I Did Not Smoke 100 Cigarettes Today! Avoiding False Positives in Real-World Activity Recognition

Le T. Nguyen, Ming Zeng, Patrick Tague, Joy Zhang
Carnegie Mellon University
Moffett Field, CA, USA
{le.nguyen, ming.zeng, patrick.tague, joy.zhang}@sv.cmu.edu

ABSTRACT
Activity recognition (AR) systems are typically built and evaluated on a predefined set of activities. AR systems work best if the test data contains and only contains these predefined activities. In real world applications, AR systems trained in this manner generate serious false positives, for example if "smoking" is one of the activities in the training data but "lifting weights" is not. Due to the similarity of two activities, an AR system may report a user smoking 100 times a day but he actually did a bicep workout 100 times. In this work, we propose a new approach to train an AR system leveraging the large quantity of unlabeled data which reflects activities users perform in real life. The proposed *mPUL* (Multi-class Positive and Unlabeled Learning) approach significantly reduces the false positives. We argue that mPUL is a much more effective training method for real-world AR applications.

Author Keywords
Activity Recognition; Multi-Class Positive and Unlabeled Learning; Semi-Supervised Learning; Open-World

ACM Classification Keywords
I.2.1 Artificial Intelligence: Applications and Expert Systems

INTRODUCTION
An activity recognition (AR) system is typically trained to recognize a set of activities that are relevant to some applications. For example, a mobile application designed to monitor a user's life style for potential diabetes risk is interested to know how often the user smokes, exercises, or performs various other activities. Researchers list 10 activities that are relevant to diabetes including smoking, jogging, walking and biking and collect training data that contain these 10 activities with labels. An AR system is trained from this labeled data and then deployed in the mobile application to monitor the user's life style.

Ubicomp '15, September 07 - 11, 2015, Osaka, Japan

ACM 978-1-4503-3574-4/15/09$15.00
DOI: http://dx.doi.org/10.1145/2750858.2804256

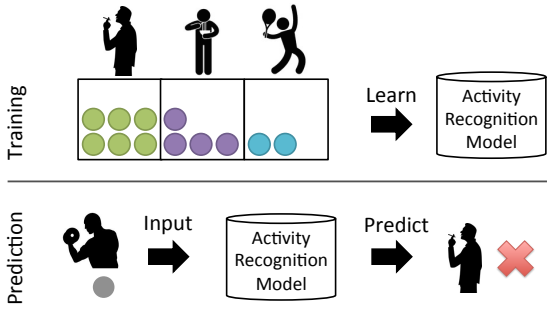


Figure 1: Traditional AR systems perform poorly when encountering unknown activities (activities not seen in the training data) such as lifting weights.

The performance of an AR system is usually evaluated on a held-out set measuring the accuracy of recognizing labeled activities. While many AR systems report high accuracy in a lab setting, they suffer in real-world applications. One of the main problems is false positive. For example, smoking and bicep workout are similar in arm motions and the AR system is only trained to recognize smoking, the AR system detects a user smoked 100 times in a day when the user actually went to the gym and did a bicep workout 100 times (Figure 1).

The fundamental problem of false positive activity recognition is that the training setup assumes that the labeled activities are all possible activities the system will ever encounter. Since the system does not have knowledge about any activities other than the ones observed in the training data, it is prone to making false positive predictions, i.e., the system will falsely predict bicep workout as smoking. We refer to this training setup as the **Closed-Word (CW)** condition because it assumes the training data contains all possible activities the system will observe [10]. Obviously, CW assumption is not realistic since there are large number of human activities [4] users perform; it is very expensive, if possible at all, to label all activities in the training data. We refer to the application of AR systems in realistic scenarios as the **Open-World (OW)** condition, i.e., the training data contains only a subset of activities an AR system will encounter after deployed.

The goal of this work is to build an AR system with lower false positive rate in the OW condition. We propose an approach called Multi-class Positive and Unlabeled Learning (mPUL), which leverages both labeled training data and additional unlabeled data collected in the OW. The unlabeled data

is collected from mobile devices users carry daily. Since we do not require any annotation, such data is abundant and almost free of cost. Activities in the unlabeled data (e.g., bicep workout) are referred to as **unknown** or **unobserved** activities if they are not those labeled in the training data (Figure 2).

We reframe the problem of reducing false positive as: *how to identify the unknown activities to avoid classifying them as one of the known activities*. The key challenge is to identify unknown activities without having any labeled data for the unknown activity classes.

The proposed mPUL approach is a multi-class extension of Positive and Unlabeled Learning (PUL) [5]. PUL is traditionally used to solve two-class classification problems, where one class is known but the other is not; we have labeled data only of the known activities, but not of the unknown activities. As the traditional PUL handles only two-class problems with one known class, we extend this approach to M-class problems with m ($m < M$) known classes to accommodate the multi-class AR problem. We show that our proposed approach is highly effective, especially in cases with a small number of known activities ($m \ll M$). To the best of our knowledge, this work is the first to explore the PUL learning paradigm in the AR domain, specifically to address the challenges in the Open-World.

Our contributions are summarized as follow.

- **Closed-World (CW) vs. Open-World (OW) Assumptions**: We explicitly formulate the CW assumption, which is implicitly made by the existing AR approaches. We relax such assumption and show how OW better reflects the real-world conditions. We then study how traditional AR techniques perform under both CW and OW assumptions.

- **mPUL**: To address the challenges in OW, we propose Multi-class Positive and Unlabeled Learning (mPUL), which leverages the unlabeled data to effectively handle activities unknown at the training time and thus reduces false positives when deployed in the OW.

- **Empirical study**: We study the performance of mPUL in both controlled and "in the wild" condition. We show that mPUL is highly effective, especially in cases when only a small number of activities are known at the training time. Furthermore, we identify weaknesses of the proposed approach and discuss potential extensions to address them.

RELATED WORK

Chavarriaga et al. [3] use a thresholding approach to reject predictions with a low confidence score to reduce false positives. In the OW condition, however, thresholding has the potential of increasing false negatives [22].

Another way of dealing with unknown activities is to ask users to label sensor data after the system is deployed. Active learning has been used to reduce the required number of user annotations by asking users to label only "informative" instances [19, 14, 16]. Such approaches assume users are cooperative and have time and cognitive bandwidth to annotate activities in their daily routine.

