 and 0.60, respectively. These values are obtained with the largest budget and strongest memory.

5.2 Comparative Analysis

We compared the performance of EMMA with that of several active learning approaches including EMA, MMA, upper-bound (UB), and lower-bound (LB). See Section 4.5 for a description of these alternative approaches. For brevity, we focus on accuracy as our main performance measure here. Similar to our analysis in the previous section, we examined the performance of each algorithm while the budget value ranged from 0 to 200 and the memory strength was set such that different memory retention levels, R_1 (10%–99%), R_2 (25%–99%), R_3 (50%–99%), and R_4 (70%–99%) are obtained. The results of this analysis are shown in Figure 4, Figure 5, and Figure 6 for HART, DAS, and AReM datasets, respectively. The accuracy numbers in these graphs represent average values computed over all the users in each dataset.

Each subfigure in Figure 4–Figure 6 represents the accuracy performance for a particular memory retention level (i.e., R_1, R_2, R_3, or R_4), and the x-axis in each graph shows the query budget.

The results show that the performance improvement due to using EMMA over EMA and MMA is most notable when the amount of budget is small and the memory is weak (i.e., memory retention level of R_1). This observation emphasizes the importance of considering both uncertainty and memory retention in health monitoring applications where data collection is extremely costly and end-users are likely to be cognitively impaired. It can be seen from the results that EMMA achieves an average accuracy that is 13.5% higher than that of EMA and 14% higher than that of MMA for cases of weaker memory and smaller budget.

TABLE 3
Precision, recall, and F1-score of EMMA on HART dataset.

Budget	Precision					Recall					F1 score				
	R_1	R_2	R_3	R_4	R_5	R_1	R_2	R_3	R_4	R_5	R_1	R_2	R_3	R_4	R_5
5	0.46	0.48	0.47	0.52	0.54	0.31	0.34	0.32	0.41	0.44	0.35	0.37	0.36	0.42	0.45
20	0.61	0.68	0.66	0.68	0.76	0.59	0.65	0.63	0.68	0.75	0.56	0.62	0.60	0.64	0.72
60	0.65	0.67	0.73	0.79	0.85	0.65	0.67	0.74	0.81	0.87	0.62	0.64	0.71	0.78	0.84
120	0.61	0.65	0.72	0.78	0.87	0.62	0.65	0.73	0.79	0.88	0.59	0.63	0.70	0.77	0.86
200	0.53	0.57	0.65	0.79	0.88	0.54	0.58	0.67	0.79	0.88	0.52	0.56	0.64	0.78	0.87

TABLE 4
Precision, recall, and F1-score of EMMA on DAS dataset.

Budget	Precision					Recall					F1 score				
	R_1	R_2	R_3	R_4	R_5	R_1	R_2	R_3	R_4	R_5	R_1	R_2	R_3	R_4	R_5
5	0.29	0.31	0.32	0.34	0.39	0.10	0.12	0.12	0.14	0.21	0.14	0.17	0.17	0.19	0.25
20	0.48	0.51	0.55	0.60	0.69	0.33	0.39	0.43	0.50	0.61	0.38	0.42	0.46	0.52	0.62
60	0.49	0.55	0.64	0.79	0.87	0.44	0.49	0.59	0.78	0.87	0.43	0.49	0.59	0.76	0.85
120	0.56	0.71	0.80	0.93	0.95	0.51	0.69	0.79	0.94	0.96	0.50	0.67	0.77	0.93	0.95
200	0.65	0.80	0.84	0.96	0.98	0.618	0.79	0.82	0.97	0.98	0.60	0.78	0.81	0.96	0.98

TABLE 5
Precision, recall, and F1-score of EMMA on AReM dataset.

Budget	Precision					Recall					F1 score				
	R_1	R_2	R_3	R_4	R_5	R_1	R_2	R_3	R_4	R_5	R_1	R_2	R_3	R_4	R_5
5	0.21	0.22	0.17	0.27	0.33	0.08	0.08	0.12	0.18	0.18	0.11	0.11	0.07	0.20	0.22
20	0.29	0.34	0.33	0.48	0.51	0.18	0.27	0.23	0.42	0.45	0.19	0.26	0.23	0.39	0.42
60	0.27	0.38	0.49	0.56	0.60	0.21	0.28	0.48	0.57	0.60	0.21	0.29	0.43	0.54	0.55
120	0.46	0.43	0.58	0.69	0.64	0.42	0.41	0.61	0.70	0.61	0.38	0.37	0.54	0.68	0.60
200	0.45	0.57	0.57	0.69	0.64	0.41	0.53	0.61	0.69	0.61	0.38	0.52	0.52	0.66	0.60

