

A Human-centered Wearable Sensing Platform with Intelligent Automated Data Annotation Capabilities

Roger Solis

Texas A&M University
College Station, TX
roger.solis@tamu.edu

Arash Pakbin

Texas A&M University
College Station, TX
a.pakbin@tamu.edu

Ali Akbari

Texas A&M University
College Station, TX
aliakbari@tamu.edu

Bobak J. Mortazavi

Texas A&M University
College Station, TX
bobakm@tamu.edu

Roozbeh Jafari

Texas A&M University
College Station, TX
rjafari@tamu.edu

ABSTRACT

Wearable computers provide significant opportunities for sensing and data collection in user's natural environment (NE). However, they require both raw data and annotations to train their respective signal processing algorithms. Collecting these annotations is often burdensome for the users. Our proposed methodology leverages the notion of location from nearable sensors in Internet of Things (IoT) platforms and learns users' patterns of behavior without any prior knowledge. It also requests users for annotations and labels only when the algorithms are unable to automatically annotate the data. We validate our proposed approach in the context of diet monitoring, a significant application that often requires considerable user compliance. Our approach improves eating detection accuracy by 2.4% with requested annotations restricted to 20 per day.

CCS CONCEPTS

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

KEYWORDS

Intelligent annotations, nearables, environment-aware, dietary monitoring, wearables

ACM Reference Format:

Roger Solis, Arash Pakbin, Ali Akbari, Bobak J. Mortazavi, and Roozbeh Jafari. 2019. A Human-centered Wearable Sensing Platform with Intelligent Automated Data Annotation Capabilities. In *International Conference on Internet-of-Things Design and Implementation (IoTDI '19), April 15–18, 2019, Montreal, QC, Canada*. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3302505.3310087>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

IoTDI '19, April 15–18, 2019, Montreal, QC, Canada

© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6283-2/19/04...\$15.00
<https://doi.org/10.1145/3302505.3310087>

1 INTRODUCTION

Wearable devices are becoming ubiquitous. This has opened new opportunities for sensing and collecting data in natural environments (NE). Devices, such as smartwatches, track user behavior to help users better manage their life-style. For example, tracking food intake enables users to maintain a healthier diet. Extensive amounts of data such as sensor readings and their corresponding annotations are needed to build such tracking algorithms. In many such systems, users are to provide annotations. However, if not performed intelligently, annotating may become a burden to the user. Therefore, a system that collects such data from wearable devices while intelligently requesting annotations is needed. Such a system can ensure high compliance rates and achieve accurate NE tracking.

Manual data annotation approaches require burdensome interaction, impacting user adherence [20, 22]. Additionally, the annotation process is often performed in laboratory environments, adding unwanted bias because the labels may not realistically represent NE. As a result, algorithms trained on this data tend to perform poorly when released in NE [1]. In activity recognition, approaches have tried to simulate NE in laboratory [2, 4]. Semi-supervised approaches, such as self-learning [17] and co-training [6], use the probability returned by the classifier to guide their labeling process. When uncertainty levels are low, the label assigned by the classifier is considered correct. However, these approaches overfit and ignore samples for which the classifier's probability output was low even though the assigned label was correct. Automatic labeling approaches, such as active learning (AL) and experience sampling methods (ESM) that obtain labels directly from the user increase accuracy at higher rates [15, 18]; however, such approaches do not limit the number of interactions, impacting the adherence [22].

We propose a smartwatch-based platform that collects sensor data and intelligently asks for user annotations for minimum burden in NE through an AL approach. Our AL algorithm is based on classifier uncertainty, a maximum number of interactions allowed per day, true positive rates, and similarity with already seen samples. Through environment aware sensing we score the need for annotation and test our algorithm on the detection of eating moments for better diet monitoring.

