Autonomous Sensor-Context Learning in Dynamic Human-Centered Internet-of-Things Environments

Seyed Ali Rokni, and Hassan Ghasemzadeh
Embedded and Pervasive Systems Laboratory (EPSL)
School of Electrical Engineering and Computer Science
Washington State University, Pullman, WA 99164–2752
{alirokni, hassan}@eecs.wsu.edu

ABSTRACT

Human-centered Internet-of-Things (IoT) applications utilize computational algorithms such as machine learning and signal processing techniques to infer knowledge about important events such as physical activities and medical complications. The inference is typically based on data collected with wearable sensors or those embedded in the environment. A major obstacle in large-scale utilization of these systems is that the computational algorithms cannot be shared between users or reused in contexts different than the setting in which the training data are collected. For example, an activity recognition algorithm trained for a wrist-band sensor cannot be used on a smartphone worn on the waist. We propose an approach for automatic detection of physical sensor-contexts (e.g., on-body sensor location) without need for collecting new labeled training data. Our techniques enable system designers and end-users to share and reuse computational algorithms that are trained under different contexts and data collection settings. We develop a framework to autonomously identify sensor-context. We propose a gating function to automatically activate the most accurate computational algorithm among a set of shared expert models. Our analysis based on real data collected with human subjects while performing 12 physical activities demonstrate that the accuracy of our multi-view learning is only 7.9% less than the experimental upper bound for activity recognition using a dynamic sensor constantly migrating from one on-body location to another. We also compare our approach with several mixture-of-experts models and transfer learning techniques and demonstrate that our approach outperforms algorithms in both categories.

1. INTRODUCTION

Many emerging Internet of Things (IoT) applications, from medical monitoring and home automation to automotive engineering and automatic security surveillance, involve human subjects where humans and things operate synergistically towards satisfying objectives of the application [1–4]. At the heart of these human-centered IoT systems is *human monitoring* where physiological and behavioral context of the user are assessed using wearable sensors or those deployed in the environment. Typically, sensors acquire physical measurements, use computational algorithms such as machine learning and signal processing techniques for local data processing and information extraction, and communicate the results to their outside world, for example, the cloud.

Computational algorithms offer core intelligence of these systems by allowing for continuous and real-time extraction of clinically important information from sensor data. The generalizability of these algorithms, however, is a challenge due to the dynamically changing configuration of the system. In fact, the algorithms need to be reconfigured (i.e., retrained) upon any changes in configuration of the system, such as displacement/ misplacement/ misorientation of the sensors. Practically, development of the computational algorithms requires algorithm training using sufficiently large amount of labeled training data, a process that is deemed time consuming, labor-intensive, expensive, and a major barrier to personalized and precision medicine [5]. Therefore, it is imperative to develop new methodologies for sharing already trained computational algorithms in order to prevent the costly process of collecting labeled training data for every sensor-context. The development of multi-view learning solutions that enable transfer of the machine learning knowledge from previously trained models to new physical contexts in human-centered IoT applications is an entirely new research area, which has remained virtually unexplored by the community. Our sensor-context learning approach presented in this paper contributes to development of generalizable and robust machine learning algorithms operating with high accuracy even in previously unseen context settings, such as utilization of the system by a new user or wearing the sensors on body-locations different than the data collection setting.

Our pilot application in this study is activity recognition. Recent findings [6–8] suggest that one can develop computational algorithms that compensate for context dynamics (e.g., sensor displacement, misplacement, mis-orientation). These algorithms, however, are accurate only if we collect sufficient labeled training data for all possible sensor-contexts. In fact, an implicit assumption in development of current computational algorithms for human-centered IoT applications is that the training and future data are in the same feature space and have the same distribution [9]. Therefore, most algorithms require significant amounts of training data for each network configuration or sensor-context.

In this paper, we take first steps in developing automatic and real-time training of sensor-context detection without labeled training data. Specifically, we focus on cases where multiple context-specific algorithms (i.e., 'expert models') are shared for use in a dynamic view where the sensor is worn/used on various body locations each representing one sensor-context. We propose an approach for learning a gating function for choosing the most accurate expert model based on the observed sensor data. Our ap-

ICCAD'16, Nov 07 - 10 2016, Austin, TX.

ACM ISBN 978-1-4503-2138-9.