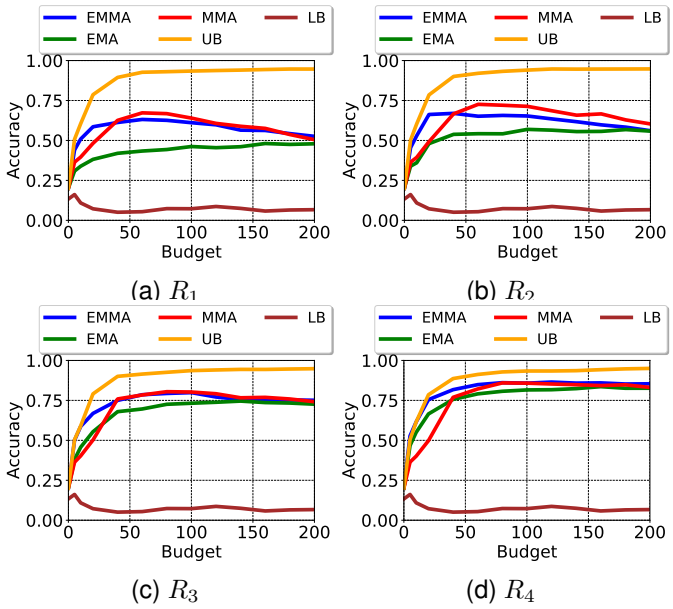

Fig. 4. Activity recognition accuracy of various active learning algorithms on HART dataset using four different memory retention levels, R_1, R_2, R_3, and R_4.

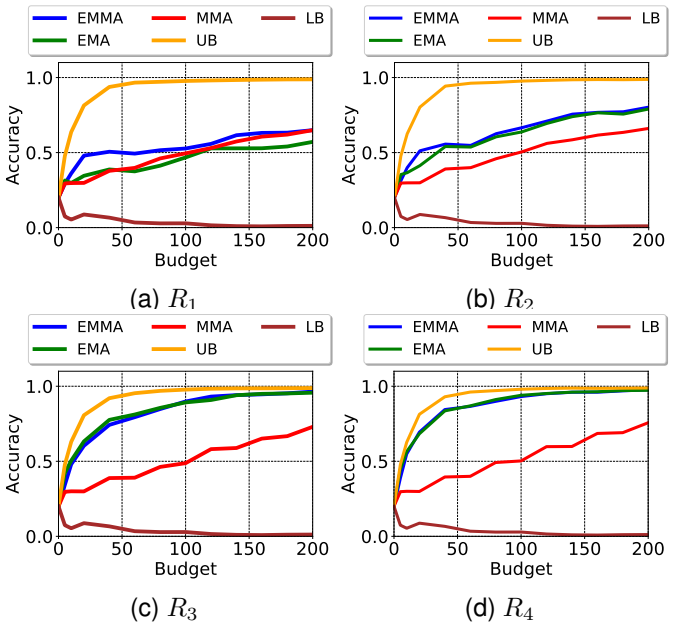

Fig. 5. Activity recognition accuracy of various active learning algorithms on DAS dataset using four different memory retention levels, R_1, R_2, R_3, and R_4.

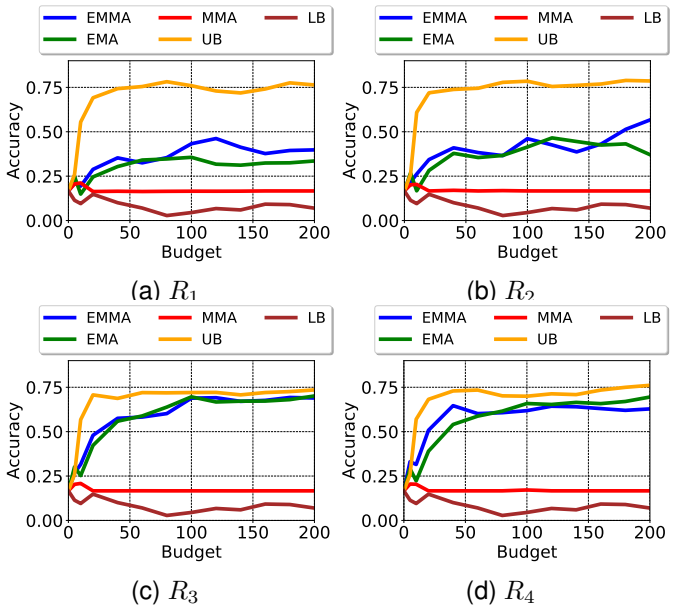

Fig. 6. Activity recognition accuracy of various active learning algorithms on AReM dataset using four different memory retention levels, R_1, R_2, R_3, and R_4.