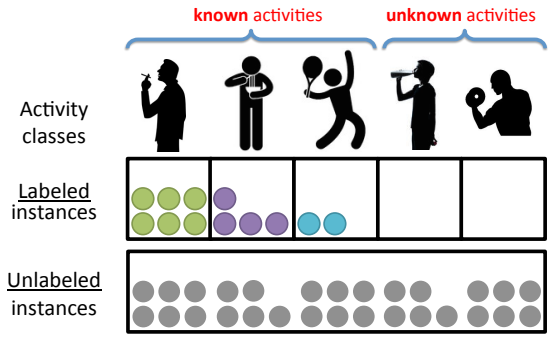


Figure 2: To improve the AR model, we propose leveraging unlabeled data captured while the system is in use. This unlabeled pool contains instances of both known and unknown activities.

The problem of identifying unknown activities is to a certain degree related to the one-class learning problem [9], where the training data contains labeled instances of only a single class. In the AR domain, the one-class problem has been used to detect anomalies such as people falling on the floor in elder care scenarios [25, 23]. In such cases, one can collect training data of simulated falls. However, collecting training data for all not-falling activities is not feasible.

The approach proposed in this work can be considered as a multi-class, semi-supervised version of the one-class learning approach. In AR research, many semi-supervised learning techniques have been explored to take advantage of availability of the large amount of unlabeled sensor reading. One of the most commonly used technique in AR is self-training [19, 14]. In self-training, a small pool of labeled data is used to train an initial classifier. This classifier is then used to recognize the unlabeled instances. Instances with high confidence are then moved into the labeled pool, and the classifier is retrained iteratively until no more instances are added. Besides self-training, techniques such as co-training [6, 19, 14] and graph-based label propagation [18] have been proposed to utilize the multiple sensing modalities and the temporal nature of unlabeled signal data. Further work [1, 7] suggests adapting AR systems by considering the similarity between training data and unlabeled data collected when the system is used. All the semi-supervised studied so far implicitly make the CW assumption, where unlabeled instances are assumed to belong to one of the known activities.

To a certain degree, our work is related to the work aiming at recognizing activities for which little or no training data is available (attribute-based learning [15, 4]). The authors assume that an activity (such as cycling) can be described through a set of attributes (e.g., translation motion, cyclic motion and intense motion). Since driving can be also described by the translation motion attribute, one can use the training data labeled with driving to learn a model, which can recognize cycling even though no training data for cycling is available. This approach requires knowledge about the unobserved activities and how they can be described by the set of predefined attributes.

The key difference between our proposed approach and the work mentioned above are: 1) we leverage unlabeled data to identify unknown activities (in contrast to the traditional AR systems, one-class learning and active learning approaches); 2) we assume that unlabeled data can belong to known or unknown activities (in contrast with the traditional semi-supervised learning approaches); and 3) we do not assume any knowledge about the unknown activities (in contrast to attribute-based learning).

Although the proposed approach is applicable to any type of sensor data and features extraction method, we demonstrate its effectiveness in an AR system using wearable sensors.

LEARNING ACTIVITIES IN THE OPEN-WORLD

The goal of this work is to address challenges of AR used in a real-world environment. In the following, we first define the Closed-World assumption implicitly made by the existing AR work and propose a relaxed Open-World formulation to better describe the real-world conditions. We show the weaknesses of the traditional AR models applied in the Open-World and propose mPUL (Multi-class Positive and Unlabeled Learning) to address these weaknesses.

Closed-World (CW) and Open-World (OW)

In this work, the terms CW and OW characterize the completeness of the available information. In CW, we assume that in the training phase we observe the whole world and therefore have all the needed information. On the other hand, in OW we assume that in the training phase we can observe only a small part of the world. In the following, we formally define the terms CW and OW and explain their implications on an AR system.

To train an AR model we use a dataset (x, y), where x is an instance (e.g., a feature vector) and y is a class label. We denote A_O as the set of activity classes **observed/known** in the training dataset ($y \in A_O$):

$$A_O = \{a_1, a_2, \cdots, a_m\}. \tag{1}$$

We denote A as the universe of all possible activities with $A_O \subseteq A$, i.e., A contains all possible activities the AR system will encounter in the wild. The activity classes observed only in the wild, but not in the training data is referred to as **unobserved/unknown** classes $\overline{A_O}$:

$$\overline{A_O} = A - A_O. \tag{2}$$

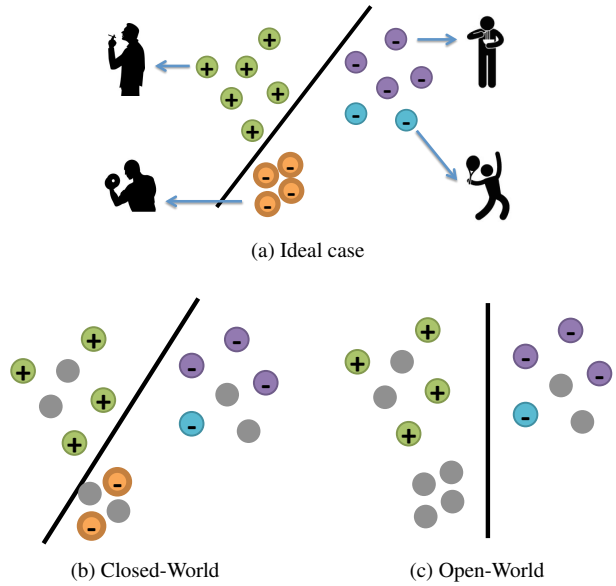
We formulate the CW and OW problems as

$$A_O = A \quad \text{in } \mathbf{CW} \tag{3}$$

$$A_O \subsetneq A \quad \text{in } \mathbf{OW}. \tag{4}$$

This directly implies that $\overline{A_O} = \emptyset$ in CW, whereas $\overline{A_O} \neq \emptyset$ in OW.These unobserved classes will have significant implications for the performance of an AR system.

In the following, we illustrate the implications through the scenario presented earlier. Our goal is to build a AR model for recognizing smoking, eating and playing tennis. The universe of all possible activities contains three mentioned activities and also the weight-lifting activity. Figure 3a shows the training dataset in an ideal case, where for each class in the activity universe we have sufficient labeled instances to learn a correct decision boundary.



(a) Ideal case

(b) Closed-World

(c) Open-World

Figure 3: In Open-World, some activity classes (such as lifting weights) are not observed in the labeled training dataset. Thus, using supervised learning methods, the learned classification boundary is less accurate compared to the Closed-World case, where all activity classes are observed.