The contributions of this work are as follows:

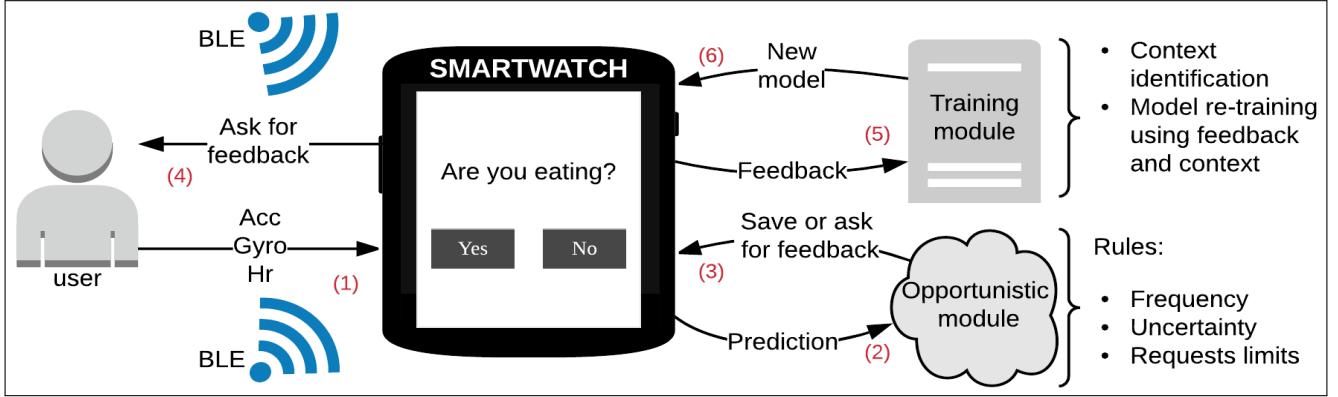


Figure 1: The flow of the platform is as follows: 1) data is collected through the smartwatch’s sensors, 2) a prediction is made on the watch, 3) the annotation is asked if the score dictates, 4) a user provides feedback, 5) the feedback is provided to the server for re-training, and 6) the new model is deployed to the watch.

- An algorithm for intelligent data annotation that reduces user annotation burden.
- Improvement of an event detection classifier through environmental information.
- Evaluation of our AL algorithm on diet monitoring.

2 RELATED WORK

Different approaches exist for data annotation. We discuss those that are automatic and most related to our work. We also mention current approaches to eating moment detection and how their solutions are cumbersome to users. Finally, we discuss pervasive platforms and how context can be leveraged in this type of systems.

Automatic annotation approaches aim to ask users for annotations only when needed. For example, some approaches ask only if the predicted label probability is above a threshold [17]. Then, the sample is added to the training set for retraining. However, this can add noise to the dataset when high confidence predictions are wrong. Other work such as co-training [6], En-Co-Training [13], and democratic co-learning [24] use multiple classifiers to predict a label to address this. While these techniques require little or no interaction with users, methods which interact with the user have better accuracy [18], but with lower adherence [9]. Thus, AL approaches ask for an annotation only when the uncertainty levels are below a threshold [19]. Most of these approaches do not limit the number of annotations requested per day [7]. AL based approaches have been underutilized in many current applications, diet monitoring included.

Several sensor-based approaches to eating moment detection have been proposed. For example, with the processing of audio signals researchers were able to identify family eating moments [5], which is not always applicable to meals and environments. Another approach involved the use of inertial measurement unit (IMU) data from a smartwatch, but data was labeled via video recording, which involved a large number of interactions with user [21]. In fact, authors mentioned that in a longitudinal trial users would have to carry a power bank, making this an ineffective solution. Finally, other approaches have designed their own sensors. Wearable

devices placed around the neck or close to the mouth that aim at detecting chewing events like EarBit [3] or a necklace [14] are an example. However, these devices should be investigated further for long-term usability and pervasiveness. Smartwatches on the other hand are more ubiquitous and more likely to be worn long-term, which provide an advantage for a variety of tracking paradigms like diet monitoring. However, the pervasiveness of such devices should be exploited in order to get algorithms that adapt to users.

