DrinkWatch: A Mobile Wellbeing Application Based on Interactive and Cooperative Machine Learning

Simon Flutura, Andreas Seiderer, Ilhan Aslan, Chi Tai Dang, Raphael Schwarz, Dominik Schiller and Elisabeth André
University of Augsburg
Augsburg, Germany
{lastname}@hcm-lab.de

ABSTRACT

We describe in detail the development of DrinkWatch, a wellbeing application, which supports (alcoholic and non-alcoholic) drink activity logging. DrinkWatch runs on a smartwatch device and makes use of machine learning to recognize drink activities based on the smartwatch's inbuilt sensors. DrinkWatch differs from other mobile machine learning applications by triggering feedback requests from its user in order to cooperatively learn the user's personalized and contextual drink activities. The cooperative approach aims to reduce limitations in learning performance and to increase the user experience of machine learning based applications. We discuss why the need for cooperative machine learning approaches is increasing and describe lessons that we have learned throughout the development process of DrinkWatch and insights based on initial experiments with users. For example, we demonstrate that six to eight hours of annotated real world data are sufficient to train a reliable base model.

CCS CONCEPTS

• **Human-centered computing → Ubiquitous and mobile computing systems and tools**;

KEYWORDS

Mobile Social Signal Processing, interactive Machine Learning

ACM Reference Format:
Simon Flutura, Andreas Seiderer, Ilhan Aslan, Chi Tai Dang, Raphael Schwarz, Dominik Schiller and Elisabeth André. 2018. DrinkWatch: A Mobile Wellbeing Application Based on Interactive and Cooperative Machine Learning. In *DH'18: 2018 International Digital Health Conference, April 23–26, 2018, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3194658.3194666

1 INTRODUCTION

Utilizing mobile devices to collect personal behavioral data has many benefits, such as concisely informing medical professionals of a patient's "in the wild" behavior, and help in identifying appropriate intervention methods. However, there are still multiple technical and conceptual concerns, considering the mobile collection and processing of health related data. A paradigmatic example is the concern for privacy. Other concerns relate to data processing approaches based on machine learning (ML) in mobile settings. While mobile ML can be a powerful tool, its downsides include potential performance issues in learning individual and contextual differences (e.g. [19]), and negative user experiences due to a lack of system transparency and loss of user control (e.g. [1, 12]). We believe that many of the concerns associated with mobile ML applications can be addressed by taking inspiration from Horvitz's idea of mixed-initiative systems [13] and implementing a cooperative style of ML, in which users are interactively integrated into the ML process. In addition, by enabling cooperative ML completely on a mobile device, one would be able to address privacy-related user concerns associated with outsourcing data storage and processing to non-personal devices and unknown locations.

Because of the important role that ML approaches for mobile (health) applications will play in the foreseeable future, there is a need to explore human-centered techniques and paradigms towards balancing human needs and experiences with "machine autonomy". The increasing importance of designing for wellbeing and contributing factors, such as (perceived) human autonomy and competence is highlighted by researchers, such as Calvo and Peters [7], who promote a paradigm shift towards "Positive Computing" and away from designing solely to increase productivity. While ML is a well studied field with many fellow researchers working on improving ML's performance and its impact on productivity increase for various domains, the field of mobile cooperative ML is rather unexplored. Towards exploring the potential benefits and limitation of mobile cooperative ML applications, we came to understand that an initial, but important step is exposing ourselves to the process of developing an exemplary application and gathering initial insights with users.

In the following section, we provide background on previous research including descriptions of relevant forms of ML. Then we present in detail the development process of the smartwatch application DrinkWatch as an exemplary mobile cooperative ML application, which aims to provide logging support of drink activities (i.e. taking alcoholic or non-alcoholic liquids) throughout a day as a behavior. We conclude by discussing lessons learned and guidelines for the development of mobile cooperative ML applications, such as how the performance of different ML algorithms (i.e. Naive Bayes vs. linear SVM from LibLinear) is related to their learning curve with simulated user interaction.

DH'18, April 23–26, 2018, Lyon, France
https://doi.org/10.1145/3194658.3194666

2 BACKGROUND

Arguably, the advent of mobile and ubiquitous technology has disrupted how we (as users) envision technology's role in our everyday life. While originally mobile devices were perceived as personal information management tools, and thus as *tools* in a traditional sense, today's mobiles have access to a vast amount of knowledge from which they can learn, and seemingly become a companion capable to contest a user's agency and autonomy.

There are some benefits of this ongoing shift of agency and capabilities towards mobiles or technology in general, such as technology becoming able to recognize harmful behavioral habits of users and assist users in reflecting on their habits and hopefully provide support in adopting positive habits. Be it to regularly taking a walk or drinking enough, behavior change bears great potential towards improving wellbeing. Most new year's intents, for example, will have been given up by the time you read this paper.

In the following, we summarize related work in human activity recognition, which is an essential part in recognizing human behavior, and describe different ML approaches with regard to their characteristics and application domains.

2.1 Human Activity Recognition

Over the last two decades, research in Human Activity Recognition (HAR) has been focusing on a wide range of applications, such as surveillance and security [29], ambient intelligence [23] (e.g. to assist older adults [31]), or health care [38]. In particular, in ubiquitous computing environments or smart home environments, Human Activity Recognition is a key feature, for example, to monitor daily activities of users or provide assistance [39].

The rapid technical development of mobile devices and wearables, such as smartphones and smartwatches, has further expanded the possibilities for HAR. Mobile devices are equipped with a plethora of sensors and are worn or carried around all day. Thus, many activities of users can potentially be recognized. Consequently, a lot of research that investigated methods and applications [37] for HAR has emerged, in particular, research employing inertial sensors of smartphones [19, 28].

In more detail, HAR is used to automatically recognize a person's activities from a stream of sensor data, for example to pro-actively provide assistance, log daily routines, or to initiate necessary procedures (such as calling an ambulance or neighbors in case a person has fallen [5]). This makes them an important entity among today's e-health topics, be it detecting stereotyped movements in children with developmental disabilities [17] or automatic monitoring of rehabilitation processes [30] or using smart cups to track the behavior of residents of an inpatient nursing care facility [41]. In comparison to smartphones, smartwatches have a decisive advantage, which makes them particularly suitable for HAR. They are body-mounted and therefore always at the same place (i.e. constantly attached to the user's arm throughout a day). The human arm is actively involved in most of daily activities, whereas movements of the body can be smaller and may only reveal few activities.

Smartphones usually detect only movements related to the whole body due to their typical placement in the pocket. Therefore, the number of identifiable activities with smartphones is limited. Examples from the literature include walking, running, jogging, standing, sitting, walking up/down stairs, or using an elevator [11, 19, 20]. In contrast, smartwatches or wrist-worn wearables have the potential to detect more activities than with smartphones, such as drinking, smoking, typing on the keyboard, or eating with a knife and fork. Thus, new application areas can be addressed, such as food/drink reminders and related habit awareness applications (e.g. [22, 28]). The fact that smartwatches record the subtleties of each individual's arm movements in turn allows ML algorithms to generate personalized models for activity recognition. Personalized models usually result in higher precision of recognition algorithms and require less amount of sample data than user-independent models.