Under the CW assumption, each class has at least one labeled instance (including the lifting weight class as shown in Figure 3b). In OW, some classes do not have any labeled instances (as shown in Figure 3c). Given that all activity classes are observed in CW, the learned decision boundary for the smoking activity is very similar to the boundary learned in the ideal case. On the other hand, in OW the negative training instances are not representative of the negative feature space. Thus, relying on them to learn a classification boundary will result in an inaccurate AR model.

Learning with Unlabeled Data

Since learning with unlabeled data is essential for the proposed approach, we first discuss the different assumptions about the unlabeled instances.

CW: Many existing AR works assume that unlabeled instances have to belong to one of the classes A_O observed in the training dataset [6, 19, 14]. Thus, semi-supervised techniques such as self-training can iteratively propagate labels to the unlabeled instances. This assumption corresponds to the CW assumption, since the unlabeled data cannot belong to any other classes besides A_O.

Semi-OW: Other AR systems treat instances not labeled by the user as a separate (null) class [17, 20, 3]. In other words, the unlabeled instances belong to one of the unobserved classes $\overline{A_O}$. This scenario assumes that the user intentionally left an instance unlabeled to indicate that the instance

does not belong to any of the activities of interest. AR models making this assumption would perform poorly if instances are left unlabeled for other reasons (e.g., the user forgot to label some of the known activities). This assumption can be considered as semi-OW, since it accepts that not all activity classes may have been observed at training time. However, it forces unlabeled instances to unobserved classes $\overline{A_O}$.

OW: In this work, we assume that an unlabeled instance can belong to any of the possible activity classes A. This fully supports the OW condition, since we assume that at the training time the unlabeled instances can belong to any of the observed A_O or unobserved classes $\overline{A_O}$.

mPUL: Multi-class Positive and Unlabeled Learning

We next describe the proposed mPUL approach in detail. First we explain the concept of the binary PUL and then describe how mPUL generalizes PUL to multi-class AR.

PUL (Positive and Unlabeled Learning) is a semi-supervised learning approach used to solve two-class (positive and negative) classification problems, where the training data contains labeled instances of only the positive class, but no negative instances are available. Such problems occur in many real-world applications such as in information retrieval, where the goal is to build a system for retrieving documents with a user's topic of interest (e.g., politics) [24, 11]. In such scenarios, one can easily obtain positive instances by asking the user to provide a few documents of interest. However, it is challenging and labor-some to collect a representative (unbiased) set of negative instances. For example, while a user might label a few sport-related documents as negative instances, they likely do not represent the whole negative document space. It is known that using a non-representative set of instances leads to significant degradation of prediction results [12].

To address this problem, PUL uses only the positive instances with the unlabeled data (easily obtainable from the Internet) to build the binary classification model. The key assumption is that the unlabeled pool contains both positive and negative instances. More importantly, it is assumed that the negative instances in the unlabeled pool form a representative set of negative instances. PUL is therefore used to identify which instance in the unlabeled pool is positive and which ones are negative. This can be done by initially assuming that all unlabeled instances are negative [11]. Based on this assumption, an initial classifier is trained and then used to predict the labels of instances in the unlabeled pool. These new predictions are used to re-label the training dataset. This iterative process of training and relabeling is repeated until convergence is achieved (similar to self-training [19]).

In this work, we build mPUL on top of a PUL formulation, which solves the above problem directly (instead of iteratively) [5]. Similar to the iterative approach, we first assume that all unlabeled instances are negative and build a classifier g based on this assumption. Obviously, this classifier is biased towards a small set of labeled positive instances in the training data. Thus, by scaling the probabilistic output of g by the constant c, one can obtain the true conditional probability $p(y = 1|x)$. In the following, we formally describe this approach in detail.

Let x be an instance and $y \in \{-1, 1\}$ its negative or positive activity label. Additionally, $s \in \{0, 1\}$ indicates whether the instance label is known. Thus, each instance is represented as (x, s, y)-tuple. For example, $(x_1, 1, 1)$ represents a positive instance, while $(x_2, 0, ?)$ represents an unlabeled instance. Note that $s = 1$ implies $y = 1$, since in the training dataset we know the labels of only the positive instances.

Our goal is to learn the function $f(x) = p(y = 1|x)$ from the training dataset. In case $s = 1$ for all instances x, we can learn $f(x)$ directly from the labeled dataset (x, y). However, in our scenario, many of instances have unknown labels ($s = 0$). In such cases, it has been shown that $f(x)$ can be learned in a non-traditional way using the following equation [5]:

$$f(x) = g(x)/c \tag{5}$$

$$g(x) = p(s = 1|x) \tag{6}$$

Note that $g(x)$ is a classifier itself, estimating the probability that the label of x is known. This classifier is learned from the dataset (x, s). c is a constant independent of x, estimated using a holdout validation set [5]. This problem formulation can be interpreted in the following way. 1) Since we cannot learn $f(x)$ directly from (x, y), we learn this function indirectly from (x, s). 2) Since $s = 1$ implies $y = 1$ (all instances with known labels are positive), (x, s) corresponds to the dataset, which assumes that all unlabeled instances are negative. 3) $g(x)$ learns a conditional probability of $p(y = 1|x)$ under the assumption that all unlabeled instances are negative. 4) $g(x)$ is biased towards a small set of labeled positive instances in the training data. Thus, by scaling $g(x)$ by the constant c, one can obtain the true conditional probability $p(y = 1|x)$.

For the above equations to hold, it is assumed that the instances with known labels ($s = 1$) are selected randomly from the set of all actually positive instances,

$$p(s = 1|x, y = 1) = p(s = 1|y = 1). \tag{7}$$

This assumption can be explained through the following example: suppose the user smoked 10 times but only labeled five smoking instances. Our assumption states that the decision of which activity instance to label does not depend on how the activity was performed. As a counter-example, if the user smokes using both hands, but she only labels instances when the cigarette is in her left hand, this assumption does not hold. Hence, we assume she labels smoking activities independent of how she holds the cigarette.

In this work, we assume the latter case, i.e., the obtained activity labels are independent of how the activity was performed. The validity of the assumption, however, often depends on the strategy for obtaining the activity labels. For example, if the user is asked to input the activity label into a mobile device while the activity is performed then the annotation process will directly impact how the activity is performed. On the other hand, if the user can label activities after they are performed, the fact that an activity is labeled (or not) is dependent on the user's memory, but not on how the activity was performed. This observation brings up an interesting

question of how activity annotation strategies impact AR performance, which will be further explored in our future work.

mPUL: In this work, we extend binary PUL to solve the multi-class AR problem. The proposed extension involves three key parts: 1) 1-vs-other decomposition, 2) extension from positive unlabeled (P-U) to positive, negative and unlabeled (P-NU) and 3) implicit detection of unknown activities. We describe each of these parts in detail.