Context-aware systems improve the performance of a given task by adapting to environments and have been shown to have high accuracy in activity recognition applications [23], where contexts and locations are known. Therefore, both labeling and eating detection systems need context-aware approaches to develop pervasive recognition systems. OmniTrack [16] is a platform that attempts to overcome pervasiveness challenges in self-tracking. However, the system does not adapt to user behavior, resulting in a constant need for tuning. A methodology is needed that is pervasive and personalized, ideally through context-aware analytics.

3 METHODS

Our platform is composed of two components: an Android-wear smartwatch able to continuously collect data for 15 hours through a custom-made application, and a server for external processing as illustrated in Fig. 1. In addition to the data collected from its embedded sensors, the smartwatch captures bluetooth low energy (BLE) addresses being broadcast as a mean to obtain environmental information related to location. While ideal location information could be obtained through GPS, this approach consumes a lot of power and does not work indoors [12]. In order to develop a system that can collect data in NE and assist diet monitoring we need two algorithmic components. First, there is a need for accurate detection of events (e.g. eating moments), second, intelligent annotations user’s data are crucial. In order to determine when and where to ask for annotations, models should be developed to detect events of interest accurately, for which a context-aware approach is developed through the BLE data.

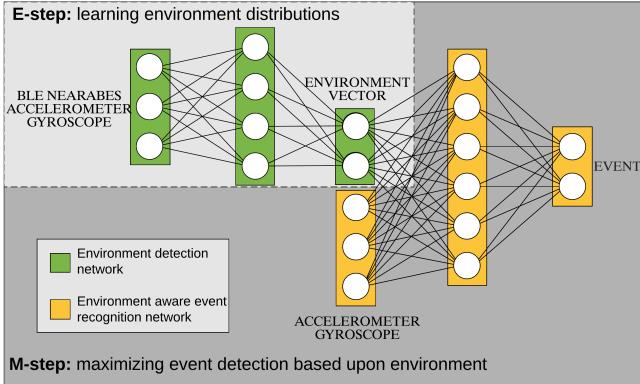


Figure 2: A deep-learning framework for environment and event detection trained EM.

3.1 Environment Utilization for More Accurate Models

Considering the benefits of context-aware approaches, available environmental information should be leveraged to reduce annotation burden on the user through increased event recognition accuracy. As discussed, one main criteria for prompting the user for feedback is the detection of uncertainty. Robust models which capture the complex nature of data collected in daily living decrease this uncertainty. For example, if a user is at work, her/his eating probability is lowered. Here, we use environment data to cluster user activities in order to build context-specific models. In order to obtain environmental data, we use nearable sensors, namely, addresses broadcasted by BLE devices.

3.1.1 Graph Based BLE-Location Mapping. Environment was identified by detecting consistent, repeated BLE addresses. Every day the system is exposed to a number of unique BLE addresses. In order to recognize a repetitive pattern in BLEs, we consider their co-occurrence. The idea behind forming co-occurrence is the assumption that in certain places where there are static devices, they tend to be observed with each other in all visits to that place. Therefore, the co-occurrence of addresses means that they likely belong to the same location. Subsequently, we store the BLE observation data in a graph structure. In this graph, every vertex represents a BLE address, and edges represent the number of co-occurrences at different locations. For example, if two BLE addresses have been seen together 5 times in location A and 3 times in location B, the edge connecting them would hold a vector which contains 5 instances of A and 3 instances of B. In testing, given a list of BLEs seen at a time frame, we consider all pairs of BLEs and check whether they have been seen together previously and if they have, what the location was. To limit the search space for this, we require 30 co-occurrences at least, which translates to two BLE addresses being together for 90 seconds or greater.

3.1.2 Environment-aware clustering. In light of location information obtained from our BLE-location mapping, we cluster the activities based on their locations and their inherent similarity so that we can model similar activities together. Each specific context

is associated with a certain set of activities. Therefore, we learn contexts which make the observation of activities most likely. We learn unknown contexts and the distribution of each user's specific activity likelihood in each of these contexts to improve event recognition. This is determined in an expectation maximization approach (EM) where the mixture components were context specific activity recognition models learned and identified by the user, and the eating detection improved under that. This was achieved by defining the log-likelihood of observing activities given contexts as:

$$\begin{aligned} \log P(\text{ACTIVITY}|X, \theta) &= \sum_{n=1}^N \log P(\text{activity}_i|x_i, \theta) \\ &= \sum_{n=1}^N \log \sum_{c=1}^{N_c} P(\text{activity}_i, c_i = c|x_i, \theta) \quad (1) \end{aligned}$$

where θ are the mixture component parameters, N is the number of data samples, and c is the number of expected clusters (contexts) that each data point belongs to.