With the rapid development of smartwatch technology, HAR on smartwatches is an ascending topic [4, 27]. In contrast to previous work that utilizes smartwatches, the work at hand combines online learning and interactive ML to continuously improve activity recognition models. Moreover, the complete learning process is done solely on the smartwatch without access to any online resources or requiring network connectivity.

2.2 Machine Learning

2.2.1 Interactive and Cooperative Machine Learning. Another very active field of research in Human Activity Recognition addresses interactive machine learning (iML) on mobile devices [19, 27]. Interactive ML distinguishes from classical ML by directly involving users in the ML process. Training of models is part of the deployed product (continuously improving models) and not just applied during the development process (static and previously trained models). For this reason, iML bears great potential for applications that have to continuously adapt to a user.

Ware et al. [36] describe an approach in which the user interacts with an ML system by selecting individual attributes for the creation of a decision tree. In their case, the user requires a certain amount of expertise in ML. In contrast, we aim to involve non-expert users in the ML process to exploit the fact that the user can judge for his/her activities best. Fails et al. [9] showed a similar approach for design tools using perceptual interfaces.

The work of Shahmohammadi et al. [27] evaluated iML based on smartwatch sensor data for Human Activity Recognition and found that only few training samples are required to achieve high recognition accuracy. They demonstrated that personalized models from iML performed significantly better than classic learning approaches. In comparison to their work, we make use of online learning approaches which require less computing resources and thus enable us to run the whole application solely on the smartwatch without the need to stream data to more powerful devices for further processing. By this means, our system increases mobility and makes the application independent of external dependencies or permanent reliable and stable network connections. We also rely on naturally recorded annotated sensor data instead of data based on instructions given to users to perform specific actions.

Similar to iML, *Cooperative Machine Learning* (cML) aims to leverage the capabilities of human and machine to solve an ML problem. Here, the focus lies on the effort required for labeling recorded data. Not the deployed model is handed over to the user for modification, but the human annotators are supported by the machine to speed up their work [40]. To implement such a cML

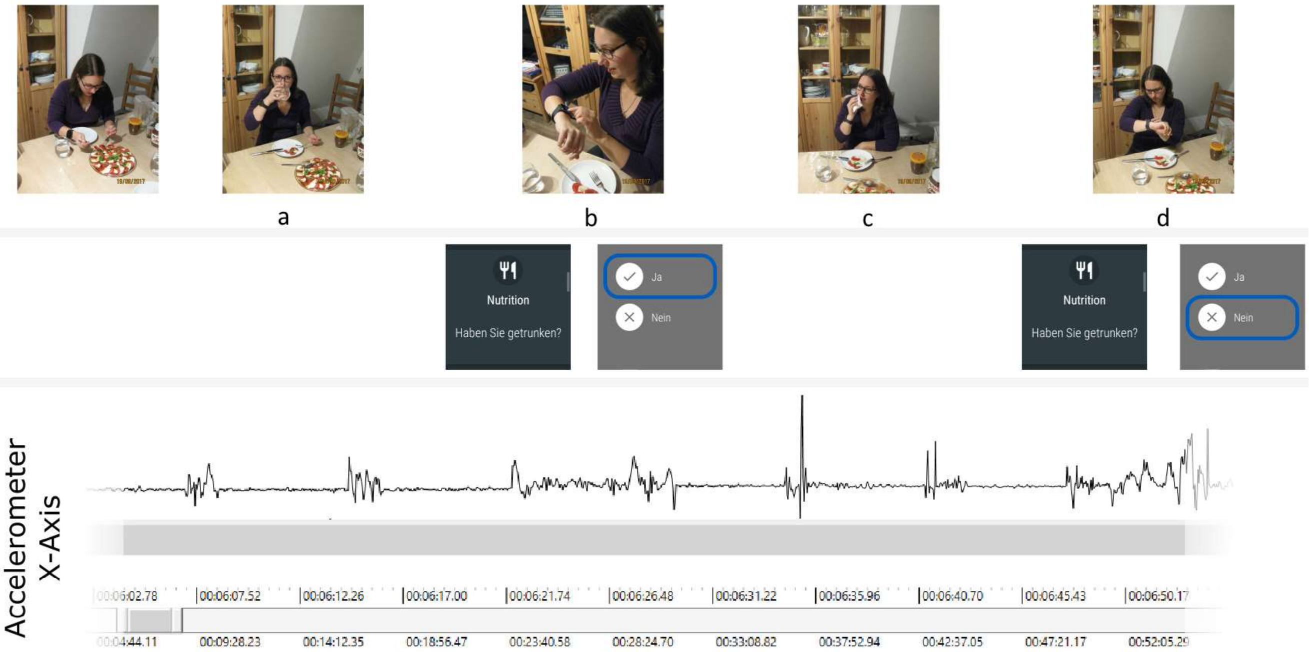

Figure 1: An exemplary health application scenario presenting the interaction and cooperation between a user and the DrinkWatch application. The second row provides screenshots of the DrinkWatch application and the third row presents raw accelerometer data of one movement axis as exemplary sensor data, which are used to recognize the drink activity.

approach the tight integration between an annotation software and the respective ML platform is crucial. In our previous work, this integration is provided by the NOVA annotation tool which has been developed by Baur et al. [3]. This tool makes use of the Social Signal Interpretation framework (SSI) [35] as an ML framework to speed up the annotation of social signals [34]. More specifically, the combination of both platforms enables users to train a new model based on only a few annotations from a recorded session and subsequently use this model to automatically predict annotations for the remaining part of the session. The annotator then only needs to correct the system's prediction which is potentially much less time-consuming than annotating the data from scratch. In an initial simulated study, we demonstrated a reduction of the labeling effort by 40 %. Following a similar approach, we make use of both tools in order to simulate the cooperative ML process of our mobile implementation as described in Section 4.2. They form the basis of the mobile implementation of Section 3.3.

2.2.2 Active Learning. Miu et al. [19] presented an Online Active Learning framework and studied how to collect user-provided annotations to bootstrap personalized activity models. They demonstrated that generating personalized Human Activity Recognition models can be achieved on-the-fly and does not require expert supervision or retrospective annotation of sample data. While Miu et al. made use of a smartphone app to query the user, our work queries annotations through a smartwatch interface. A smartwatch app has the benefit that queries on a smartwatch can be handled more comfortably and quickly since smartwatches don't require users to get it out of the pocket first.