The first part of the extension involves decomposing the m-class AR problem into m binary-class problems using the 1-vs-others strategy [21], i.e., one binary classifier for each activity class. Using this strategy, there are two main questions: 1) Training: How to train each binary classifier. 2) Prediction: How to fuse predictions of m binary classifiers.

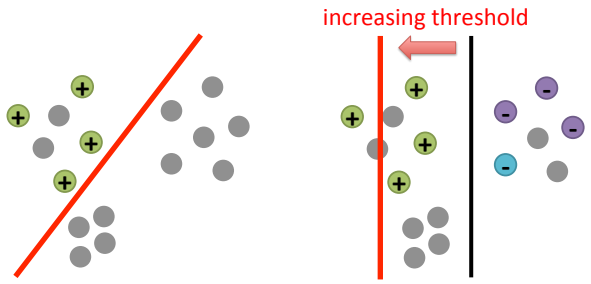
Training: In a traditional supervised setting, a binary classifier is trained using positive and negative instances. For example, to build a classifier to recognize smoking, we need instances of smoking and non-smoking activities. Given the scenario presented in introduction (Figure 2), the non-smoking activities are traditionally represented by the eating and playing tennis activities. Obviously, these two activities are not representative of all non-smoking activities (since it does not include instances of drinking and lifting weights). As illustrated in Figure 3c, training a model using an non-representative set of negative instances will lead to learning an incorrect decision boundary.

To find a representative set of negative instances, we propose to use both Labeled and Unlabeled data, instead of relying only on the labeled instances. The pool of labeled instances can be divided into the Positive and the Negative pool. We propose to merge the Negative pool with the pool of Unlabeled instances and further refer to this merged pool as the NU (Negative+Unlabeled) pool. Figure 4a shows the instances of the NU as gray circles. Note that each NU instance belongs to either the positive (smoking) or to the negative (not-smoking) class. More importantly, the set of negative instances in NU forms a *representative* set of negative instances, i.e., this set contains instances of all negative (not-smoking) classes (including the "unknown" drinking and lifting weights activities).

Since labels of many instances in NU are not known, we use PUL to identify which of these instances belong to the positive and negative classes. The identified positive and negative instances are then used to find the unbiased decision boundary. Note that in contrast to traditional PUL, which uses only the positive and unlabeled instances (P-U), mPUL uses all available positive, negative and unlabeled instances (P-NU).

The outcome of the mPUL training process is a set of m classifiers, each estimating a conditional probability $f_i(x) = p(y_i = 1|x)$ for an activity a_i ($i \in \{1, \ldots, m\}$).

Prediction: At prediction time, given a new unseen instance x, we use all m binary classifiers to compute the conditional probabilities $f_i(x) = p(y_i = 1|x)$. We then select the class y^* with highest conditional probability $p^* = \max_i f_i(x)$.



(a) mPUL (b) Thresholding

Figure 4: Thresholding moves the initial decision boundary in the direction of reliable positive instances. This results in a higher precision at the cost of lower recall.

As the third part of the extension, we propose to output y^* as a prediction only if $p^* > 0.5$, otherwise the unknown class is predicted:

$$y_{mPUL} = \begin{cases} y^* & \text{if } p^* > 0.5 \\ unknown & \text{else} \end{cases} \quad (8)$$

Thus, instead of blindly trusting the most confident classifier, we also require that this classifier would make a positive prediction in the binary case. The unknown class is predicted if none of the binary classifiers believe that the unseen instance x belongs to their class.

mPUL vs. Thresholding

The proposed approach has some similarities to the thresholding approach, since both approaches aim at improving the performance of an AR system by reducing the number of false positives and thereby increasing the precision. Thresholding is a wrapper built on top of a base classifier, which can output a confidence score for its prediction. Thresholding decides whether to reject the prediction based on this score, i.e., predictions with confidence scores lower than a threshold τ are rejected and thus assigned to the "unknown class". This is desirable in scenarios when precision is highly important, i.e., cost of false positives is significantly higher than the cost of false negatives. Clearly the thresholding imposes a trade-off between precision and recall, i.e., with a higher threshold, we expect to achieve a higher precision at the cost of reducing recall [22].

The key difference between the thresholding approach and the mPUL is illustrated in Figure 4. mPUL uses a representative set of negative instances to learn the decision boundary. On the other hand, thresholding uses a traditional supervised classifiers as the base classifier to learn the initial decision boundary. As explained above, in OW the supervised classifier is prone to learning from non-representative set of instances, resulting in an incorrect decision boundary to begin with. By increasing τ, the thresholding approach moves the initial decision boundary in a direction orthogonal to the boundary. Thus, with a high τ, the wrapper algorithm accepts only highly confident predictions, in fact increasing precision. However, it also causes a significant reduction of recall, since many actually positive instances are not correctly detected.

EVALUATION

The goal of this work is to address the challenges in the OW condition, where the training dataset contains only a subset of activities appearing in the wild. To evaluate the proposed approach, we conduct experiments on two public datasets, both containing a large amount of activities. This allows us to evaluate use cases where the training data contains only a small set of activities of interest. We first study the performance of the proposed approach under various OW conditions with respect to the number of observed/unobserved classes and the number of labeled/unlabeled instances. We then further study how the proposed approach performs in real-world conditions while dealing with additional challenges such as imbalanced activity distribution.

Evaluating AR in OW

Dataset: For the evaluation of the first set of experiments we use a public Daily and Sport activity dataset [2], which contains 19 different activities performed by 8 users. The activities include basic locomotion activities (sitting, standing, walking, etc.) and more specialized exercise activities (exercising on a stepper, cycling on an exercise bike, rowing, playing basketball, etc.). This dataset is suitable for evaluating scenarios where we want to recognize specialized activities such as playing basketball, but we do often not have labeled instances from all other activities. Data was collected from five sensors placed on different parts of each user's body. To consider a more realistic scenario, we use data of only two sensors, one attached to the right arm (typical of a smart watch) and one attached to the left leg (typical of a mobile device in a pocket).

Feature extraction: For each user and each activity, 60 five-second segments of sensor readings were collected. From each five-second segment we extract one instance (i.e., feature vector) composed of statistical features including mean, standard deviation, minimum, maximum, energy and correlation between sensor axes of individual sensors [10]. Thus, we have 1140 instances (= 19 classes $\times$ 60 instances) for each user, totaling 9120 instances. Each instance is annotated with a user id and an activity class label.