DOI: 10.1145/1235

proach, called *Synchronous Sensor-Context Learning* (SSCL), first generates and automatically labels a training datasets by examining observations of the dynamic sensor and associating those observations with synchronously sampled observations of a static sensor node. This training dataset is then used to learn the gating function for expert model activation.

Our multi-view learning approach presented in this paper is a novel method for sharing activity recognition capabilities of several sensors, with already training activity recognition classifiers, for use by a dynamic sensor, which does not have any previously trained activity recognition model. Our approach allows to transfer machine learning knowledge from an existing sensor, called *static view*, to a new sensor, called *dynamic view*, and combine the knowledge with already shared capabilities and develop an extensive model for the dynamic view.

2. PROBLEM STATEMENT

Figure 1 shows how several computational models with different context settings share their activity recognition (AR) knowledge to build an integrated model for sensor-context detection without need to collecting labeled training data. Initially, as shown in Figure 1a, four activity recognition classifiers each with a trained model based on data from a static sensor attached to a fixed on-body location share their model for use with a dynamically moving sensor (e.g., a smartphone is used on various locations on the body). In this example, each shared model is limited to the exact on-body location for which the activity recognition model has been trained (e.g., 'arm', 'ankle', 'pocket', 'right wrist'). At the end of the model sharing phase, the dynamic/movable sensor has a database of different models each limited to a specific physical context setting. In the next phase, called *gating function training*, the dynamic sensor uses its local sensor observations and those of an assistive static sensor (e.g., 'left wrist') to learn a model for sensor-context detection. Throughout this paper, we refer to the observations made by the dynamic sensor as 'dynamic view' and those generated by the assistive static sensor as 'static view'. We note that the static sensor is a sensor attached to a fixed location on the body and utilized only during gating function training phase. This sensor is eliminated from the network once the gating function in trained.

The dynamic sensor does not have an inference model (e.g., sensor localization algorithm) to detect its current context/setting (e.g., on-body sensor localization). As discussed previously, on-body sensor localization requires collecting labeled training and developing a machine learning algorithm that detects the on-body location of the wearable sensor. We note that our goal is to detect dynamic context of the sensor without collecting labeled training data. As soon as the dynamic context is detected, the sensor will choose the corresponding shared model for activity recognition. Our sensor-context learning approach captures sensor data in both 'static view' and 'dynamic view' simultaneously, compares predictions of different models and constructs a model selector machine to partition the observation domain of the dynamic sensor corresponding to its appropriate model Figure 1c.

2.1 Synchronous Sensor-Context Learning

An observation X_i made by a wearable sensor at time 'i' can be represented as an D-dimensional feature vector, $X_i = \{f_{i1}, f_{i2}, \ldots, f_{iD}\}$. Each feature is computed from a given time window and a marginal probability distribution over all possible feature values. The activity recognition task is composed of a label space $\mathcal{A}$=$\{a_1, a_2, \ldots, a_m\}$ consisting of the set of labels for activities of interest, and a conditional probability distribution $P(\mathcal{A}|X_i)$ which is the probability of assigning a label $a_j \in \mathcal{A}$ given an observed instance X_i.

Although a trained static sensor can detect some activities, its observations are limited to the body segment on which the sensor is worn. Therefore, some activities are not recognizable in one sensor's view point. For example, a sensor worn on 'ankle', although expert in detecting lower-body activities such as *walking*, cannot detect activities such as 'eating' that involves upper-body motions. Furthermore, when the context of the sensor changes (e.g., on-body sensor location), the activity recognition model fails to accurately classify movements. This potentially limits scalability of human-centered IoT systems because users are constrained to wear the sensors only on predefined locations on the body or use them according the context or experimental protocol with which the data collection and activity recognition training has taken place. To extend flexibility of the user and the ability to detect activities of different body segments, it is reasonable to have a dynamic sensor that user can wear around the body as desired. Our framework in the paper intends to automatically detect sensor context and activate machine learning model appropriate for its current setting.

We note that advances in embedded sensor design and high wearable electronics allow end-users to utilize new movable wearables such as smart phones. In such a realistic scenario, the dynamic sensor, however, is able to gather a repository of models of different contexts/locations, it has not been trained to detect its context. The general goal of our study is to develop an autonomous learning algorithm to train sensor-context detection algorithms by devising a transfer learning approach. Without loss of generality, we assume that the static view consists of a single sensor, in our formulation of this sensor-context learning problem.