The log-likelihood introduced above can be maximized using EM algorithm [8]. However, in this unsupervised formulation, clusters do not necessarily contain environmental information. In order to inject this information into the latent variables, the E-step of the EM algorithm is regularized with the Kullback-Leibler (KL) divergence [11]. The KL divergence was used as a measure of similarity between two distributions to penalize the posteriors that were less similar to the context distribution collected from the user's annotations. Therefore, the E-step in EM algorithm is changed to:

$$q^{k+1} = \arg \max_{\theta} \mathbb{E}_{q^{k+1}} \log P(X, C|\theta) - \lambda KL(q||context) \quad (2)$$

By using this formulation, we promote posterior distributions which are more similar to environmental distribution in order to inject environment meaning into them. The M-step remains as standard to improve activity recognition tasks [8].

3.1.3 Context-aware models. In order to implement our context-aware system, two separate neural networks are used. One network was dedicated to environment estimation. The other network performed user's event recognition given sensor readings and the location estimated by the first network. Our framework is shown in Fig. 2. Each iteration has two sub-steps: environment estimator training and environment-aware classifier training. In the first sub-step, the environment estimator is to be optimized and the environment-aware classifier is frozen. In the second sub-step, the environment-aware classifier network is optimized given the distribution over locations which is the output of the environment estimator network. As a consequence, a number of parameters are to be modified accordingly.

The number of different environments is considered a hyperparameter that essentially represents the number of mixture models. This hyperparameter needs to be determined beforehand. In our supervised approach, we chose the number of environments to be the number of places for which each user has annotated data, that can vary by user. In Equation 2, λ is the penalty parameter used to constrain the posterior distribution over environments. In essence, this varies the impact of environment on the event recognition,

allowing activities that do not belong to certain environments to still be detected. We grid search between [0, 10] to evaluate the improvement in event recognition. We limit ourselves to 10 because we observed that the largest impact came when λ changes from zero to one and that the impact lessens with each subsequent increase.

3.2 Intelligent Annotation

With an accurate, environment-aware classifier, we then focused on intelligent annotations to enhance the use of wearable sensors in NE. To increase the accuracy through a better training dataset, the system needs to have label information for critical and novel samples. These novel observations are created by variations in the way an activity can be performed or an event can happen. This challenge becomes more significant when algorithms are trained on the data of certain users, but used by a new user. Thus, the system needs to update over time. To address this issue, the system needs to have access to the labels for those novel observations for retraining. This information can be obtained by asking the user for data annotations.

The annotations obtained from users can improve the event detection accuracy. However, such annotations should be obtained intelligently. An intelligent system should be able to detect informative instances. Such samples are typically those for which the system is not confident. Therefore, we propose a score that takes into account the classifier's uncertainty and restricts the number of queries per day. The score is used to detect critical samples and then request annotations. These annotations are then used to improve accuracy and generalizability of the system through retraining.

The proposed score for identifying the importance of a sample takes into account the following parameters: 1) uncertainty of the classifier $C(p(y|X))$, 2) the number of questions that have already been asked in the same day $\phi(k)$, 3) the confusion matrix (γ) and 4) the similarity of the input data to the previously observed data ($D(X)$). Equation 3 shows the proposed score, where α_{1-4} are hyperparameters that are chosen heuristically ranging from 0 – 1 in the increments of 0.25. Those hyperparameters determine the importance of each score parameter on the final decision.

$$S = \alpha_1.C(p(y|X)) + \alpha_2.\phi(k) + \alpha_3.\gamma + \alpha_4.D(X) \quad (3)$$