Classifiers: In the following, we compare the results of mPUL with traditional supervised, semi-supervised and threshold-based classifiers. For mPUL, we use Logistic Regression as the base classifier to learn the conditional probability $p(y_i = 1|x)$.

For the supervised classification, we evaluated a range of classifiers including Naive Bayes, k-NN, Decision Tree, Logistic Regression and SVM. Since SVM with a linear kernel achieved the best results, we show the results only of this classifier.

For semi-supervised learning, we use self-training (**ST**) with Logistic Regression as the base classifier. In each self-training iteration, we move all unlabeled instances with prediction confidence higher than t to the labeled pool and retrain the model; we empirically set $t = 0.9$.

Since both SVM and ST do not have the capability of predicting an unknown class, we additionally evaluate self-training with thresholding (further referred to as **STT**). STT takes the predictions of ST and predicts an unknown class, whenever the ST confidence lower than τ, which is estimated using a holdout dataset through a cross-validation process [22].

Evaluation strategy: We evaluate the performance of AR systems using both a personalized and a universal AR model [13]. Since we observe similar trends from the results of both models, we report the results of only personalized AR model noting that similar observations are made about the results of the universal model.

In the personalized model, we train a classifier for each user based on a subset of that user's individual data and then test the model on the remaining user data. This simulates the scenario of a user labeling a few of her activities for building an activity recognition model. This model is further used for recognizing her own activities. We randomly split the 60 instances for each class into 30 *candidate* training instances and 30 test instances. Since the selection of training data involves randomness, we report average results over 10 runs.

Evaluation metric: As the evaluation metric, we use the macro-F1 score, computed as the F1 score for each activity class, averaged across classes. In the second part of this section, we will discuss in more detail why macro-F1 score is more informative than measures such as accuracy, commonly used in AR research [10].

F1 score with respect to an activity of interest a_i is calculated as the harmonic mean of precision and recall:

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$$
$$precision = \frac{TP}{TP + \mathbf{FP}}$$
$$recall = \frac{TP}{TP + FN}.$$

TP, **FP** and FN denote true positive, false positive and false negative counts. In this work, our goal is to avoid false positives caused by the unknown classes. This directly translates into achieving high precision for any OW condition. Due to the precision-recall trade-off [22], in the following experiments we evaluate the system's performance using all three measures - F1, precision and recall.

Amount of Labeled Training Data

In the first experiment, we consider the case of observing only a small number of activity classes in the training data. This corresponds to a scenario of building an AR system for recognizing a small set of activities of interest with only a small number of labeled instances.

To evaluate this scenario we first introduce two key parameters: the number of observed activity classes m and the number of labeled instances per class l (as shown in Figure 5). Both relate to the properties of the training dataset (and do not impact the test dataset). Let's consider a training set described by a (m, l)-tuple. To create such as a training set, we randomly select m activities from the Daily and Sport activity dataset. For each selected activity, we randomly select l

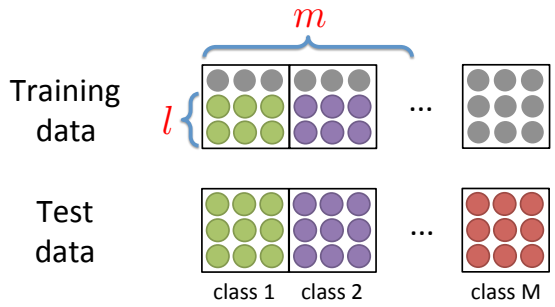


Figure 5: Two key evaluation parameters: the number of observed classes annotated as m and the number of labeled instances per class annotated as l. M indicates the total number of activities in the universe.

instances belonging to this activity class. Labels of all not-selected training instances are discarded, i.e., these instances become unlabeled. The obtained training data is then used to train a classification model, which is then used to predict labels of the test instances. Note that the test instances of the unknown activities should be ideally predicted as a separate "unknown class".

Figure 6 shows the results for $m = 5$, while varying the number of instances per class l. As expected, with the increasing amount of labeled data, mPUL improves its prediction performance. However, this is not the case with SVM and ST. This is mainly caused by these classifiers not being able to detect the unknown activities. Since we can observe only five activities in the training data, the remaining 14 activities remain unobserved. Thus, instances of all these 14 activities should be predicted as the unknown class. However, since both SVM and ST make a CW assumption, they will try to label these instances as one of the five observed activities resulting in high number of false positives.

One would expect that the threshold-based approach STT would perform significantly better than SVM and ST, since it theoretically has the capability of detecting unknown activities. However, as discussed in the previous section (Figure 4b), learning from a non-representative set of instances will lead to learning an biased decision boundary. Thus, using a threshold-based classifier provides only an insignificant improvement.

Amount of Known Classes

In the following, we fix the number of labeled instance $l = 15$ and analyze the behavior of mPUL under different number of observed classes m. As shown in Figure 7, mPUL performs well for all values of m, while the other classifiers perform poorly when only a few activity classes are observed. As discussed in the previous subsection, this is mainly caused by many unobserved activities being predicted as one of the observed activities.

Obviously, mPUL achieves the highest improvement in cases of small m. On the other hand, for $m = 19$, mPUL performs worse than the other classifiers. Note that $m = 19$ corresponds to the Closed-World condition, where we assume

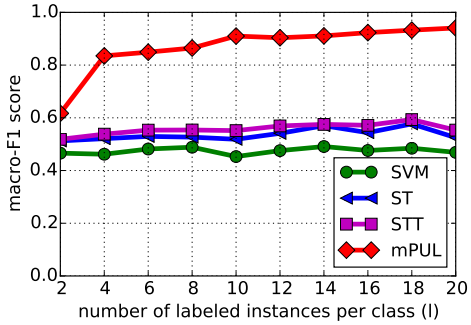


Figure 6: The training data contains instances of only five activity classes ($m = 5$). Thus, SVM, ST and STT do not improve their performance even when more labeled data is available. On the other hand, since mPUL does not rely on observing all activity classes, it performance improves with the increasing amount of labeled data.