Active Learning has been investigated for different models and classification types (e.g. Support Vector Machines [32]) as well as different types of query strategies. An overview is given by Settels et al. [25]. The most widely used approach and therefore selected for our applied entry point in Section 3.3.5 is Uncertainty Sampling [18] or Query on Uncertainty. In these approaches, a labeling system picks up samples for which the target class cannot be determined with a high certainty. This way the system is not locked into just learning from data that it already handles well. According to Lewis et al. [18], this approach performs better than relevance sampling which picks high confidence samples for relabeling.

Also common is the approach called Query on Committee [26], where multiple models, that have a strongly different way of operating, are grouped into a committee. Those samples are chosen for labeling by an oracle, where the individual models disagree most. Other query methods are focused on error reduction [25]. They either directly try to maximize the expected error reduction with the selected sample or they look at the expected model change that is expected from all possible labels of the sample.

To the best of our knowledge, only few recent works on HAR have investigated iML based on a smartwatch [27, 28] or online learning with a smartwatch [19], but no one has attempted to combine both approaches to realize interactive online machine learning solely on a smartwatch independent of external computing resources. We designed and built a smartwatch application prototype that implements this combination of approaches and present insights from a technical evaluation.

3 DRINKWATCH PROTOTYPE

Across all the state-of-the-art and off-the-shelf mobile devices, smartwatches seem most suitable in providing least intrusive and immediate feedback in mobile settings, and thus, allowing users to reflect on their immediate activities and contextual habits. While their form factor and small size is indeed an advantage when considering their integration in everyday situations, it is also often challenging to design and to develop interactive applications for smartwatches. For example, smartwatches provide only a very small-sized screen which limits the amount of information that can be presented to users. This limitation is, however, not relevant for the intended use case of our system as we mainly make use of the smartwatch's movements for hand activity logging. Our system only occasionally shows notifications to users and asks them for feedback related to activities. The prototype system further aims to reduce the complexity and amount of interaction (required to recognize and log drink activities) through automation.

DrinkWatch aims at recognizing drink activities (by means of inertial sensor data of the smartwatch) and tracks each drink activity for later analysis (see Figure 1). If DrinkWatch senses "interesting data", which potentially represent a drink activity worth learning from (Figure 1a), the smartwatch queries the user for assigning a label to the recorded sample data (Figure 1b). Thereby, the user is actively involved in the ML process and may choose to adapt the drink activity model or not. Consequently, not only drinking, but also activities, such as blowing one's nose or wiping one's mouth (Figure 1c) may lead to a query to the user (Figure 1d).

DrinkWatch serves three main functions. First, it offers a graphical *user interface* for querying the user for annotations and for reviewing recognized/logged activities.

Second, DrinkWatch continuously collects data samples from the watch's accelerometer and other potential data sources. In our prototype, we included a smartscale which is outlined in Section 3.2.2. This data collection, our *corpus* (Section 3.2), serves as the basis for a *warmstart model* in our ongoing cML process. For the purpose of later evaluations, all collected data samples are also locally logged on the smartwatch. However, this is not required for the online learning approach since the learning process requires only the latest annotated sample, see Section 3.3.4.

Third, Drinkwatch integrates an *ML logic*, which runs as a service on the smartwatch. While most of the logic, such as the online learning classifier, are implemented in the C++ programming language, part of the logic is embedded in a thin Java layer connecting the ML logic with the Android system (e.g. user interface) via JNI.

In the following, we describe each of the three parts of the DrinkWatch, including the implementation of the ML logic (see Section 3.3) in detail.

3.1 User Interface

DrinkWatch is implemented as a stand-alone application that runs on the smartwatch Asus ZenWatch 2, which is using the mobile operating system Android Wear 2 (Figure 2). Beneath the up-to-date OS, it can be charged and programmed fast using a USB connection, which is handy for development and experiments. There are hardware solutions with a wider range of sensors or fitted input hardware, such as a bezel, that might be more attractive for long-term use. We designed a minimal user interface on the watch (see Figure 2) to handle queries to the user and to start and stop the learning pipeline. Thus, users have control over when and whether to provide labels. The simple interface allows non-expert users to easily provide feedback on the go. Drink activities that lie within the desired confidence range of the iML model trigger a request/notification. Notifications are given by playing the standard notification sound of the watch and displaying a text ("Have you been drinking?" instead of "Waiting on Event"). In our current prototype implementation, we had to turn off the vibration function of the watch as it influenced its accelerometer sensor. This issue will be solved in a next iteration by disabling sensor reading while a vibration is being executed by the watch.

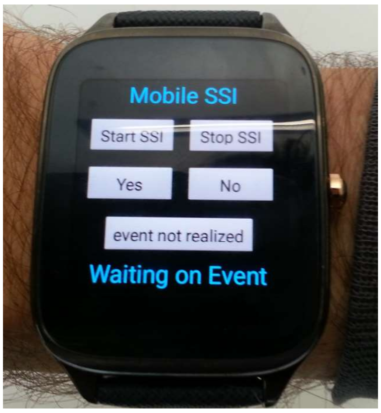

Figure 2: Cooperative Learning Interface on the smartwatch. The first two buttons enable the user to start or stop the recognition pipeline. Whenever a drink activity is detected, the user can inform the system whether the recognition was correct ("Yes") or incorrect ("No"). Additionally, with the last button, the user is able to indicate whether a drink activity was not detected.

3.2 Corpus for the Warmstart Model

3.2.1 Recording setup. In contrast to many other studies on activity recognition, we do not ask people to perform specific actions, but rather record sample data in everyday situations to label them afterwards based on a ground truth. Our recording setup was slightly different from session to session. Recording of acceleration data from the wrist was always performed using an Asus Zenwatch 2. In addition, the setup also included a camera to record video of the user when possible. The number of users per session varied from three to one, while 22 sessions (out of 25) had only a single user (see Figure 3). Every user was asked to wear the watch on their preferred hand. All recordings, except for five sessions, contained smartscale data, which can be used by our iML approach to speed up the annotation process.

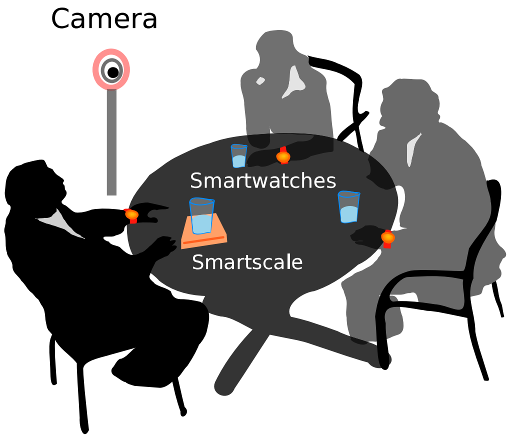

Figure 3: Recording setup with up to three people wearing smartwatches to record labeled accelerometer data for the initial classification model. The weight of one person's drinking vessel acquired by a smartscale and video data were additionally recorded to be able to annotate drink activities afterwards.