Moreover, we note that, for all datasets, as a strong memory is used, EMMA and EMA methods converge and achieve accuracy values closer to the upper-bound. Additionally, the performance of EMMA is at most 20% less than the experimental upper-bound and up to 80% higher than the experimental lower-bound, on average.

For the HART dataset, the performance of MMA converges to those of EMMA and EMA. However, for the other datasets the performance of MMA improves slower. The reason is that MMA tends to select the observations sequentially in time and ignores the informativeness of the observations. Therefore, if the user repeats an activity for a long time or the activity does not change (e.g., 'sleeping', 'watching TV'), MMA continues to query for the same activity over and over again, resulting in a model that is incapable of recognizing a wide range of activities. Consequently, the active learning process needs to query for many new observations in order to learn new activities, which is the case for the DAS and AReM datasets. Note, as mentioned previously, AReM contains less discriminative features compared to the two other datasets. This is the reason why the performance of MMA is closer to the lower-bound on AReM.

Figure 4 to Figure 6 can be used to compare the overall performance of EMMA, EMA, and MMA. In Figure 4, MMA, which selects samples based on their time order, outperforms EMMA slightly for larger budgets, but it is much worse than EMMA in Figure 6. This is because of fast switching between unknown activities in the HART dataset, helping MMA to learn new activities quickly (while the AReM dataset switches much slower from one activity to another). These observations suggests that the performance of MMA is highly dependent on the dataset or, in fact, on the order in which the physical activities occur.

Although EMMA outperforms EMA in Figure 4, these two algorithms achieve a similar performance in Figure 6.

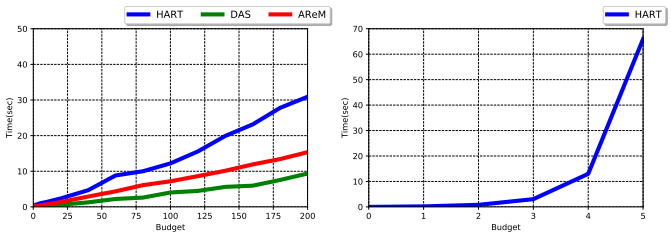

Fig. 7. Run-time of greedy approach for EMMA vs. brute-force approach: a) run-time of EMMA for three different datasets as a function of budget; b) run-time of brute-force solution for HART dataset as a function of budget.

This observation can be explained as follows. The activity sequence changes frequently in the HART dataset (Figure 4). These frequent activity transitions allow EMMA to query activity data that are more likely to be labeled correctly because such sensor observations exist closer to the query time. Because EMA does not take into consideration memory retention, it queries sensor observations that might have been captured far in the past and therefore are less likely to be labeled correctly. Figure 5 represents the case in between these two scenarios where EMMA has the most accurate and stable performance compared to the competing algorithms.

5.3 Time Complexity Analysis

In this section we present our results on the run-time of the greedy approach for EMMA as well as that of the brute-force approach. First, for the greedy algorithm, we measured the run-time for all three datasets when changing the budget and then averaged the results over all the users and number of iterations. The results are shown in Figure 7a. As it can be observed from these graphs, the time complexity of EMMA is linear in the the amount of query budget, which is consistent with our theoretical analysis in Section 3.4.

In next step, to experiment the time complexity of the brute-force solution, we made use of the HART dataset. It is infeasible to run this method for big budget numbers, as it will take very long to create all the ordered subsets for each budget value. Therefore, we limited the budget number to range from 1 to 5 only for this analysis. The results for run-time are shown in Figure 7b, again averaged over all users and iterations. As we showed in (12), the number of ordered subsets increases exponentially and it can also be seen in the run-time of the brute-force solution in this graph. This again confirms the efficiency of EMMA over the brute-force algorithm.

5.4 Performance of EMMA for Batch Active Learning

In this section, the performance of EMMA as an active learning approach in batch-mode is presented. As discussed previously, in batch-mode active learning, instead of selecting one sample in each iteration of the EMMA algorithm (referred to as EMMA-1 in this section), multiple sensor observations are selected (EMMA-M and EMMA-C). See section Section 4.6 for a description of batch-mode EMMA algorithms. In this manuscript, k-means clustering algorithm was used for clustering the sensor observations in the EMMA-C active learning approach [44].