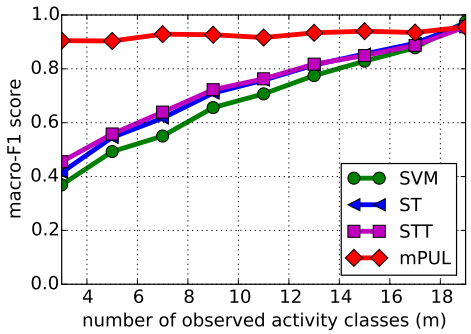


Figure 7: mPUL significantly outperforms other classifiers when only a small number of activity classes are observed in the training data ($m \ll M$). However, the other classifiers performs better if we can assume that we can collect labeled training data of all activities a user performs in her everyday life ($m = M$).

that the training data contains instances of all possible activity classes. Thus, if we can assume that we can obtain labeled data from all possible activities user perform in everyday life, the traditional supervised and semi-supervised techniques are preferable. However, in scenarios where the goal is to recognize a set of activities of interest (especially when this set is small), mPUL significantly outperforms the traditional AR models.

Avoiding False Positives

To better understand the performance of the classifiers, Figure 7 further shows the precision and recall of STT and mPUL for the previous experiment (SVM and ST exhibits a similar trend and therefore is omitted from the following discussion). For mPUL, both precision and recall remain at a similar level for any number of observed activity classes. For STT, recall remains the same, while the precision suffers with decreasing m. This is caused by the increase of false positives when only a few classes are observed in the training dataset.

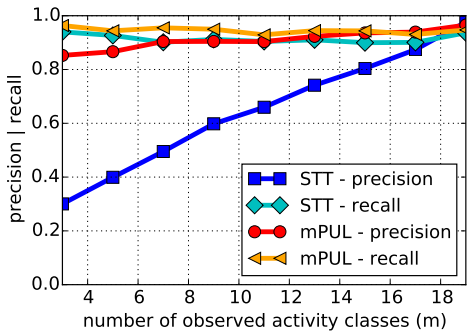


Figure 8: When a STT model is trained using only a small number of classes, the model will make many false positive predictions, resulting in low precision.

The above observation is more obvious when we study the misclassifications for individual activities. Suppose we are interested in recognizing when a user is exercising on a stepper [2]. To train a model, we ask the user to collect data for a few stepping sessions and additionally a few not-stepping sessions while performing activities such as sitting, standing, walking and running, totaling $m = 5$ activities. Additionally, we passively collect a large amount of unlabeled data while the user carries the sensing device during her everyday life, with no additional activity labels.

Table 1a shows the confusion matrix of STT model trained on the above-described dataset. For simplification, we merged classes other than stepping into "not-stepping" class. From the results we can observe that STT falsely classifies many not-stepping instances as stepping, resulting in high number of false positives. Table 2 shows the number of these false positives in each class. From the results, we observe that the false positives are mainly caused by the instances from classes not observed in the training dataset such as playing basketball, ascending stairs, etc.

Table 1b shows the results for the mPUL model. From the results we can observe that mPUL significantly reduces the false positives, while introducing some false negative predictions (classifying some stepping instances as non-stepping). These false negatives are caused by instances lying close the decision boundary and are by mPUL conservatively predicted as an unknown activity.

Evaluating AR with Imbalanced Activity Distribution

In the following, we evaluate mPUL on a dataset collected in a real-world scenario and study how mPUL deals with the additional challenge of imbalanced activity distribution. For this evaluation we use the TU Darmstadt dataset [8], which contains data from one user collected for a duration of seven days performing 33 different activities. The sensor data were collected from two sensors, one placed in the user's hip pocket and the other on the right wrist. We use the feature extraction method presented in the previous subsection using a sliding window of 10 seconds and 50% overlap. Activities in this scenario are highly imbalanced. As shown in Figure 9, the

		Predicted	
		Stepping	Not Stepping
Actual	Stepping	240	0
	Not Stepping	**205**	4115

(a) STT

		Predicted	
		Stepping	Not Stepping
Actual	Stepping	233	7
	Not Stepping	**0**	4316

(b) PUL

Table 1: Confusion matrices for predicting the stepping (exercising on a stepper) activity. mPUL significantly reduces the number of false positives.

Not-Stepping	Number of FP
playing basketball	52
ascending stairs	35
rowing	34
exercising on a cross trainer	27
descending stairs	9
moving around in an elevator	7
walking on a treadmill	5
jumping	2

Table 2: Number of False Positives (FP): Number of not-stepping activity instances, which were predicted by STT as stepping.

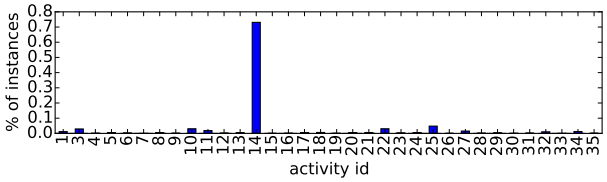


Figure 9: The TU Darmstadt dataset is highly imbalanced; the dataset is dominated by one frequent activity.

user was sitting at their desk (activity number 14) 73% of the time.

For the evaluation, we extend the traditional leave-one-day-out cross validation in the following manner (as illustrated in Figure 10). In each iteration, we use six days of data for training and one day for testing. In the training dataset, we use labels collected during one day, while keeping the remaining five days unlabeled.

The imbalanced activity distribution motivates us to use macro-F1 score as the evaluation metric instead of accuracy, which has been commonly used in existing AR work [8, 10]. Using the above leave-one-day-out cross validation method, SVM achieves an accuracy of 0.74 but a macro-F1 score of only 0.17. The high accuracy might lead one to believe that the AR model is performing well. However, when looking at the classifier's performance for each activity class individ-

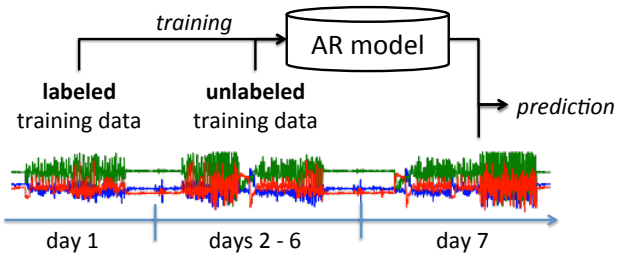


Figure 10: In each cross validation iteration, we use one day of labeled data and five days of unlabeled data to train a model, which is then evaluated on the remaining one day.

ually as shown in Figure 11, we can observe that the model performs well only on the small set of more frequent activities, while performing poorly on the others. Due to this fact, the macro-F1 is low.

In imbalanced problems, accuracy typically evaluates the strength of the classifier with respect to the frequent classes, whereas macro-F1 score weighs all classes equally. Since we are interested in the overall performance of the AR system (including its performance of recognizing infrequent activities), macro-F1 is more informative evaluation metric.