3.2.2 Smartscale. The smartscale prototype [24] in our system (see Figure 4) continuously broadcasts weight data of vessels placed on it via Bluetooth 4.0 to every receiver that is nearby. In our case, the smartwatch received and recorded the data whenever the watch was in reach of the smartscale.

Figure 4: A glass of apple juice standing on the smartscale.

Figure 5 exemplary shows recorded data of the smartscale. The graph resulted from drinking from two 0.5 l PET bottles (one by one). After each drink activity the bottle was placed on the smartscale. When the first bottle was empty it was replaced by a full one. The plot shows that the first bottle was not completely full and has not been placed on the sensor after being empty.

Whenever someone wants to drink out of a vessel placed on the scale, he or she usually first takes the vessel from the smartscale (weight is 0 g), drinks out of the vessel, and places the vessel back on the smartscale. The weight is now lower than before. The mass can increase if additional fluid is filled into a vessel or another vessel is being used which is heavier and/or contains more fluid.

In comparison to accelerometer data, the weight data of the smartscale is easier to interpret so that an annotator can quickly detect a drink activity, but also enables automated annotations. The video data can be used to validate the labeled time segments but does not have to be completely watched.

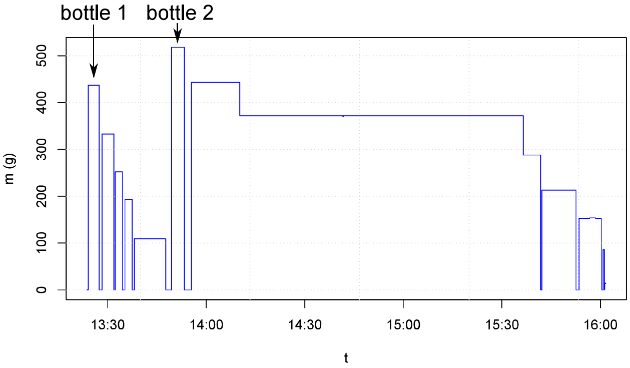

Figure 5: Weight data of the smartscale. Two filled 0.5 ml PET bottles have been drunken during this session. Whenever the drinking vessel is lifted the weight is 0 g (short lifting is omitted). After drinking the weight is reduced.

3.2.3 Dataset. We recorded 25 sessions, which consist of 16 hours and 30 minutes of every day activities containing 5117 samples of drink activities and 26288 samples of non-drink activities. One sample consists of a 1 second frame step together with 7 seconds of overlapping preceding data. A typical snippet of a drink activity is shown in Figure 7. Such an activity is characterized by three phases: picking up, bringing the vessel to the mouth and back as well as finally putting the vessel down. We employed random under-sampling to balance both classes in the training process. Acceleration data were recorded with 25 samples per second using the accelerometer sensor of an Asus ZenWatch 2. As ground truth we synchronously recorded video and smartscale data. An annotation session containing all data can be found in Figure 9. Furthermore, the Android system provides a so-called *linear acceleration sensor*, which represents the raw acceleration sensor exempt from the earth gravitation influence. Our prototype makes use of this linear acceleration sensor as it provides better performance for HAR [27]. These data were used to simulate a cML process and to gain a warmstart model for further iML, see Section 4. Thus, the data set is an important input for the ML module. The ML module is described in the following.

3.3 Implementation of the ML Module

We employ activity recognition to reduce manual logging effort that is required by the user when using a notebook or a conventional logging app. To this end, we continuously track the user's wrist activities in order to detect specific time windows (frames) that may be interpreted as an indicator of drinking. In case of high confidence, a drink event is automatically registered by a higher level app, e.g. a nutrition logging app. In case of low confidence, the system has to decide whether to ask the user for confirmation

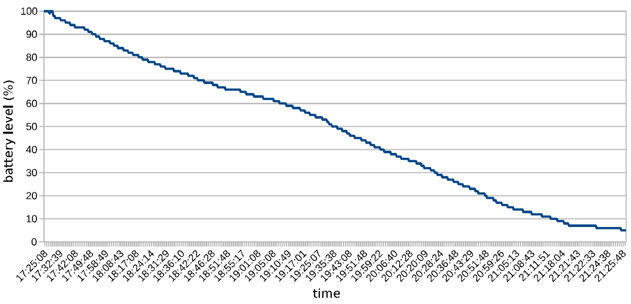

Figure 6: Battery level of Asus ZenWatch 2 running the mobileSSI iML pipeline

or not. We consider information gain as well as the user's situation, as discussed by Amershi et al. [1]. For example, the user should not be disturbed if the expected information gain is very low.

The maximum runtime of the system without WiFi is about four hours, as can be seen in the graph of Figure 6. In case of low battery (2 %), our prototype app stops the ML pipeline in order to properly finish the session. From the two days maximum battery life under optimized circumstances, this means a strong reduction.

Our prototype relies on mobileSSI [10]. It is open source and available on Github[1]. While mobileSSI already has ML capabilities for a range of classifiers, implementation follows a classic non-interactive approach. Our extensions include online learning capabilities (see Section 3.3.4) that enable the user to interact with the model using a simple user interface while the model actively (see Section 3.3.5) queries the user. Our prototype also shares parts with a classic ML pipeline, such as data collection and feature extraction, which are described in the following.

A brief overview of the pipeline and application concept is given in Figure 8. The red arrows mark continuous streams with a fixed sample rate kept in sync by the SSI framework. Blue dotted arrows mark events that are sporadic, but contain a time stamp and duration. Gray components are either future work, the user moderation and context component, or not described in this paper, namely the integration with the nutrition logging app.

3.3.1 Frame Size. In order to continuously process data, segmentation of the data has to be addressed. We selected a fixed window size of 1 second together with an overlap of preceding 7 seconds. This allows us to capture the whole event in most cases while having a reactive system, giving quick feedback. Given our chosen sample rate of 25 Hz we gain 200 raw data points in three dimensions, as our accelerometer has three axes.

3.3.2 Feature Selection. Accelerometer data are widely used in Human Activity Recognition and a lot of features have been experimented with. Features are needed to simplify the classification process in contrast to end-to-end learning. Our feature set is based on related work. In particular, we selected a range of features that are known to work well on acceleration data [2, 8, 14, 16, 21] and have been used for the recognition of drink activities.

On each axis/dimension, the following features were calculated:

- Mean
- Std. deviation
- Variance
- Energy
- Interquartile range (IQR)
- Mean absolute deviation (MAD)
- Root mean square (RMS)
- Min
- Max

Additionally following features are generalizing over all axes:

- Correlation between XY, XZ, YZ axis
- Mean, Std. deviation, Min, Max, IQR on length of per sample vector over all axes (magnitude)

This results in overall 35 features calculated on the previously described 1 + 7 seconds containing 200 samples.