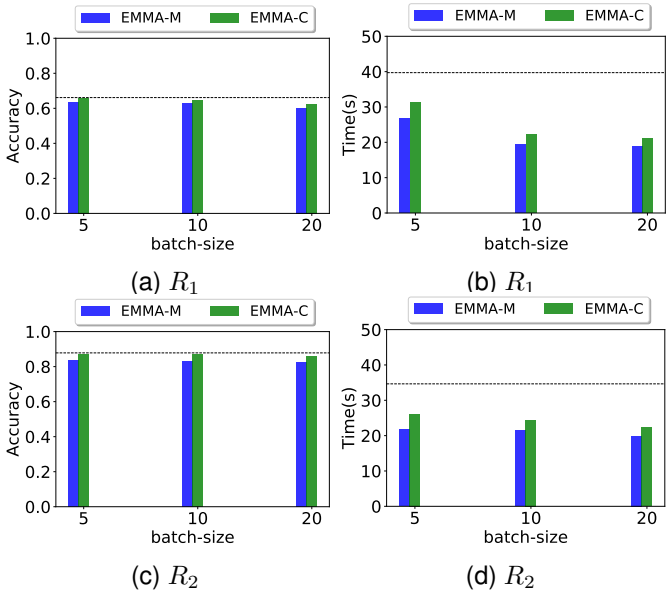

Fig. 8. Activity recognition accuracy and time of Batch-EMMA active learning algorithms on HART dataset using two different memory retention levels, R_1 (low) and R_2 (high), with horizontal line showing accuracy and time for EMMA-1 algorithm.

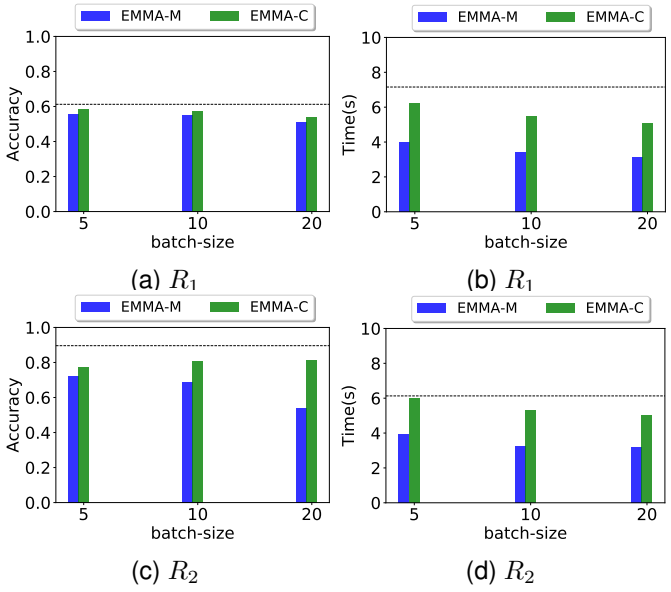

Fig. 9. Activity recognition accuracy and time of Batch-EMMA active learning algorithms on DAS dataset using two different memory retention levels, R_1 (low) and R_2 (high), with horizontal line showing accuracy and time for EMMA-1 algorithm.

In order to evaluate the performance of the EMMA-M and EMMA-C methods, we chose three different batch sizes of 5, 10, and 20 with a fixed budget value of 60 for our experiments. Furthermore, one small memory strength level and one large memory strength value, representing weak and strong memories, were considered. These memory retention levels were R_1 (15%–99%) and R_2 (70%–99%).

Then the EMMA-M and EMMA-C methods were applied to the three datasets, and the accuracy achieved as well as the run-time of each algorithm per subject and per iteration are presented in Figure 8 to Figure 10. Moreover, the accuracy and time for the EMMA-1 algorithm is presented as a horizontal dashed line in each graph as a reference.

Considering Figure 8 to Figure 10, we see that EMMA-M has a smaller run-time complexity compared to EMMA-C,

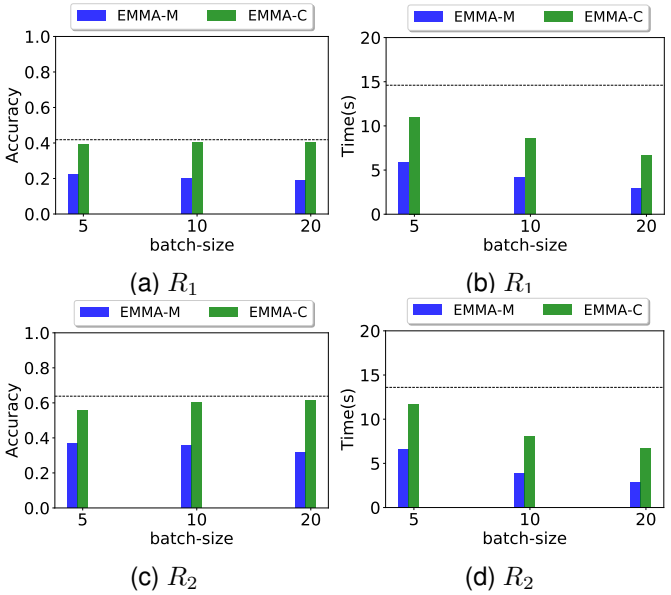

Fig. 10. Activity recognition accuracy and time of Batch-EMMA active learning algorithms on AReM dataset using two different memory retention levels, R_1 (low) and R_2 (high), with horizontal line showing accuracy and time for EMMA-1 algorithm.