The TU Darmstadt dataset consists of labeled data of 33 activities. Collecting such training data is very labor-some and in many scenarios not necessary as one might be interested in only recognizing a small subset of activities of interest. In the following, we study the case when only a subset of activities is actually annotated.

We first sort the activities based on their count of occurrences. We select m activities with highest number of occurrences and discard labels of all other activities (thus making them unlabeled). Thus, the training data contains labeled instances of m activities from one day, unlabeled data of unselected activities from the same day and additional unlabeled data from the other five days. We train STT and mPUL classifiers from this training dataset and test them on the dataset of the remaining one day. Figure 12 shows the difference in macro-F1 between mPUL and STT. For small m, mPUL achieved over 0.2 macro-F1 score improvement, while for large m, mPUL achieves a comparable or even worse performance than STT. These results are directly correlated with the number of instances belonging to the unknown class in the test data. For $m = 3$, only the three most frequent activities are observed and therefore, instances of the remaining 30 activities should be predicted as an unknown class. Instances of these 30 remaining activities represent approximately 25% of the test dataset. Due to the high number of instances belonging to the unknown class, mPUL significantly outperforms STT, since it can better detect unknown activities. On the other hand, once more activities are observed in the training data (high m), the number of instances belonging to the unknown class decreases and the advantage of mPUL becomes negligible. Furthermore, since mPUL is designed to operate under the assumption that there are activities of unknown classes, it will conservatively assign uncertain instances to the unknown

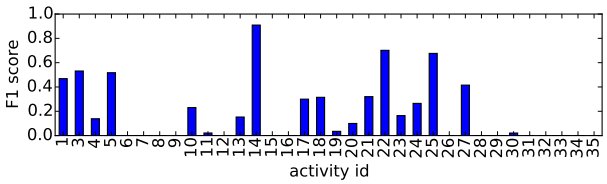


Figure 11: Through the F1 score we observe that a traditional classifier is optimized towards achieving good performance on more frequently occurring activities.

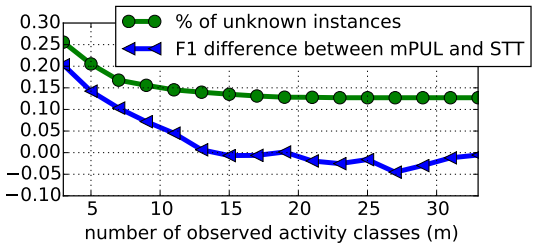


Figure 12: When only a small number of activities are observed, a large amount instances belong to unknown classes (e.g., for $m = 3$, 25% of test instances belong to the unknown class). In such cases, mPUL is by 0.2 macro F1-score better than STT, since it can effectively detect instances of these unknown classes.

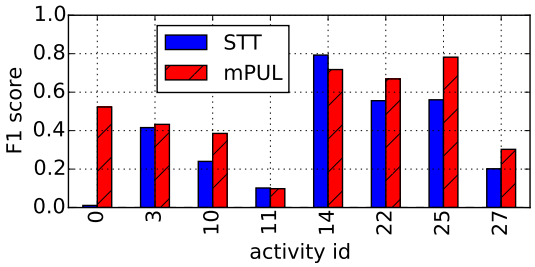


Figure 13: STT outperforms mPUL on the most frequent activity number 14, while mPUL achieves a comparable or better result on the remaining activity classes, especially on detecting the unknown class (represented as 0).

class. This behavior can, however, degrade the performance of the AR system for high m.

Figure 13 shows the F1 score of each individual activity for $m = 7$ (0 represents the unknown class). First we observe that mPUL performs worse than the STT for the most frequent activity number 14, while achieving a comparable or better result on all the other activities. This can be explained by the fact the traditional classifiers (including STT) are optimized towards increasing accuracy, i.e. increasing the performance for the most frequent activities.

Since activity 14 appears 73% of the time in our dataset, having labeled training data of this activity class will help the system avoid making misclassifications. However, consider the case, when the activity 14 is not among the activities of

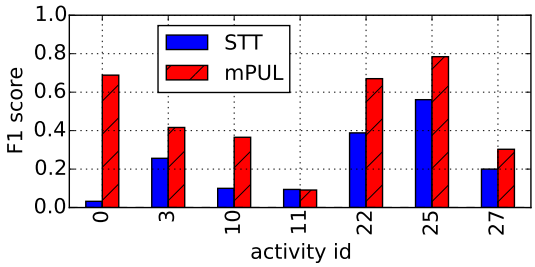


Figure 14: By not observing the frequent activity number 14 in the training dataset, we observe a significant degradation of STT. On the other hand, the performance of mPUL remains approximately the same, since it can better detect the unknown class and therefore avoid making false positive prediction on the known classes.

interest. Thus, the training dataset does not have any labeled instances of this class. Figure 14 shows the F1 score for a similar configuration as above, except excluding the activity 14 from the training dataset. As we can observe the F1 scores of mPUL remained approximately the same. However, STT performance is significantly degraded. The reason is that the system does not have any information about the activity 14. Ideally, the system should assign instances of activity 14 to the unknown class. However, STT performs poorly at detecting instances of the unknown classes. Thus, most of these instances will be incorrect assigned to one of the known activities. This results in a large number of false positives especially due to the high number of occurrences of activity 14.

DISCUSSION

When to use or not use mPUL: mPUL is suitable for scenarios where we want to recognize a small set of activities of interest and have labeled data only for these activities. mPUL achieves a significant improvement over the traditional approaches especially in cases when the activities of interest are infrequent. In such cases, there are a large number of instances of unknown activities, which are prone to be falsely predicted as the activities of interest. On the other hand, mPUL does not bring any improvement in cases when we can collect labeled data of all activities a user performs in her everyday life.

Collecting unlabeled data: In this work, we assume that the training data contains unlabeled instances of most activity classes. One of the ways for obtaining such unlabeled dataset is to build an AR model from the initial training dataset and deploy it in the real world. Once the system is in use, we can collect additional unlabeled data while the user carries the sensing device in her everyday life. Even though collecting unlabeled sensor readings does not require any human labeling effort, one still needs to consider the cost with respect to the energy consumption of the sensing device. This can be minimized by collecting the sensor readings only when the AR system is used to recognize activities. In such cases, the system needs to capture sensor readings to make the prediction. Instead of discarding these sensor readings after the prediction, we keep them for future update of the AR system.