3.3.3 Normalization. Normalization is scaling all features' data range to fit a certain range, in our case within 0 and 1. This is, for example, done by using the accelerometer's maximum output value that can be queried using the Android API. Compared to a classical approach in mobileSSI and many other implementations of classical ML, the responsibility of data normalization is moved from the training process, iterating over all samples in the data set, to the feature calculation, on the current chunk of data. This is necessary because with low initial sample count determining the minimum and maximum on already known data might not be representative for future data. There are alternatives to feature based normalization, such as adaptive scaling. While normalization is not strictly necessary for Gaussian Naive Bayes it is recommended to keep features with higher values from dominating features with small values. Our pipeline provides a feature vector of dimension 35 that is fed into the following online classifier component every second.

3.3.4 Online Learner. Classification of the current data frame is handled by our pipeline, as it would be the case in a classic ML pipeline. Our main objective is to continuously improve learned models for fluid intake based on tracked data and user input. Online learning enables us to learn a new model from scratch in the deployed application. Furthermore, the model can be improved at the moment the user provides new labeled data and the next input can be analyzed with the improved model without the need to restart or stop the application. To speed the process up, a classically trained model is used as a starting point for further incremental training. This procedure is called warm-start.

We chose Naive Bayes which can be easily adapted for online learning (see e.g. the implementation used in MOA [6]). The online learning variant of Naive Bayes incrementally calculates mean, variance and standard derivation and additionally stores the sample count to be able to adjust with new data proportionally. The algorithm is described in detail by Knuth [15] on page 115. The calculations are executed per feature and class, thus our model consists of 210 float values and a sample count. As Naive Bayes classification results into confidence values, it enables us to query the user based on the level of uncertainty. Furthermore, it is fast in training and execution. This makes Naive Bayes a good option for restricted platforms, such as smartwatches. Moreover, it offers an

[1] https://github.com/hcmlab/mobileSSI

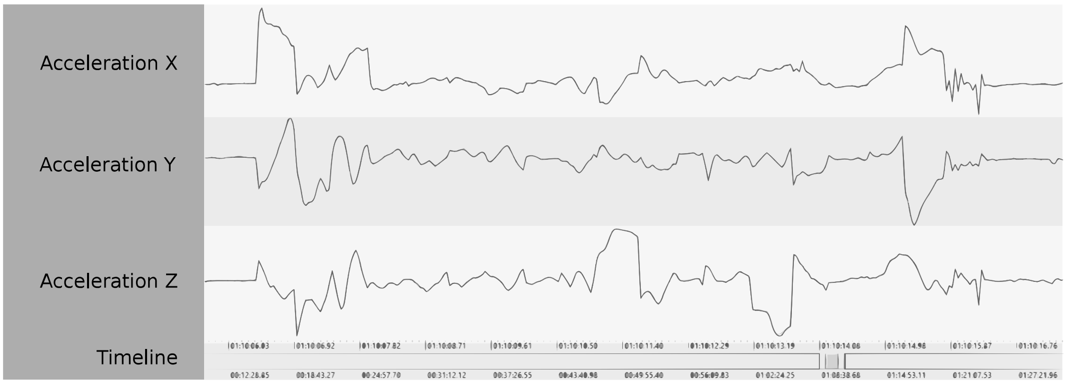

Figure 7: Three axis accelerometer data of a drink activity. The start and end of the signal describe the movement of the drinking vessel to and from the mouth. In the middle of the signal the rotation of the vessel by turning the wrist takes place.

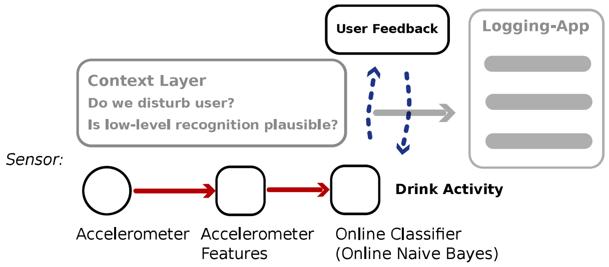

Figure 8: Overview over iML Pipeline and future system components.

advantage in data security, as no other data that can give an insight in user behavior or health related information are permanently saved to the watch.

At this point LibLinear is only integrated without online learning capabilities, but a solution exists according to Tsai et al. [33]. The future integration of LibLinear as additional online learning library depends on the result of our evaluation, see Section 4.

3.3.5 Active Learning. Our Active Learning implementation uses query on uncertainty for sample selection, see Section 2.2.2 for further background. We can specify the credibility range that triggers user requests, thus supports relevance sampling as well as uncertainty sampling. The option is part of an online classifier component shown in Figure 8. It manages the assembly of sample lists from user annotations and data streams as well as the training process of our online model. The model's predictions are also handled by the online classifier. Both, requests and predictions, are handled as events instead of streams with fixed sample rate.

4 EVALUATION AND RESULTS

Following system implementation and data collection, three steps of evaluation are presented in this section: the static evaluation of the fully annotated data set in Section 4.1, the evaluation of different learning strategies in Section 4.2, and the interactive run performed with end users in Section 4.3.

4.1 Evaluation of Static Models

To give an overview of our collected data and provide an impression of what accuracy fully trained models are able to achieve, Table 1 shows results of Naive Bayes and linear SVM (implementation: LibLinear) models trained on the full data set, evaluated on the fixed test set that is also used for the simulation of cooperative ML.

	Results of full Training	
	Naive Bayes	linear SVM
Drinking	81.4%	84.9%
Not drinking	71.6%	79.8%
Unweighted Average	**76.5%**	**82.3%**

Table 1: Results of training on all annotations contained in the training set, evaluated on the test set.

Our results are in line with other results on drink activity recognition found in literature. The linear SVM model shows a six percent points lead over Naive Bayes, which again is as expected. While there is a difference on the "drinking" class, it is larger on the "not drinking" class. As "not drinking" is by far larger and more complex, Naive Bayes meets its limitations in describing it.

4.2 Learning Strategy Simulation

As we aim to utilize the learning process within an end user application that is designed to continuously adapt to the specific activity patterns of the user, it makes sense to not only evaluate the complete model, but also the relative improvements of the classifier when increasing the amount of training data. To evaluate this continuous refinement of the classification system, we simulated the iterative training process by using the NOVA [3] toolkit.