The intuition behind each of above parameters are as follows. The uncertainty of the classifier represents how confident the classifier is about its decision for sample. Low confidence, or too much confidence, would indicate that the label that is predicted by the classifier could be wrong. We calculate the mean and variance of the output of the classifier, $p(y|X)$ (*i.e.*, the probability of each activity to be correct) for all classes over all training data. For class j , μ_j^y is the mean and σ_j^y the Standard Deviation (SD) of the probabilities given by the classifier over the whole training data. For a new sample, the value of $C(p(y|X))$ is calculated based on Equation 4.

$$C(p(y|X)) = \begin{cases} 0, & \text{if } |p(y|X) - \mu_j^y| < \sigma_j^y \\ |p(y|X) - (\mu_j^y + \sigma_j^y)|, & \text{otherwise} \end{cases} \quad (4)$$

where $c = \operatorname{argmax} p(y|X)$.

To reduce the user burden, the number of questions that can be asked per day is restricted to K . If the system has already asked

the user K times within a day, then asking more questions should be done very carefully and less frequently. In Equation 5 when the number of already asked questions k is large, the probability of asking question is reduced.

$$\phi(k) = \frac{K - k}{K} \quad (5)$$

Classifier uncertainty cannot always be trusted [10], so it should not be the only deciding factor. A recognition confusion matrix can be used to judge uncertainty. For classes which the system predicts correctly, fewer user annotations are needed. However, for classes on which the system makes mistakes, there is a need to request user annotations more frequently. This is calculated by Equation 6.

$$\gamma = \frac{TP_c + FN_c}{TP_c} \quad (6)$$

where TP_c is the true positive and FN_c is the false negative for the class c , which is the decision of the classifier ($c = \operatorname{argmax} p(y|X)$). After the system asks the user for annotations on a sample, the classifier is retrained and the confusion matrix is updated.

Finally, if the input data is not similar to the data of any detected class (c), the sample could potentially be an unseen data type, and a prompt for user annotation would occur. On the other hand, that data could be an outlier, not requiring user prompting. To address this complication, the method analyzes the Euclidean distance between the current sample (X) and the mean point of the training data that belongs to the same class μ_c^X as Equation 7.

$$D(X) = \begin{cases} 0, & \text{if } \|X - \mu_c^X\| < m * \sigma_c^X \text{ or} \\ & \|X - \mu_{neighbor}^X\| > n * \sigma_{neighbor}^X \\ \frac{\|X - \mu_c^X\|}{\max(\|X - \mu_j^X\|)}, & \text{otherwise} \end{cases} \quad (7)$$

where μ_c^X is the mean of all training data that belong to class c and σ_c^X is their SD and $c = \operatorname{argmax} p(y|X)$ is the decision of the classifier. $\mu_{neighbor}^X$ is the closest mean class point to the sample X and $\sigma_{neighbor}^X$ is its SD. $\max(\|X - \mu_j^X\|)$ shows the maximum distance from the input X to the mean of any class and is used to normalize the value of $D(X)$. On the other hand, m and n are hyper parameters that can be chosen empirically. In our study $m=1$ and $n=5$. In Equation 7 for a sample that is close to the other samples of the same class, the probability of asking is then reduced. The same happens for a sample that is farther than all class boundaries, as it is a candidate of being an outlier.

After calculating S , it is compared to a constant threshold parameter, Th . The instances for which the score is higher than the threshold will be nominated for user annotation. However, a single windowed prediction may not be accurate. Additionally, most activities of daily living, such as eating, occur as longer, cyclical, repetitive motions. Thus, S is calculated for a series of consecutive windows. If S is higher than Th for N times within the last T windows, the system asks the user for that specific time interval. In this work, T was set to 10 (30 seconds) and N was set to 5. If the fifth instance with score $S > Th$ happens within last 10 windows, that instance is taken as a candidate for requesting feedback.

We should also bear in mind that we should not ask for annotations too often. A system parameter t_{min} determines a window

Table 1: The accuracy of detecting eating moments.