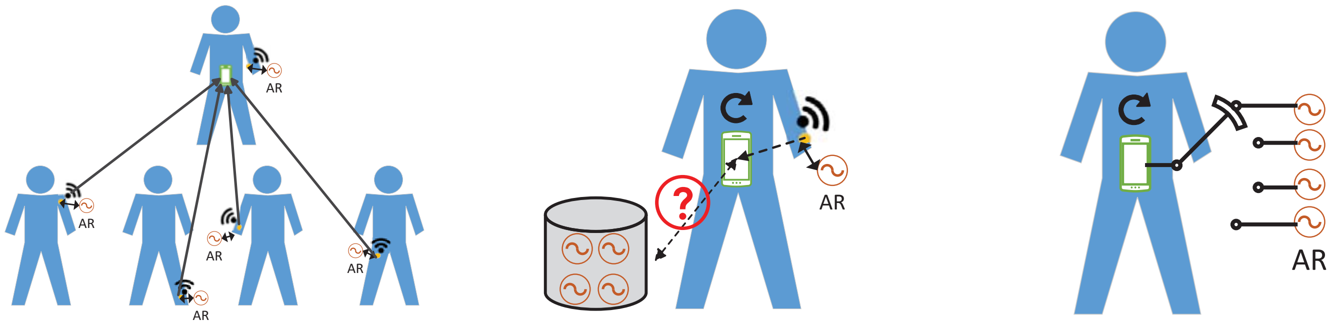
PROBLEM 1 (SSCL). *Let C = $\{c_1, \ldots, c_K\}$ be a set of K possible sensor-contexts each contributing an expert model AR_i thus forming a set of experts $AR = \{AR_1, \ldots, AR_K\}$ as their corresponding activity recognition expert models. Let $C_d \subset C$ be a set of possible placements of the dynamic sensor. Moreover, let $\mathcal{A}$=$\{a_1, a_2, \ldots, a_m\}$ be a set of m activities/labels that the system aims to recognize, and $\mathcal{X}$=$\{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N\}$ a set of N observations made by the dynamic sensor when used in any of the contexts $c \in C_d$. The Synchronous Sensor-Context Learning (SSCL) problem is to train a sensor-context detector to accurately detects context of the dynamic sensor and activate an expert model such that the activity recognition error is minimized.*

Each expert model has a limited ability in detecting activities depending on the sensor-context (i.e., physical placement of the sensor on the body). Now, we define the expertise domain of an expert model.

DEFINITION 1 (EXPERTISE DOMAIN). *Let $\mathcal{X}$ be set of all observations of dynamic sensor. For expert model $AR_i \in AR$, the Expertise Domain is a subset $S \subset X$ where E_i, the error of misclassification model AR_i on S, is hugely less than other expert models. In other words $E_i \ll E_j$, $j \neq i$.*

When a sensor is moving, it enters and leaves the expertise domain of an expert model. That is, the domain of observation of dynamic sensor could divided into parts based on expertise of expert models in detecting activities. To decide among expertise domain of we should train a gating function to accurately select corresponding expert model.

DEFINITION 2 (GATING FUNCTION). *The Gating Function is a deciding function g which gets current observation X_t as input and assigns probability p_i to expert model AR_i based on its expertise on X_t. Note $\sum_i p_i = 1$.*



(a) Model Sharing Phase

(b) gating Function Training Phase

(c) Context Detection Phase

Figure 1 (a) Several networks with static computational models share their model for utilization by a dynamically moving sensor. (b) Dynamically moving sensor utilizes the shared models, its local sensor observations, and synchronous observations made by a local static sensor to learn a gating function for detecting the correct sensor-context (e.g., sensor location). (c) The trained gating function is used to switch between the shared models based on the detected sensor-context.