First of all, we trained our base model on a small stack of eight annotations from one session. From there on we used this baseline classifier to predict the rest of the training data. Subsequently, we took the first label where the confidence is equal or greater than the lower end of a predefined confidence interval. In case the confidence value lies within the interval we queried the oracle to correct our

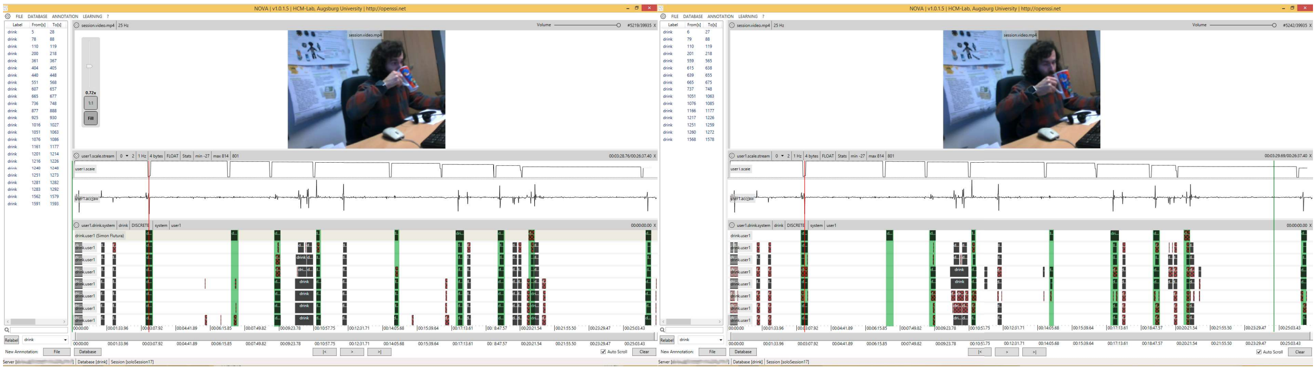

Figure 9: Cooperative Machine Leaning in NOVA: Predictions of LibLinear (left) and Naive Bayes (right) on one session. Video, smartscale and acceleration data are followed by annotations. The first line contains the hand labeled annotation and is followed by predictions of models with increased number of training data. Areas marked in green are drink activity.

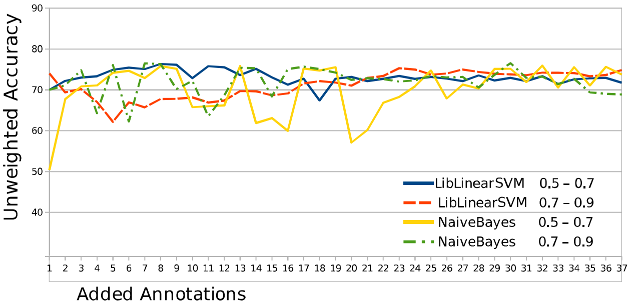

Figure 10: Training progression using different confidences and models

answer. The oracle is simulated by the full hand-labeled annotation. In case the confidence value of the prediction is higher than the upper limit of the interval we assume that the classification of the sample is correct, and forwarded to the logging application. Afterwards we add our newly annotated sample to the training data and retrain our classifier before repeating the same steps again. This process continues iteratively until all available data has been annotated. While in theory the classifier could learn from data with high classification confidence and improve without explicit user input, we stick to user (oracle) labeled data only because those are guaranteed to be true positive samples as long as the user gives correct feedback.

The study has been conducted by applying an uncertainty sampling strategy which utilizes a low confidence interval ranging from 0.5 to 0.7 as well as a relevance sampling strategy using a high confidence interval from 0.7 to 0.9, see Figure 10. While one would expect the unweighted average accuracy to increase steadily with the number of available training data our simulation results paint a different picture as shown in Figure 10. Naive Bayes is clearly more unstable than LibLinear's linear SVM. Obviously, it is less robust against variations across sessions and users as well as untypical drink activities, for example, those with long pauses while holding the vessel.

All models stabilize over the course of the simulation. By the time 30 additional labels are added to the base stock, the variations in accuracy narrow down to five percent points for Naive Bayes and three percent points for LibLinear, when adding new labels to the training process. While low confidences seem to be preferred by the LibLinear SVM model, queries based on high confidences seem to be the better choice for Naive Bayes. The progress of both models is best judged using predictions, as shown in Figure 9. One can see where the classifier triggers and with what confidence, as indicated by hatching and color. The first line contains the hand labeled annotation and is followed by predictions of models with increased number of training data. Areas marked in green are drink activities. Naive Bayes changes in accuracy, seen in Figure 10 manifest themselves as low confidence, red bars on the right.

4.3 Interactive Machine Learning Sessions

We recruited two users who used DrinkWatch for one hour to track their drink activities. For this experiment, we picked the high confidence range (0.7 to 0.9) as it promises an earlier stabilization for Naive Bayes. To create a reliable base model, we used at least 40 annotations.

One can judge the quality of the model by the appropriateness and frequency of queries. While both users had the impression that drink activities were accurately recognized in general (e.g. "Five out of six" stated by one of our two users), there were many wrong positives due to the unbalanced nature of both classes. The unfiltered requests were described as annoying by the users and made the system unusable.

The behavior of the system appeared transparent to users. They noted that moving a vessel containing fluid, slow and steady was a key trigger for recognizing drink activities. It was also easy for them to mimic activities triggering the model, describing properties of the movement that lead to requests.

4.4 Discussion

In the beginning we have motivated the need for mobile interactive and cooperative ML approaches by highlighting shortcomings of traditional ML approaches, considering (i) difficulties in getting authentic data of every day living, and (ii) a deficit of transparency and user control. We have also argued that interactively integrating users into the ML process would have the potential to address both issues, allowing users to label their own activities, to gain some understanding of and control over machine functionalities, and to ultimately peek behind the curtain of automation and to leave users with a feeling of competence and self-efficacy.

Since mobile cooperative learning is a novel research area with many conceptually and technically open issues, we have exposed ourselves into the process of developing the DrinkWatch application and its integration with smart data sources, such as the smartscale. Our intention and aim was to become able to infer limitations and potentials of future mobile cooperative ML application. After developing the core functionalities of the DrinkWatch application, we spent a time period of six months iterating the application based on multiple tests, including a longer period of time testing the application with ourselves and short episodes collecting insights from letting colleagues and friends try the application. Consequently, our main contribution is DrinkWatch as a hardware and software solution, demonstrating technical feasibility, providing detailed information for scientific reproduction, and last but not least initial user impressions and insights considering how we expect users will experience DrinkWatch. We also hope to have provided fellow researchers a methodological scheme, which can be reused and adapted for developing and evaluating other interactive and cooperative ML applications.

By building and testing DrinkWatch, we have learned that interactive cooperative machine learning is already feasible on today's state of the art smartwatches. We believe the feedback provided by the model (i.e. the machine intelligence) as a direct consequence to a drink activity is intuitively graspable by users even when feedback is provided through simple audio notifications. Based on the model performance in recognizing drink activities, we believe (as it is typical with many ML based models) that it can be adopted easily to recognize other hand-based activities.

LibLinear's linear SVM does not only show higher accuracy compared to Naive Bayes, but also a smoother learning curve. Since both models have opposing tendencies when it comes to confidence intervals, fusing both models in a Query on Committee [26] implementation, seems promising. The committee might also be accompanied by static models, such as the warmstart model or save points that can be created by the user as well.