which is anticipated because EMMA-C uses clustering as an extra step. By calculating the time difference between the two algorithms for various batch size and R values and averaging them, it can be seen that EMMA-C consumes 3.23 sec., 2.02 sec., and 4.40 sec. longer than EMMA-M for HART, DAS and AReM datasets, respectively.

It can be also seen that both EMMA-M and EMMA-C algorithms require less time to complete compared to EMMA-1. This is rather expected because EMMA-1 selects sensor observations one by one and requires a round of machine learning algorithm training during each iteration of the algorithm. The effect of adding each sensor observation to the training set is evaluated on the entropy of the remaining sensor observations and consequently, their expected gain, ($\mathcal{E}$), one by one. However, in batch-mode EMMA, the time for this process is reduced by evaluating the effect of adding multiple sensor observations at a time on the remaining observations expected gain, ($\mathcal{E}$). However, it should be noted that, since EMMA-C has an extra step for clustering, it is possible that for a very large dataset, where more and more clusters are needed, the clustering time would dominate the run-time for sequential learning in EMMA-1.

By calculating the time difference between the two batch algorithms and EMMA-1 for various batch sizes and R values, we obtained the following results. EMMA-M and EMMA-C require 12.6 sec. and 15.9 sec. less time than EMMA-1 on the HART dataset. EMMA-M and EMMA-C consume 2.2 sec. and 2.1 sec. less time compared to EMMA-1 on the DAS dataset; and they need 9.7 sec. and 5.3 sec. less time compared to EMMA-1 on the AReM dataset. We note that the differences between the time results for the three datasets are due to the complexity of each problem within the associated dataset, which is determined by factors such as number of features and number of sensor observations among others.

Figure 8 to Figure 10 suggest that the achieved classifi-

cation accuracy of EMMA-M is less than EMMA-C. This is an expected outcome because EMMA-C tries to select more diverse and less similar observations among all unlabeled data to be added to the labeled training set. Therefore, in each iteration of the algorithm less redundant and newer observations are added to the training set, resulting in an increase in the classification performance.

We also computed the difference between the accuracy of the two algorithms for various batch sizes and R values. The overall results are as follows. EMMA-C achieved 2.6%, 8.8% and 22.0% higher accuracy than EMMA-M, on average, on the HART, DAS, and AReM datasets, respectively. The differences in the accuracy results across the three datasets suggest that the AReM dataset has more similar data samples compared to two other datasets, and therefore EMMA-C achieves a higher accuracy on AReM. Furthermore, it indicates that the DAS dataset has more redundant samples compared to HART.

Both EMMA-M and EMMA-C achieve less accurate results than EMMA-1 on average. The accuracy of EMMA-M is generally less than that of EMMA-1 because EMMA-1 evaluates the effect of adding each observation on the expected gain for the remaining observations one-by-one and, therefore, is more accurate. By calculating the the difference between the achieved accuracy of EMMA-1 and that of EMMA-M for various batch sizes and R values, it can be seen that EMMA-1 obtains 4.4%, 15.8% and 25.4% higher accuracy compared to EMMA-M on the HART, DAS and AReM datasets, respectively.

However, the performance of EMMA-C is usually close to that of EMMA-1. The reason is that EMMA-C considers the similarity of important samples and by considering redundancy, tries to select the important and at the same time less similar samples for adding to the training set. Therefore, based on the nature of the dataset and how similar its significant samples are, the achieved accuracy of the trained model can be improved. By calculating the difference between the achieved accuracy of the EMMA-1 and EMMA-C algorithms for various batch sizes and R values and averaging them for all datasets, it can be seen that EMMA-1 achieves 1.4%, 7.7%, and 3.3% higher accuracy compared to EMMA-C on the HART, DAS, and AReM datasets, respectively.

By comparing the accuracy for the three datasets, we observe that using EMMA-M over EMMA-1 results in a higher performance decline in the AReM dataset compared to the HART or DAS datasets. However, when utilizing the EMMA-C algorithm, this difference in accuracy decline has been reduced. Therefore, it is concluded that AReM has more similar and redundant samples and, therefore, EMMA-C can help with its accuracy more compared to HART or AReM.