Computational complexity: Training a mPUL model involves training m binary classifiers and estimating the scaling constant c. The computational complexity mainly depends on the efficiency of training and prediction of the base classifiers. However, the training and prediction complexity of mPUL is comparable to models such as Multi-class SVM, which also uses the one-vs-other strategy. The main difference is that mPUL uses the additional unlabeled data for training. In the case that the unlabeled pool is large, additional questions arise such as: how much unlabeled data is sufficient and can we identify and sub-select relevant unlabeled instances for training to reduce the computation overhead? In future work, we will conduct additional experiments to address these open questions.

Recognizing infrequent activities: As shown in this work, recognizing infrequent activities is a challenging task. First, the existing classifiers are optimized towards increasing accuracy, thus, improving the recognition performance on frequent activities. Secondly, we typically do not have a lot of labeled training data for infrequent activities. mPUL can help to a certain degree to improve an AR performance by avoiding false positives (i.e., avoid predicting unknown activities as one of activities of interest). However, further research is needed to combine mPUL with learning paradigms such as attribute-based learning [15] to overcome the challenges associated with learning from a small number of training instances.

Active Learning with mPUL: The proposed mPUL uses unlabeled data to recognize whether an incoming instance belongs to a known or an unknown activity. This capability can be extended by integrating active learning to ask the user to provide labels for the unknown activities. This is significantly different from the existing active learning work, which mainly uses the unlabeled data for finding uncertain instances lying close the decision boundary of the known activity classes. In the future work, we will explore the integration of mPUL with active learning to accelerate the discovery and learning of unknown activity classes.

CONCLUSION

In this work, we show that traditional AR systems achieve poor performance in many Open-World scenarios. Errors are mainly caused when handling instances from classes for which no labeled training data is available. Our proposed mPUL approach addresses this issue by assuming that the training data contains unlabeled data of these unobserved classes. By leveraging the unlabeled instances using mPUL, we showed that we can improve the performance by significantly reducing the number of false positive predictions cause by unknown classes.

ACKNOWLEDGEMENT

This research is supported in part by the National Science Foundation under award 1346066: SCH: INT: Collaborative Research: FITTLE+: Theory and Models for Smartphone Ecological Momentary Intervention.

REFERENCES

1. Abdullah, S., Lane, N. D., and Choudhury, T. Towards population scale activity recognition: A framework for handling data diversity. In *AAAI* (2012).
2. Barshan, B., and Yüksek, M. C. Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units. *The Computer Journal* (2013), bxt075.
3. Chavarriaga, R., Sagha, H., Calatroni, A., Digumarti, S. T., Tröster, G., Millán, J. d. R., and Roggen, D. The opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters 34*, 15 (2013), 2033–2042.
4. Cheng, H.-T., Sun, F.-T., Griss, M., Davis, P., Li, J., and You, D. Nuactiv: Recognizing unseen new activities using semantic attribute-based learning. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, ACM (2013), 361–374.
5. Elkan, C., and Noto, K. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM (2008), 213–220.
6. Guan, D., Yuan, W., Lee, Y.-K., Gavrilov, A., and Lee, S. Activity recognition based on semi-supervised learning. In *13th IEEE Internationa Embedded and Real-Time Computing Systems and Applications, 2007. RTCSA 2007l Conference on*, IEEE (2007), 469–475.
7. Hachiya, H., Sugiyama, M., and Ueda, N. Importance-weighted least-squares probabilistic classifier for covariate shift adaptation with application to human activity recognition. *Neurocomputing 80* (2012), 93–101.
8. Huynh, T., Fritz, M., and Schiele, B. Discovery of activity patterns using topic models. In *Proceedings of the 10th international conference on Ubiquitous computing*, ACM (2008), 10–19.
9. Khan, S. S., and Madden, M. G. A survey of recent trends in one class classification. In *Artificial Intelligence and Cognitive Science*. Springer, 2010, 188–197.
10. Lara, O. D., and Labrador, M. A. A survey on human activity recognition using wearable sensors. *Communications Surveys & Tutorials, IEEE 15*, 3 (2013), 1192–1209.
11. Li, X., and Liu, B. Learning to classify texts using positive and unlabeled data. In *IJCAI*, vol. 3 (2003), 587–592.
12. Li, X.-L., Liu, B., and Ng, S.-K. Negative training data can be harmful to text classification. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, Association for Computational Linguistics (2010), 218–228.
13. Lockhart, J. W., and Weiss, G. M. The benefits of personalized smartphone-based activity recognition models. In *Proc. SIAM International Conference on Data Mining (SDM)* (2014), 614–622.
14. Longstaff, B., Reddy, S., and Estrin, D. Improving activity classification for health applications on mobile devices using active and semi-supervised learning. In *4th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2010*, IEEE (2010), 1–7.
15. Nguyen, L. T., Zeng, M., Tague, P., and Zhang, J. Recognizing new activities with limited training data. In *International Symposium on Wearable Computers (ISWC)*, IEEE (2015).
16. Nguyen, L. T., Zeng, M., Tague, P., and Zhang, J. Superad: Supervised activity discovery. In *International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, ACM (2015).
17. Stiefmeier, T., Roggen, D., Ogris, G., Lukowicz, P., and Tröster, G. Wearable activity tracking in car manufacturing. *IEEE Pervasive Computing 7*, 2 (2008), 42–50.
18. Stikic, M., Larlus, D., and Schiele, B. Multi-graph based semi-supervised learning for activity recognition. In *International Symposium on Wearable Computers, 2009. ISWC'09*, IEEE (2009), 85–92.
19. Stikic, M., Van Laerhoven, K., and Schiele, B. Exploring semi-supervised and active learning for activity recognition. In *12th IEEE International Symposium on Wearable Computers, 2008. ISWC 2008*, IEEE (2008), 81–88.
20. Ward, J. A., Lukowicz, P., and Gellersen, H. W. Performance metrics for activity recognition. *ACM Transactions on Intelligent Systems and Technology (TIST) 2*, 1 (2011), 6.
21. Weston, J., and Watkins, C. Multi-class support vector machines. Tech. rep., Citeseer, 1998.
22. Yang, Y. A study of thresholding strategies for text categorization. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM (2001), 137–145.
23. Yin, J., Yang, Q., and Pan, J. J. Sensor-based abnormal human-activity detection. *Knowledge and Data Engineering, IEEE Transactions on 20*, 8 (2008), 1082–1090.
24. Yu, H., Han, J., and Chang, K. C.-C. Pebl: positive example based learning for web page classification using svm. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM (2002), 239–248.
25. Zhang, T., Wang, J., Xu, L., and Liu, P. Fall detection by wearable sensor and one-class svm algorithm. In *Intelligent Computing in Signal Processing and Pattern Recognition*. Springer, 2006, 858–863.