As also described in the interactive ML paradigm [1], we believe that queries should be forwarded to the user with care since wrong positives cause frustration and users tend to describe the experience associated with wrong positives as "annoying". When it comes to adaptability to new health-related hand activities, we presented several observations that can be used as reference points. The minimal strength of the Naive Bayes warmstart model for our problem can be set at circa 40 overall annotations, this equals about eight hours of recording in our case while the linear SVM stabilizes at about 30 overall annotations or six hours of recording.

We introduced the smartscale as an option to integrate data from other data sources, in the hope to improve the (initial) quality of the model. The use of a smartscale reduced the annotation effort drastically and we came to understand that it is a suitable physical object to facilitate logging of fluid intake as well as to support annotation and online learning on smartwatches in a stationary setting.

4.4.1 *Limitations and Future Directions.*

We have focused on single user scenarios, investigating the question of how DrinkWatch can provide users with "power" over their ML applications. However, in many health applications, the user is not the only person who should interact with the ML application. Furthermore, there are places and contexts in which desired and available autonomy of users may vary. For example, in hospitals or retirement homes, the environment may impose autonomy constraints. A patient's behavior may need to be observed more intensively, and observation and interaction responsibility in hospitals may be distributed among patients and others (such as nurses and medical doctors), resulting in a nurse that helps observe patient behavior and label data. While on the go the integration of additional data sources comes with challenges (due to privacy issues), in (smart) homes the performance of ML applications may be improved by integrating additional knowledge sources, such as smartscales.

There is no doubt that ML-based automation in health monitoring applications will increase, resulting in a larger number of everyday user interactions with "smart" applications, which observe user behavior and query user feedback. One of the future challenges will be to integrate multiple applications and systems and to regulate not only the pure amount, but also the nature of user-system interactions. We therefore plan to integrate a context-aware mediator layer, filtering requests and learning when is a good point in time to bother the user with queries. As the query frequency will then be changed, the unfiltered query frequency can be transported to the user by other modalities, such as volume, length or pattern of the notifications' sound or vibration. In our immediate future work, we aim to study when to trigger user feedback on smartwatches in order to explore how modality and timing of notifications interfere with user experience and willingness for cooperation. Using the refined prototype, DrinkWatch can be employed "In the Wild" with a larger group of users for deeper insights.

5 CONCLUSIONS

In this paper we presented the adoption of cooperative machine learning on a smartwatch. We evaluated a prototype via simulation and initial interactive sessions. Our approach shows that today's smartwatches are capable of executing interactive machine learning for activities of daily life. The model generates sufficient feedback to let the user judge its state by means of query frequency and time. Smartwatches enable the user to intuitively mimic recognized behavior and explore the model's capabilities. From here a variety of options are open for future research, be it refining the machine learning process, integration into existing logging applications, or studying and refining how the system interacts with the user.

REFERENCES

[1] Saleema Amershi, Maya Cakmak, W. Bradley Knox, and Todd Kulesza. 2014. Power to the People: The Role of Humans in Interactive Machine Learning. *AI Magazine* (December 2014).

[2] Ling Bao and Stephen S. Intille. 2004. *Activity Recognition from User-Annotated Acceleration Data.* Springer Berlin Heidelberg, Berlin, Heidelberg, 1–17. https://doi.org/10.1007/978-3-540-24646-6_1

[3] Tobias Baur, Gregor Mehlmann, Ionut Damian, Florian Lingenfelser, Johannes Wagner, Birgit Lugrin, Elisabeth André, and Patrick Gebhard. 2015. Context-Aware Automated Analysis and Annotation of Social Human-Agent Interactions. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5, 2 (2015), 11.

[4] Sourav Bhattacharya and Nicholas D. Lane. 2016. From smart to deep: Robust activity recognition on smartwatches using deep learning. In *2016 IEEE International Conference on Pervasive Computing and Communication Workshops, PerCom Workshops 2016, Sydney, Australia, March 14-18, 2016*. 1–6. https://doi.org/10.1109/PERCOMW.2016.7457169

[5] Noemi Biancone, Chiara Bicchielli, Fernando Ferri, and Patrizia Grifoni. 2016. Falls Detection and Assessment. In *Proceedings of the 8th International Conference on Management of Digital EcoSystems*. ACM, New York, NY, USA, 204–207. https://doi.org/10.1145/3012071.3012088

[6] Albert Bifet, Geoff Holmes, Richard Kirkby, and Bernhard Pfahringer. 2010. MOA: Massive Online Analysis. *J. Mach. Learn. Res.* 11 (Aug. 2010), 1601–1604. http://dl.acm.org/citation.cfm?id=1756006.1859903

[7] Rafael A Calvo and Dorian Peters. 2014. *Positive Computing: Technology for Wellbeing and Human Potential*. MIT Press.

[8] Yen-Ping Chen, Jhun-Ying Yang, Shun-Nan Liou, Gwo-Yun Lee, and Jeen-Shing Wang. 2008. Online classifier construction algorithm for human activity detection using a tri-axial accelerometer. *Appl. Math. Comput.* 205 (2008), 849–860.

[9] Jerry Alan Fails and Dan R. Olsen, Jr. 2003. Interactive Machine Learning. In *Proceedings of the 8th International Conference on Intelligent User Interfaces (IUI '03)*. ACM, New York, NY, USA, 39–45. https://doi.org/10.1145/604045.604056

[10] Simon Flutura, Johannes Wagner, Florian Lingenfelser, Andreas Seiderer, and Elisabeth André. 2016. MobileSSI: Asynchronous Fusion for Social Signal Interpretation in the Wild. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction (ICMI 2016)*. ACM, New York, NY, USA, 266–273. https://doi.org/10.1145/2993148.2993164

[11] Arindam Ghosh and Giuseppe Riccardi. 2014. Recognizing Human Activities from Smartphone Sensor Signals. In *Proceedings of the 22Nd ACM International Conference on Multimedia*. ACM, New York, NY, USA, 865–868. https://doi.org/10.1145/2647868.2655034

[12] Tad Hirsch, Kritzia Merced, Shrikanth Narayanan, Zac E. Imel, and David C. Atkins. 2017. Designing Contestability: Interaction Design, Machine Learning, and Mental Health. In *Proceedings of the 2017 Conference on Designing Interactive Systems*. ACM, New York, NY, USA, 95–99. https://doi.org/10.1145/3064663.3064703

[13] Eric Horvitz. 1999. Principles of Mixed-initiative User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. ACM, New York, NY, USA, 159–166. https://doi.org/10.1145/302979.303030

[14] Tâm Huynh and Bernt Schiele. 2005. Analyzing Features for Activity Recognition. In *Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-aware Services: Usages and Technologies*. ACM, New York, NY, USA, 159–163. https://doi.org/10.1145/1107548.1107591

[15] Donald E. Knuth. 1985. *The Art of Computer Programming, Volume 2 (2nd Ed.): Seminumerical Algorithms*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[16] O. D. Lara and M. A. Labrador. 2013. A Survey on Human Activity Recognition using Wearable Sensors. *IEEE Communications Surveys Tutorials* 15, 3 (Third 2013), 1192–1209. https://doi.org/10.1109/SURV.2012.110112.00192

[17] YeongJu Lee and Minseok Song. 2017. Using a Smartwatch to Detect Stereotyped Movements in Children With Developmental Disabilities. *IEEE Access* 5 (2017), 5506–5514.