Moreover, it can be seen that as the batch size increases, both time and accuracy results decrease with the EMMA-M algorithm. The decrease in the running time of the algorithm is due to the fact that adding more sensor observations at each iteration of the algorithm leads to a smaller number of algorithm retraining tasks. The accuracy loss can also be explained by the fact that adding a group of sensor observations, rather than adding them one-by-one, eliminates the chance for learning from all labeled data. For EMMA-C, the algorithm is more time-efficient compared to EMMA-1 based on our experiments. However, there is a chance that, with a larger dataset and number of clusters, the algorithm time increases compared to EMMA-1 and for each dataset. The optimal number of clusters that leads to maximum accuracy differs based on its nature.

Overall, batch-EMMA algorithms are more effective and recommended when working with larger dataset. In such cases, utilizing EMMA-1 may be time-consuming. Furthermore, for larger datasets, the efficiency of batch-EMMA algorithms in terms of time-efficiency, compared to EMMA-1, will be more considerable. Moreover, the choice between EMMA-M and EMMA-C is application-based since EMMA-M is a more time-efficient approach while EMMA-C is mainly focused on accuracy performance.

6 Future Work

In this paper, we focused on pool-based active learning. Our ongoing work involves developing mindful active learning strategies that make query decisions on-the-fly as wearable sensor data become available in real-time.

In our experiments we assumed that the time delay is equal to the difference between query time and sampling time (i.e., when the sensor is sampled by the microcontroller). However, it is possible that the user may not respond to the query immediately. If the user prefers to proactively initiate the labeling process, the active learning process needs to recompute the queried observations by adding the time difference between query time and annotation time to reflect the time delay in our formulation. Furthermore, if there are multiple sessions in a day that the user intend to annotate the sensor data, this model can be used for each session separately. We also plan to investigate active learning solutions that take into account the possibility of delayed responses through context-sensitive active learning.

In this work, we simulated the memory strength of the end-user for validation purposes. Our future work also focuses on conducting user studies that involve cognitive assessment of the user where we will assess the oracle's memory retention quantitatively.

7 Conclusions

Prior research on active learning takes informativeness of data and query budget into account when selecting the data for query. In this paper, we showed that cognitive constraints of the oracle are of significant importance that can greatly compromise active learning performance. We posed an optimization problem to combine data uncertainty with memory retention for use in ubiquitous and mobile computing applications. We derived a greedy approximation algorithm to solve the proposed mindful active learning problem. Our extensive analyses on three publicly available datasets showed that EMMA achieves up to 97% accuracy for activity recognition using wearable sensors. We also showed that integrating memory retention improves the active learning performance by 16%.

Our results indicate that the performance of EMMA improves when the oracle's memory is stronger or the query budget is higher. However, we noticed that increasing the

budget does not improve the accuracy when highly accurate labels are available due to strong memory retention. We also noted that the active learning performance can decrease with increased budget if the oracle's memory is weak.

Moreover, by comparing the performance of EMMA with multiple competing algorithms as its variations, we showed that including both informativeness of samples and memory effects on noisy and incorrect labels, results in EMMA being less dependent on the dataset compared to other algorithms. This indicates that the results obtained by EMMA are more consistent across different datasets and machine learning tasks. Finally, the gap between the performance of EMMA and other algorithms is most notable with small budgets and weak memories.

Moreover, by comparing the performance of EMMA with its batch versions, we showed that batch-EMMA results in needing less time to perform the active learning computation at a reduced accuracy performance. However the nature of a dataset is also a factor in the batch-EMMA algorithm performance. We also observed that batch-EMMA is more effective with larger datasets.

ACKNOWLEDGMENTS

This work was supported in part by the United States National Science Foundation under grant CNS-1932346. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations.

REFERENCES

[1] J. Gubbia, R. Buyyab, S. Marusica, and M. Palaniswamia, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, Sep. 2013.

[2] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the internet of things," *IEEE Access*, vol. 6, nov 2017.

[3] T. L. Koreshoff, T. W. Leong, and T. Robertson, "Approaching a human-centred internet of things," in *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*, Adelaide, Australia, Nov. 2013.

[4] I. Kotenko, F. Skorik, and S. Bushuev, "Neural network approach to forecast the state of the internet of things elements," in *Proceedings of the XVIII International Conference on Soft Computing and Measurements (SCM)*, St. Petersburg, Russia, Jun. 2015.

[5] P. Ni, C. Zhang, and Y. Ji, "A hybrid method for short-term sensor data forecasting in internet of things," in *Proceedings of the 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, Xiamen, China, Jun. 2014.