2.2 Mixture-of-Experts for Context Learning

We aim to partition the observations into subsets by not looking for observations that are similar but by exploring them to have a relationship between observations and their predicted labels that can be well-modeled by one of the expert models. Therefore, the problem cannot be modeled by a simple clustering method. The problem of partitioning based on observation-label relationship could be modeled using mixture-of-experts which encourages specialization of expert models. As Figure 3 demonstrates a simple example of mixture-of-experts with two different experts and two class labels, where Expert 1 is responsible for detecting instances of above gating line and Expert 2 is performing on instances of below gating line. This example shows why clustering algorithms are not working. In this situation, when expertise domains are disjoint, decision fusion methods such as averaging or majority voting does not work, because one model is right with high probability and all other models are wrong with high probability.

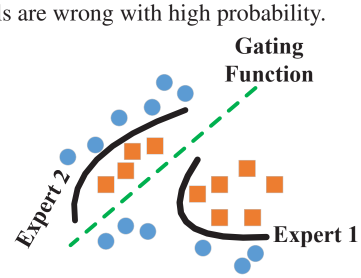


Figure 2 An example of Mixture of Expert with two class labels (blue circles and orange rectangles) and two different detection models (Expert 1 and Expert 2). Here, each expert is responsible for separation of instances of one side of gating function.

Let g be a gating function which assign a probability p_i to expert AR_i based on the observation X_t at time t. Decision of AR_i on X_t is a probability vector P over all possible activities such that $P_j = P(Y_t = j)$. Therefore, the probability of each activity label j could be computed as

$$
\begin{aligned}
P(Y_t = j|X_t, g) &= \sum_{i=1}^{K} P(Y_t, AR_i|X_t, g) \\
&= \sum_{i=1}^{K} g(AR_i|X_t)P(Y_t = j|X_t, AR_i) \qquad (1) \\
&= \sum_{i=1}^{K} p_i \; P(Y_t = j|X_t, AR_i)
\end{aligned}
$$

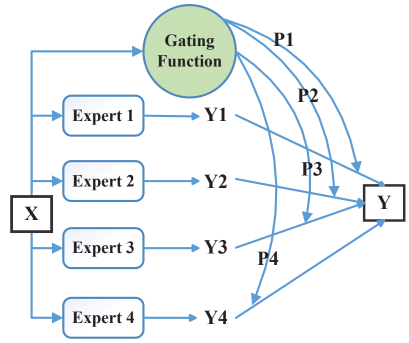


Figure 3 An example of Mixture of Expert with four Experts. All experts get X observation of dynamic view and decide about the activity. Simultaneously, gating function assign a probability P_i to output of each experts and the final output computed using combination of all experts.

It means that we first pick the context expert model AR_i with probability p_i based on the observation X_t (i.e $g(AR_i|X_t)$). Then we compute the probability of label j w.r.t AR_i and current observation: $P(Y_t = j|X_t, AR_i)$.

Therefore, the Mixture of Expert training algorithm tries to maximize likelihood of probability Equation 1 on the training data to learn the parameters of gating function g. In this situation, Expectation Maximization (EM) [9] is used as an iterative approach for finding maximum likelihood of a probability model when the problem consist of some observed random variables (e.g. activity label) and some random variables that are hidden. But here, the challenge problem is that there is no training data available for dynamic sensor. The only source of knowledge in dynamic view could be noisy predictions made by static sensor. In the next section we will introduce our approach of training gating function using Teacher/Learner transfer learning.

2.3 Training of the Gating Function

In our synchronous learning approach, the static sensor acts as a 'Teacher' and sends its prediction of the current activity as a vector of probabilities over activity labels to the dynamic sensor in real-time. At the same time, the dynamic sensor, also called 'Learner', queries all experts models and receives their prediction vector on observation of dynamic sensor. Then, dynamic sensor compares the prediction vector of 'Teacher' with prediction of all its expert models and learns from the information provided by 'Teacher'. Finally, it picks the closest decision as the candidate of sensor-context for that particular observation. Overtime it gathers that training data for gating function.

Algorithm 1 Synchronous Sensor-Context Learning (Static View)

while (all activity have been observed) **do**
  'static' performs activity recognition on X_t at time t.
  'static' assigns a probability to each possible activities.
  'static' sends probability vector P_t to the *dynamic* view.