[18] 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*. Springer-Verlag New York, Inc., New York, NY, USA, 3–12. http://dl.acm.org/citation.cfm?id=188490.188495

[19] 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

[20] Yunyoung Nam, Seungmin Rho, and Chulung Lee. 2013. Physical Activity Recognition Using Multiple Sensors Embedded in a Wearable Device. *ACM Trans. Embed. Comput. Syst.* 12, 2, Article 26 (Feb. 2013), 14 pages. https://doi.org/10.1145/2423636.2423644

[21] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L. Littman. 2005. Activity Recognition from Accelerometer Data. In *Proceedings of the 17th Conference on Innovative Applications of Artificial Intelligence - Volume 3*. AAAI Press, 1541–1546. http://dl.acm.org/citation.cfm?id=1620092.1620107

[22] Reza Rawassizadeh, Blaine A. Price, and Marian Petre. 2014. Wearables: Has the Age of Smartwatches Finally Arrived? *Commun. ACM* 58, 1 (Dec. 2014), 45–47. https://doi.org/10.1145/2629633

[23] Natalia Díaz Rodríguez, M. P. Cuéllar, Johan Lilius, and Miguel Delgado Calvo-Flores. 2014. A Survey on Ontologies for Human Behavior Recognition. *ACM Comput. Surv.* 46, 4, Article 43 (March 2014), 33 pages. https://doi.org/10.1145/2523819

[24] Andreas Seiderer, Simon Flutura, and Elisabeth André. 2017. Development of a Mobile Multi-device Nutrition Logger. In *Proceedings of the 2Nd ACM SIGCHI International Workshop on Multisensory Approaches to Human-Food Interaction*. ACM, New York, NY, USA, 5–12. https://doi.org/10.1145/3141788.3141790

[25] Burr Settles. 2010. Active learning literature survey. *Computer Sciences Technical Report* 1648 (2010).

[26] H. S. Seung, M. Opper, and H. Sompolinsky. 1992. Query by Committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. ACM, New York, NY, USA, 287–294. https://doi.org/10.1145/130385.130417

[27] F. Shahmohammadi, A. Hosseini, C. E. King, and M. Sarrafzadeh. 2017. Smartwatch Based Activity Recognition Using Active Learning. In *2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*. 321–329. https://doi.org/10.1109/CHASE.2017.115

[28] Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten, and Paul J. M. Havinga. 2016. Complex Human Activity Recognition Using Smartphone and Wrist-Worn Motion Sensors. *Sensors* 16, 4 (2016). http://www.mdpi.com/1424-8220/16/4/426

[29] C. Stauffer and W. E. L. Grimson. 2000. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 8 (Aug 2000), 747–757. https://doi.org/10.1109/34.868677

[30] Christina Strohrmann, Rob Labruyère, Corinna N. Gerber, Hubertus J. van Hedel, Bert Arnrich, and Gerhard Tröster. 2013. Monitoring motor capacity changes of children during rehabilitation using body-worn sensors. *Journal of NeuroEngineering and Rehabilitation* 10, 1 (30 Jul 2013), 83.

[31] Kristin Taraldsen, Sebastien F.M. Chastin, Ingrid I. Riphagen, Beatrix Vereijken, and Jorunn L. Helbostad. 2017. Physical activity monitoring by use of accelerometer-based body-worn sensors in older adults: A systematic literature review of current knowledge and applications. *Maturitas* 71, 19 (2017). https://doi.org/10.1016/j.maturitas.2011.11.003

[32] Simon Tong and Daphne Koller. 2002. Support Vector Machine Active Learning with Applications to Text Classification. *J. Mach. Learn. Res.* 2 (March 2002), 45–66. https://doi.org/10.1162/153244302760185243

[33] Cheng-Hao Tsai, Chieh-Yen Lin, and Chih-Jen Lin. 2014. Incremental and Decremental Training for Linear Classification. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, 343–352. https://doi.org/10.1145/2623330.2623661

[34] J. Wagner, T. Baur, Y. Zhang, M. F. Valstar, B. Schuller, and E. André. 2018. Applying Cooperative Machine Learning to Speed Up the Annotation of Social Signals in Large Multi-modal Corpora. *ArXiv e-prints* (Feb. 2018). arXiv:cs.HC/1802.02565

[35] Johannes Wagner, Florian Lingenfelser, Tobias Baur, Ionut Damian, Felix Kistler, and Elisabeth André. 2013. The social signal interpretation (SSI) framework: multimodal signal processing and recognition in real-time. In *ACM Multimedia Conference, MM '13, Barcelona, Spain, October 21-25, 2013*. 831–834.

[36] Malcolm Ware, Eibe Frank, Geoffrey Holmes, Mark Hall, and Ian H. Witten. 2002. Interactive Machine Learning: Letting Users Build Classifiers. *Int. J. Hum.-Comput. Stud.* 56, 3 (March 2002), 281–292. http://dl.acm.org/citation.cfm?id=514412.514417

[37] Che-Chang Yang and Yeh-Liang Hsu. 2010. A Review of Accelerometry-Based Wearable Motion Detectors for Physical Activity Monitoring. *Sensors* 10, 8 (2010), 7772–7788. https://doi.org/10.3390/s100807772

[38] Jun Yang. 2009. Toward Physical Activity Diary: Motion Recognition Using Simple Acceleration Features with Mobile Phones. In *Proceedings of the 1st International Workshop on Interactive Multimedia for Consumer Electronics*. ACM, New York, NY, USA, 1–10. https://doi.org/10.1145/1631040.1631042

[39] Mi Zhang and Alexander A. Sawchuk. 2012. USC-HAD: A Daily Activity Dataset for Ubiquitous Activity Recognition Using Wearable Sensors. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, New York, NY, USA, 1036–1043. https://doi.org/10.1145/2370216.2370438

[40] Zixing Zhang, Eduardo Coutinho, Jun Deng, and Björn Schuller. 2015. Cooperative Learning and Its Application to Emotion Recognition from Speech. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.* 23, 1 (Jan. 2015), 115–126. https://doi.org/10.1109/TASLP.2014.2375558

[41] C. Zimmermann, J. Zeilfelder, T. Bloecher, M. Diehl, S. Essig, and W. Stork. 2017. Evaluation of a smart drink monitoring device. In *2017 IEEE Sensors Applications Symposium (SAS)*. 1–5. https://doi.org/10.1109/SAS.2017.7894061

COMPETING INTERESTS

The authors have declared that no competing interests exist.