Subject	Train/test interval	Context		Retraining	
		Without	With	Before	After
Sub1	first	0.60	0.64	0.64	0.74
	second	0.73	0.80	0.80	0.82
Sub2	first	0.69	0.73	0.73	0.75
	second	0.73	0.76	0.76	0.77
	third	0.74	0.80	0.80	0.82
Sub3	first	0.84	0.86	0.86	0.84
	second	0.83	0.85	0.85	0.86
	third	0.84	0.86	0.86	0.88
Sub4	first	0.71	0.80	0.80	0.81
	second	0.76	0.74	0.74	0.75
Sub5	first	0.73	0.79	0.79	0.82
	second	0.70	0.75	0.75	0.76
Sub6	first	0.62	0.65	0.65	0.67
	second	0.86	0.88	0.88	0.89

in which no second questions should be asked. Thus, if there is a candidate in less than t_{min} minutes after the last labeling request time, it will be ignored.

The maximum number of annotation requests a user can get in a day is indicated by K . Assuming each day of data collection to be approximately 15 hours, the number of annotations K was set empirically to 20. This was determined since a user would likely annotate data 1 to 2 times every hour. A smaller K significantly reduces the classifier's accuracy. The value for Th on which the algorithm determines whether to ask for an annotation or not was heuristically set to 0.4, as most scores were below 0.5 and above 0.2. Higher values discourages the algorithm from requesting annotations, causing a lesser number of annotations which would cause a reduction in the classifier's accuracy. On the other hand, lowering the threshold causes the algorithm to request annotations on unimportant samples, limiting accuracy improvements.

4 EXPERIMENTAL EVALUATION

4.1 Case Study

The goal of this case study is to show our system's ability to improve the recognition of eating moments. In our study, 12 participants were given a smartwatch. Participants wore the watch throughout the day for at least two weeks and labeled their eating moments through audio recordings. Only the data of 6 participants was usable given the quality of these annotations. The participants forgot to label the eating moments and others forgot to label the data at the right time. All the signals were evenly re-sampled at 20Hz and 10 statistical features were extracted following a windowing approach with a 50% overlap based on the findings from [21]. We extracted the following statistical features: mean, SD, median, interquartile range, skewness, root mean square, zero crossing rate, variance, mean crossing rate, and kurtosis. This study was reviewed and approved by the Texas A&M University human subjects protection program (IRB #2018-0998).

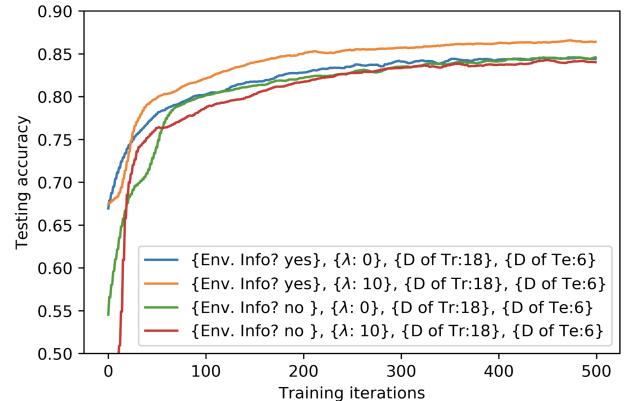


Figure 3: Improved classification testing accuracy through environment information with training iterations. In legend, D of Tr/Te means days of training/testing.

4.2 Findings and Results

Table 1 compares the accuracy of the model in detecting eating moments with and without environment information obtained from BLE addresses for each subject with different train/test intervals. We see an improvement of 2.4% in accuracy through the environment-aware modeling and that for all subjects, using environment information improves the accuracy of the system as expected. Moreover, as the amount of the training data increased, the accuracy of the model improved. The results show that the model keeps improving with increased user annotations. As an example, Fig. 3 illustrates how model accuracy improves with more training iterations.