Algorithm 2 Synchronous Sensor-Context Learning (Dynamic View)

Initialize sensor-context training data
while (static sensor sends prediction) **do**
  'dynamic' queries predictions of all expert models using its observation X'_t
  each expert e_i makes a prediction as a probability vector P'_{it}
  'dynamic' computes distance between each P'_{it} and P_t.
  'dynamic' assigns X'_t with index of minimum expert with closest prediction to 'static'
  'dynamic' adds X'_t and its label to sensor-context training data.
construct a sensor-context classifier based on training data

Algorithm 1 and 2 show our multi-view sensor-context learning approach for training the gating function. Figure 4 illustrates an example of this training algorithm when there are four different expert models in the dynamic view. We assume that during training the sensors within the two views are worn on the body at the same time while the user performs physical activities. Because the static and dynamic sensors may have different clocks, the dynamic sensor needs to know to which observation each prediction vector corresponds. We resolve the problem of different clocks by first synchronizing static and dynamic sensors. At time t, the static sensor predicts the activity probability vector P_t of the current activity; and transmits (P_t, t) to the dynamic sensor via a wireless link. Simultaneously, dynamic sensor queries its expert models for predicting its current observation. The dynamic sensor selects the expert with the closest decision to P_t and labels senor-context of its current observation as the index of the closest expert. The dynamic sensor gathers training data of sensor-context detection model by accumulating observations and labels until sufficient training data are gathered and automatically labeled by our algorithm. When there is sufficient number of training data, the dynamic sensor constructs its sensor-context detection model to act as the gating function g. The process of constructing a gating function based on the devised training dataset is straightforward and consistent with the classical classifier training in the machine learning research. We note, however, that our multi-view sensor-context learning algorithms presented in this paper is independent of the type of the classifier used for training the gating function. As soon as the gating function is learned, the static sensor can be removed from the network and the dynamic sensor can detect activities independently.

2.3.1 *Measuring Decision Disagreement*

To compute disagreement of static and each expert in dynamic view, we need a metric of comparison among experts. In our experiments, we use *Euclidean* distance function between static sensor decision vector and decision vector of each expert model in dynamic view to compute the degree of disagreement. We note that the decision vector of the static sensor and any experts receives the probability of each activity label and therefore, have equal sizes. When P is the decision vector of static sensor and P' is decision vector of one expert in the dynamic view, we can compute their distances using the following equation.

$$\Delta(P, P') = \sqrt{\sum_{j=1}^{M} (P_j - P'_j)^2} \tag{2}$$

Using Euclidean we put more weight than Manhattan distance on larger differences in any dimensions.

2.3.2 *Handling Uncertainty*

As Figure 4 shows, static model is fed by observation of static sensor and simultaneously expert models of dynamic view decide on dynamic sensor observation. Using a distance function, context with minimum distance between its expert decision and static sensor decision is selected as current sensor-context. For example in Figure 4 using Manhattan distance, both Expert 1 and Expert 3 cause minimum distance 6 from source provided vector. In this tie situation, we insert two instances of current observation one with label 1 and another with label 3. By doing this multi-labeling, we don't lose any information. On the other hand, assuming the correct label was 3, other instances of class 1, will compute this wrong instance as the outlier or noise.

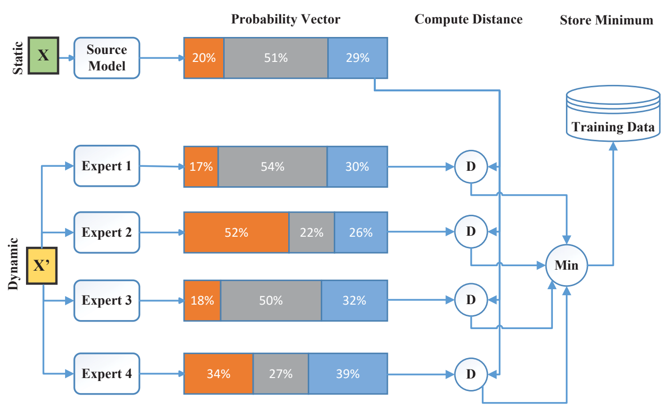


Figure 4 The process of training: When an activity happens, static model and all experts of dynamic model make their predictions. Then, distances of experts decisions and static model prediction compares and the index of closest expert is selected as sensor-context label of current dynamic observation