[6] M. Rofouei, M. Pedram, F. Fraternali, Z. Esna Ashari, and H. Ghasemzadeh, "Resource-efcient computing in wearable systems," in *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 2019.

[7] R. Fallahzadeh and H. Ghasemzadeh, "Personalization without user interruption: Boosting activity recognition in new subjects using unlabeled data," in *2017 ACM/IEEE 8th International Conference on Cyber-Physical Systems (ICCPS)*, April 2017, pp. 293–302.

[8] B. Settles, "Active learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 6, no. 1, 2012.

[9] P. Bachman, A. Sordoni, and A. Trischler, "Learning algorithms for active learning," in *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, Aug. 2017.

[10] S. Redmond, N. Lovell, G. Yang, A. Horsch, P. Lukowicz, L. Murrugarra, and M. Marschollek, "What does big data mean for wearable sensor systems?" *Yearbook of medical informatics*, vol. 23, no. 01, pp. 135–142, 2014.

[11] Z. Esna Ashari and H. Ghasemzadeh, "Mindful active learning," in *Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*. IJCAI, 2019.

[12] H. Ebbinghaus, "Memory: A contribution to experimental psychology," *Annals of Neurosciences*, 1885.

[13] S.-J. Huang, R. Jin, and Z.-H. Zhou, "Active learning by querying informative and representative examples," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 10, Oct. 2014.

[14] S. Sabato and T. Hess, "Interactive algorithms: from pool to stream," in *Proceedings of the 29th Annual Conference on Learning Theory (COLT), JMLR Workshop and Conference Proceedings*, New-York City, USA, Jun. 2016.

[15] Z. Xu, R. Akella, and Y. Zhang, "Incorporating diversity and density in active learning for relevance feedback," in *Proceedings of 29th European Conference on IR Research*, Rome, Italy, Apr. 2007.

[16] K.-S. Jun and R. Nowak, "Graph-based active learning: A new look at expected error minimization," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Washington, DC, USA, Dec. 2016.

[17] S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu, "Batch mode active learning and its application to medical image classification," in *23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

[18] X. Shen and C. Zhai, "Active feedback in ad hoc information retrieval," in *28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Salvador, Brazil, Aug. 2005.

[19] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," in *ICLR 2018, The Sixth International Conference on Learning Representations*, Vancouver, Canada, May 2018.

[20] K. Konyushkova, R. Sznitman, and P. Fua, "Learning active learning from data," in *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, Long Beach, CA, USA, Dec. 2017.

[21] Z. Wang, B. Du, W. Tu, L. Zhang, and D. Tao, "Incorporating distribution matching into uncertainty for multiple kernel active learning," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[22] S. Vijayanarasimhan, P. Jain, and K. Grauman, "Far-sighted active learning on a budget for image and video recognition," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, USA, Feb. 2016.

[23] M. Stikic and B. Schiele, "Activity recognition from sparsely labeled data using multi-instance learning," in *Proceedings of 4th International Symposium on Location- and Context-Awareness*, Tokyo, Japan, May 2009.

[24] T. Diethe, N. Twomey, and P. Flach, "Active transfer learning for activity recognition," in *Proceedings of proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, Bruges, Belgium, Apr. 2016.

[25] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *International Conference on Pervasive Computing*, Linz/Vienna, Austria, Apr. 2004.

[26] S. Sinha, S. Ebrahimi, and T. Darrell, "Variational adversarial active learning," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, South Korea, Nov. 2019.

[27] P. K. Murukannaiah and M. P. Singh, "Platys: An active learning framework for place-aware application development and its evaluation," *ACM Transactions on Software Engineering and Methodology*, vol. 24, no. 3, May 2015.

[28] R. K. Sah and H. Ghasemzadeh, "Adversarial transferability in wearable sensor systems," *arXiv*, 2020.

[29] S. A. Rokni and H. Ghasemzadeh, "Share-n-learn: A framework for sharing activity recognition models in wearable systems with context-varying sensors," *ACM Transactions on Design Automation of Electronic Systems*, aug 2019.

[30] P. Alinia, I. Mirzadeh, and H. Ghasemzadeh, "Actilabel: A combinatorial transfer learning framework for activity recognition," *arXiv*, 2020.

[31] R. Fallahzadeh, P. Alinia, and H. Ghasemzadeh, "Learning active learning from data," in *The 36th IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, Irvine, CA, USA, Oct. 2017.