To investigate the impact of the opportunistic collection of labels from the user, the data for each user was divided into three parts: training, feedback, and testing sets. Each set was obtained by selecting a number of days out of the total days of data collected from each participant. The training data was used to train the environment-aware model and the feedback data was used to ask user for annotations. The number of annotations that can be asked per day (K) was set empirically as described in Section 3.2. Fig. 4 shows an example of the calculated scores during a whole day of user data. Using the feedback set of data, a score was calculated each time window as explained in Section 3 in the annotation algorithm. The threshold is empirically chosen to be $Th = 0.4$ and shown in the Fig. 4. Based on this score, the system solicits the user for providing the label for the critical samples and then used those data to retrain the model. Table 1, under the column Retraining-After, shows the improvement in the test accuracy of the model after retraining with those data, assuming context information was applied. This improvement is gained because the model is modified to learn the correct labels of previously uncertain data samples.

5 FUTURE WORK

This work demonstrates that the number of annotations requested from a user can be reduced in order to improve adherence while maintaining performance. Future studies will focus on the prediction of nutrition intake of meals aided by our automatic detection.

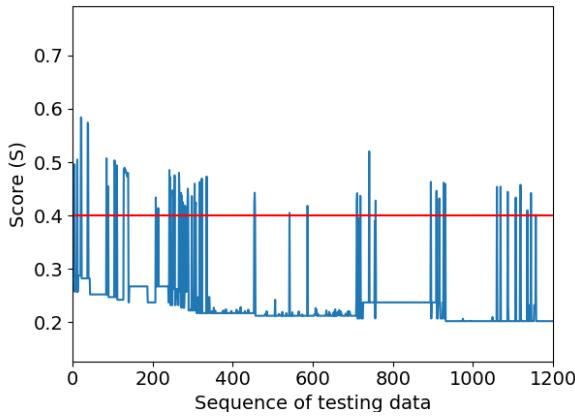


Figure 4: The proposed score calculated for one day. The red line indicates the threshold that was chosen.

Additionally, temporal shift in the labels needs to be considered, with the impact on the environment-aware recognition algorithm and the opportunistic labeling engine. The improvement of this recognition system will allow for more dynamic information to be discovered and additional application areas to be explored.