3. EXPERIMENTAL RESULT

In this section, we demonstrate the performance of our multi-view sensor-context learning algorithm using real data collected from 3 human subjects performing 12 different physical activities while wearing 5 wireless motion sensor nodes on their 'Left Wrist' (LW), 'Right Arm' (RA), 'Left Thigh' (LT), 'Right Ankle' (RA) and 'Waist' (Wa). Each sensor node had a 3-axis accelerometer and 2-axis gyroscope. The physical activities included *Stand to Sit* (St2S), *Sit to Stand* (S2St), *Sit to Lie* (S2L), *Lie to Sit* (L2S), *Jump* (Jmp), *Turn Clockwise* (TC), *Bend to Grasp* (B2G), *Step Backward* (SB), *Look Back* (LB), *Kneeling Right* (KR), *Rise from Kneeling* (RfK) and *Return from Looking back* (RfL). The sampling frequency was set to 50Hz and the subjects were asked to repeat each activity 10 times while the data were being collected wirelessly. The obtained dataset contained over $684,000$ samples of acceleration and angular velocity. The data were used to extract 10 statistical features from the individual sensor streams.

Table 1 Activity recognition accuracy of each sensor node with ground truth labels

Source/Static Node	Classification Accuracy
WA	90.83
LW	67.50
RA	83.33
LT	69.17
RA	62.50
Average	74.67

3.1 Performance of Source/Static Sensor

Before analyzing the performance of our SSCL algorithm, we first evaluate the accuracy of an activity recognition model trained in the static view. Our goal was to assess the robustness of a single static sensor node for recognizing the 12 experimental activities. We note that the static sensor is used as a source of knowledge for training of the gating function. Thus, it is important to gauge the level of robustness of the predictions made by a static sensor. For this purpose, we trained a k-Nearest-Neighbor (kNN) classifier using data collected and labeled in our experiments for each one of the 5 sensor nodes. As shown in Table 1 the accuracy of the activity recognition model is less than 75% on average. The maximum accuracy belongs to the 'waist' sensor and the 'ankle' sensor achieves the minimum accuracy (i.e., 62.5%) among all other sensors.

These results suggest that a reasonable number of activity instance are not reliably distinguishable by the static sensor node. For instance, about 37.5% of the instances are mis-classified by 'right ankle' sensor. Thus, solely relying on predictions of the static sensor while learning the gaiting function will result in a poor model. This confirms our hypothesis that a standard Expectation Maximization (EM) approach for learning the gaiting function (i.e., finding maximum likelihood for mixture-of-experts) will not work in our multi-view learning. These methods assume the source of knowledge (i.e., static sensor in this case) is perfect; in practice these approaches will result in over-fitting on the static sensor noise. As a result, we devise a multi-sampling-based learning approach for training the gating function resulting in eliminating the impact of the noise in our gating classifier for sensor-context detection.

3.2 Comparative Evaluation Method

Our approach to sensor-context learning is a hybrid method that combines mixture-of-experts and transfer learning in a unified framework. To the best of our knowledge, there is no prior research that addresses the problem of mixture model for wearable sensors. Thus, we decided to develop several baseline and intuitive mixture-of-experts-based sensor-context detection algorithms for comparison purposes. To this end, we implemented two algorithms, namely *Random* and *Majority Voting*. In the random approach, we randomly choose one of the experts for activity recognition. We expect that the random method provides minimum accuracy and therefore can be referred to as experimental lower bound. The other natural method of dealing with the problem of mixture-of-experts is voting. In majority voting, we query all experts and perform activity recognition based on majority of votes of the experts. In addition to these two mixture-of-experts-based methods, we compare our approach with the experimental upper bound obtained using ground truth sensor-context labels gathered during our data collection (i.e., assuming that the location of the sensor is known a priori, we use the correct activity recognition classifier).

We can also compare our method with a limited number of transfer learning algorithms proposed for wearable computing. In these approaches, however, the only source of knowledge will be the static sensor. Unfortunately, research in the area of transfer learning for wearables is new. To the best of our knowledge, there exist only two of such algorithms, namely *Naive* and *System-Supervised*, suggested in [10], which are applicable to the synchronous teacher/learner approach studied in this paper, although they do not incorporate the knowledge provided by external/shared classifiers. In other words, these approaches are designed and analyzed when the location of the sensor does not change. Calatroni et. al proposed the Naive approach as reusing the 'source' (i.e., static view in this case) classifier in 'target' (i.e., dynamic view in this case). They emphasized that this method only works when 'source' and 'target' are completely similar (e.g., sensors are co-located on the body and are homogeneous). The *System-Supervised* method refers to the case where 'target' labels its observations based on labels predicted by 'source'.