[32] Y. Ma, Z. Esna Ashari, M. Pedram, N. Amini, D. Tarquinio, K. Nouri-Mahdavi, M. Pourhomayoun, R. D. Catena, and H. Ghasemzadeh, "Cyclepro: A robust framework for domain-agnostic gait cycle detection," *IEEE Sensors Journal*, vol. 19, no. 10, pp. 3751–3762, 2019.

[33] T. Shany, S. Redmond, M. Narayanan, and N. Lovell, "Sensors-based wearable systems for monitoring of human movement and falls," *Sensors Journal, IEEE*, vol. 12, no. 3, pp. 658 –670, march 2012.
[34] H. M. S. Hossain, N. Roy, and M. A. A. H. Khan, "Active learning enabled activity recognition," in *Proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Sydney, NSW, Australia, 2016.
[35] F. Shahmohammadi, A. Hosseini, C. E. King, and M. Sarrafzadeh, "Smartwatch based activity recognition using active learning," in *Proceedings of IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, Philadelphia, PA, USA, Aug. 2017.
[36] J. Murre and J. Dros, "Replication and analysis of ebbinghaus forgetting curve," *PLOS ONE*, vol. 10, no. 7, July 2015.
[37] S. G. Hu, Y. Liu, T. P. Chen, Z. Liu, Q. Yu, L. J. Deng, Y. Yin, and S. Hosaka, "Emulating the ebbinghaus forgetting curve of the human brain with a nio-based memristor," *Applied Physics Letter*, vol. 103, 2013.
[38] P. Harrington, *Machine Learning in Action*. 3 Lewis Street Greenwich, CT United States: Manning Publications Co., 2012.
[39] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," in *21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN*, Bruges, Belgium, Apr. 2013.
[40] K. Altun, B. Barshan, and O. Tunel, "Comparative study on classifying human activities with miniature inertial and magnetic sensors," *Pattern Recognition*, vol. 43, no. 10, pp. 3605–3620, Oct. 2010.
[41] B. Barshan and M. C. Yksek, "Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units," *The Computer Journal*, vol. 57, no. 11, pp. 1649–1667, Nov. 2014.
[42] K. Altun and B. Barshan, "Human activity recognition using inertial/magnetic sensor units," in *First International Workshop on Human Behavior Understanding*, Istanbul, Turkey, Aug. 2010.
[43] F. Palumboa, C. Gallicchiob, R. Pucciband, and A. Micheli, "Human activity recognition using multisensor data fusion based on reservoir computing," *Journal of Ambient Intelligence and Smart Environments*, vol. 8, no. 2, p. 87107, Oct. 2016.
[44] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967.

Zhila Esna Ashari received the B.Sc. degree in electrical engineering from the University of Tehran, Tehran, Iran, and the M.Sc. degree in electrical engineering, communications from the Sharif University of Technology, Tehran, Iran. She has received her Ph.D. degree in computer science from Washington State University, Pullman, WA, USA. Her research interests are machine learning and data mining and their applications in active learning, Internet of Things and bioinformatics.

Naomi S. Chaytor received a B.Sc. degree (1998) in Psychology from the University of Victoria, Victoria, Canada, an M.S. (2000) and Ph.D. (2004) degree in Clinical Psychology from Washington State University, Pullman, WA, USA, pre-doctoral internship (2004) from Baylor College of Medicine, Houston, TX, USA and post-doctoral fellowship (2005) in Neuropsychology and Rehabilitation Psychology from the University of Washington School of Medicine, Seattle, WA, USA. She was previously an Assistant Professor at the University of Washington School of Medicine Department of Neurology (2005-2013) and Department of Physical Medicine and Rehabilitation (2013-1015). She has been Board Certified in Clinical Neuropsychology since 2007. She is currently an Associate Professor (with tenure) in the Elson S. Floyd College of Medicine, Washington State University, Spokane, WA, USA. Her research is focused on cognitive and emotional functioning, and the use of technology to improve the lives of individuals with chronic medical conditions, particularly adults and older adults with type 1 diabetes.

Diane J. Cook is a Huie-Rogers Chair Professor at Washington State University. She received her Ph.D. in computer science from the University of Illinois and her research interests include machine learning, pervasive computing, and design of automated strategies for health monitoring and intervention. She is an IEEE Fellow and a Fellow of the National Academy of Inventors.

Hassan Ghasemzadeh is an Associate Professor of Computer Science in the School of Electrical Engineering and Computer Science at Washington State University. He received the B.Sc. degree from Sharif University of Technology, Tehran, Iran, the M.Sc. from University of Tehran, Tehran, Iran, and his Ph.D. from the University of Texas at Dallas, Richardson, TX, in 1998, 2001, and 2010 respectively. The focus of his research is algorithm design, machine learning, system-level optimization, and mobile health. He is a senior member of the IEEE.