6 CONCLUSION

This paper presented an algorithm based on classifier uncertainty to obtain user annotations in NE. It leveraged environment information obtained through nearables to improve the classifier accuracy and reduce annotation requests. Using a semi-supervised approach that predetermines environments, we identified contextual information and considered uncertainty in detection algorithms that guided annotation requests for the user. The system was validated through a dietary monitoring study on which eating moment annotations were obtained from users in order to improve the detection of such events, improving accuracy while limiting annotations, enabling long-term data collection.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation, under grants CNS-1734039, EEC-1648451, National Institutes of Health under grant R01EB028106-01, and Texas A&M Triads for Transformation (Award#422). 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] Mark V Albert, Yohannes Azeze, Michael Courtois, and Arun Jayaraman. 2017. In-lab versus at-home activity recognition in ambulatory subjects with incomplete spinal cord injury. *Journal of neuroengineering and rehabilitation* 14, 1 (2017), 10.
- [2] Ling Bao and Stephen S Intille. 2004. Activity recognition from user-annotated acceleration data. In *International Conference on Pervasive Computing*. Springer, 1–17.
- [3] Abdelkareem Bedri, Richard Li, Malcolm Haynes, Raj Prateek Kosaraju, Ishaan Grover, Temiloluwa Prioleau, Min Yan Beh, Mayank Goel, Thad Starner, and Gregory Abowd. 2017. EarBit: using wearable sensors to detect eating episodes in unconstrained environments. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 37.
- [4] Terrell R Bennett, Hunter C Massey, Jian Wu, Syed Ali Hasnain, and Roozbeh Jafari. 2016. MotionSynthesis Toolset (MoST): An Open Source Tool and Data Set for Human Motion Data Synthesis and Validation. *IEEE Sensors Journal* 16, 13 (2016), 5365–5375.
- [5] Chongguang Bi, Guoliang Xing, Tian Hao, Jina Huh, Wei Peng, and Mengyan Ma. 2017. Familylog: A mobile system for monitoring family mealtime activities. In *Pervasive Computing and Communications (PerCom), 2017 IEEE International Conference on*. IEEE, 21–30.
- [6] Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*. ACM, 92–100.
- [7] Larry Chan, Vedant Das Swain, Christina Kelley, Kaya de Barbaro, Gregory D Abowd, and Lauren Wilcox. 2018. Students’ Experiences with Ecological Momentary Assessment Tools to Report on Emotional Well-being. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 1 (2018), 3.
- [8] Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)* (1977), 1–38.
- [9] Matthew Fuller-Tyszkiewicz, Helen Skouteris, Ben Richardson, Jed Blore, Millicent Holmes, and Jacqueline Mills. 2013. Does the burden of the experience sampling method undermine data quality in state body image research? *Body Image* 10, 4 (2013), 607–613.
- [10] Yarin Gal and Zoubin Ghahramani. 2015. Dropout as a Bayesian approximation. *arXiv preprint arXiv:1506.02157* (2015).
- [11] Kuzman Ganchev, Ben Taskar, and João Gama. 2008. Expectation maximization and posterior constraints. In *Advances in neural information processing systems*. 569–576.
- [12] Shravan Gaonkar, Jack Li, Romit Roy Choudhury, Landon Cox, and Al Schmidt. 2008. Micro-blog: sharing and querying content through mobile phones and social participation. In *Proceedings of the 6th international conference on Mobile systems, applications, and services*. ACM, 174–186.
- [13] Donghai Guan, Weiwei Yuan, Young-Koo Lee, Andrey Gavrilov, and Sungyoung Lee. 2007. Activity recognition based on semi-supervised learning. In *Embedded and Real-Time Computing Systems and Applications, 2007. RTCSA 2007. 13th IEEE International Conference on*. IEEE, 469–475.
- [14] Haik Kalantarian, Nabil Alshurafa, Tuan Le, and Majid Sarrafzadeh. 2015. Monitoring eating habits using a piezoelectric sensor-based necklace. *Computers in biology and medicine* 58 (2015), 46–55.
- [15] Ashish Kapoor and Eric Horvitz. 2008. Experience sampling for building predictive user models: a comparative study. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 657–666.
- [16] Young-Ho Kim, Jae Ho Jeon, Bongshin Lee, Eun Kyoung Choe, and Jinwook Seo. 2017. OmniTrack: A Flexible Self-Tracking Approach Leveraging Semi-Automated Tracking. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 67.
- [17] Li-Jia Li and Li Fei-Fei. 2010. Optimol: automatic online picture collection via incremental model learning. *International journal of computer vision* 88, 2 (2010), 147–168.
- [18] Brent Longstaff, Sasank Reddy, and Deborah Estrin. 2010. Improving activity classification for health applications on mobile devices using active and semi-supervised learning. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2010 4th International Conference on-NO PERMISSIONS*. IEEE, 1–7.
- [19] Andrea Mannini and Stephen Intille. 2018. Classifier Personalization for Activity Recognition using Wrist Accelerometers. *IEEE journal of biomedical and health informatics* (2018).
- [20] Emmanuel Munguia Tapia, Stephen S Intille, and Kent Larson. 2004. Activity recognition in the home using simple and ubiquitous sensors. In *International conference on pervasive computing*. Springer, 158–175.
- [21] Edison Thomaz, Irfan Essa, and Gregory D Abowd. 2015. A practical approach for recognizing eating moments with wrist-mounted inertial sensing. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 1029–1040.
- [22] Rui Wang, Fanglin Chen, Zhenyu Chen, Tianxing Li, Gabriella Harari, Stefanie Tignor, Xia Zhou, Dror Ben-Zeev, and Andrew T Campbell. 2014. StudentLife: assessing mental health, academic performance and behavioral trends of college students using smartphones. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing*. ACM, 3–14.
- [23] James Y Xu, Hua-I Chang, Chieh Chien, William J Kaiser, and Gregory J Pottie. 2014. Context-driven, prescription-based personal activity classification: methodology, architecture, and end-to-end implementation. *IEEE journal of biomedical and health informatics* 18, 3 (2014), 1015–1025.
- [24] Yan Zhou and Sally Goldman. 2004. Democratic co-learning. In *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*. IEEE, 594–602.