Our analysis compares the accuracy of the activity recognition classifier using all these algorithms (i.e., random, majority voting, naive, system-supervised, and upper bound) in the next section.

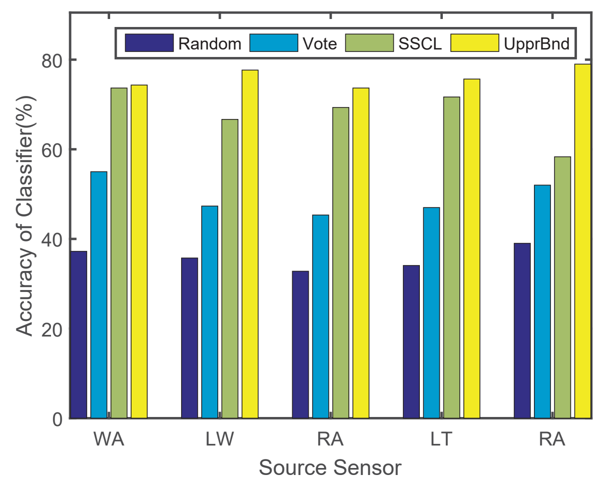


Figure 5 Activity recognition accuracy for four mixture-of-experts approaches under comparison including randomly selected expert (Random), majority voiding (Vote), our approach (SSCL), and experimental upper bound (UpprBnd)

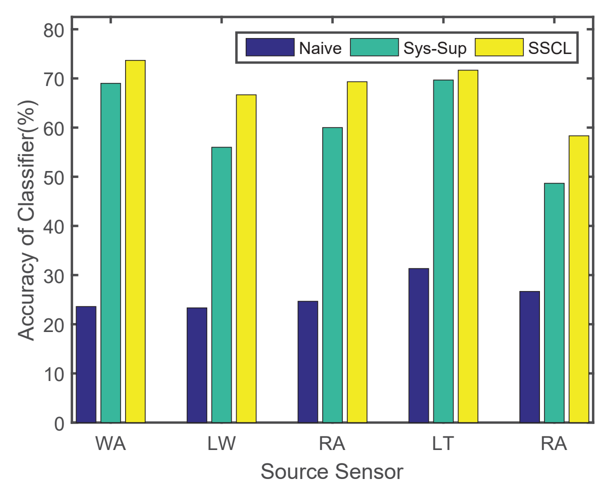


Figure 6 Activity recognition accuracy for three transfer learning approaches under comparison including Naive, the System-Supervised (Sys-Sup) and our approach (SSCL)

3.3 Comparative Analysis

We studied different scenarios where the static sensor could be any of the five sensors, namely ‘Left Wrist’ (LW), ‘Right Arm’ (RA), ‘Left Thigh’ (LT), ‘Right Ankle’ (RA) and ‘Waist’ (Wa). For each scenario, we considered each one of the other four locations as possible locations of the dynamic sensor. In other word, while the static sensor is fixed in one of the five locations, the dynamic sensor is continuously relocated among all other four locations. For example, when ‘waist’ was considered as ’static view’, we used ‘left wrist’, ‘right arm’, ‘left thigh’, and ‘right angle’ as on-body locations of the dynamic sensor.

As shown in Figure 5 and Figure 6, the accuracy of the activity recognition algorithm based on our automatic sensor-context detection approach (SSCL) ranges from 58.33% for ‘right arm’ to 78.33% for ‘waist’. On average, SSCL-based activity recognition achieves 68.4% accuracy. This accuracy is 42.1% and 18.6% higher than the accuracy of ‘Random’ and ‘Majority Voting’ respectively. Furthermore, SSCL-based activity recognition is only 7.9% less accurate compared to the activity recognition model built using ground truth labels (i.e., activity recognition model trained with known sensor-context based on video recordings of the experiments).

Compared to ‘system supervised’, which is a transfer-learning-based method, SSCL-based activity recognition achieves 7.3% higher accuracy in detecting the 12 physical activities. Furthermore, our approach outperforms by far the ‘naive’ approach. As shown in Figure 6, SSCL-based activity recognition performs 48.9% better compared to ’naive’.

4. DISCUSSION AND FUTURE WORK

In this paper, we introduce a method for reusing already trained machine learning model with an autonomous sensor-context detection algorithm that identifies the best expert model with no labeled training data regarding the sensor-context. Our work, which combines a new method of mixture-of-experts learning and transfer learning is different from prior research. Especially, Teacher/ Learner (TL) transfer learning [11] has being used when there is no direct access to training data. When the location of the target sensor is fixed, several studies [10,12,13] apply the teacher/ learner model to develop an opportunistic system capable of performing reliable activity recognition. Authors in [10] showed that by synchronizing source sensor and target sensor, the source can provide labels of future activities to the target sensor. However, the assumption of fixed position is not realistic for dynamically relocating sensors such as smartphones. To the best of our knowledge, our study is the first effort for automatic learning of sensor-context in dynamically changing wearable sensor environments.

Our study is a first step towards designing a platform for knowledge sharing among wearables that are computationally autonomous and can automatically learn machine learning algorithms without need for any new labeled training data; consequently, the accuracy of our approach is bounded to the accuracy of shared models. Dynamic attributes of sensor-context are not limited to real-time change of the sensor location. A sensor can be be misplaced, displaced, upgraded, or replaced. Our ongoing research involves development of multi-view learning algorithms that address dynamically evolving context of the sensors in human-centered monitoring applications.

In this study, we only focused on activity recognition applications using homogeneous sensor. In the future, we plan to investigate the effectiveness of our approach in a network with heterogeneous sensors with different modalities on a broader range of applications.

5. CONCLUSION

As wearable sensors are becoming more prevalent, their function becomes more complex and they operate in highly dynamic environments. Machine learning algorithms for these sensors cannot be designed only for one specific setting. To address the dynamic nature of wearable sensors, we proposed a multi-view learning approach that uses the knowledge of existing sensors to adapt with on-body sensor relocation. We used activity recognition task as our pilot application and develop an framework that enables sharing of the machine learning algorithms across different sensor contexts. We introduced a multi-view earning approach to learn computational algorithms in dynamic settings without any need for labeled training data and by using computational algorithms trained with various sensor contexts. We focus on on-body location of the sensors as pilot sensor-context in our platform. Our experiments show that we can combine knowledge of a static sensor with shared computational models to train an extensive model for dynamically relocating on-body sensor. Our results demonstrate that our multi-view learning approach achieves an activity recognition accuracy that is only 7.9% less than the upper bound performance.

6. REFERENCES

[1] Andrew Whitmore, Anurag Agarwal, and Li Da Xu. The internet of things a survey of topics and trends. *Information Systems Frontiers*, 17(2):261–274, 2015.

[2] Roy Want, Bill N Schilit, and Scott Jenson. Enabling the internet of things. *Computer*, (1):28–35, 2015.

[3] J.A. Stankovic. Research directions for the internet of things. *Internet of Things Journal, IEEE*, 1(1):3–9, Feb 2014.

[4] Steve Mann, Jason Nolan, and Barry Wellman. Sousveillance: Inventing and using wearable computing devices for data collection in surveillance environments. *Surveillance & Society*, 1(3):331–355, 2002.

[5] David Zakim and Matthias Schwab. Data collection as a barrier to personalized medicine. *Trends in pharmacological sciences*, 36(2):68–71, 2015.

[6] Parastoo Alinia, Ramyar Saeedi, Ali Rokni, Bobak Mortazavi, and Hassan Ghasemzadeh. Impact of on-body sensor localization on met calculation using wearables. In *Body Sensor Networks (BSN), 2015 IEEE International Conference on*, June 2015.

[7] Kai Kunze and Paul Lukowicz. Dealing with sensor displacement in motion-based onbody activity recognition systems. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 20–29. ACM, 2008.

[8] Ramyar Saeedi, Janet Purath, Krishna Venkatasubramanian, and Hassan Ghasemzadeh. Toward seamless wearable sensing: Automatic on-body sensor localization for physical activity monitoring. In *The 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2014.

[9] Christopher M Bishop et al. *Pattern recognition and machine learning*, volume 4. springer New York, 2006.

[10] Alberto Calatroni, Daniel Roggen, and Gerhard Tröster. Automatic transfer of activity recognition capabilities between body-worn motion sensors: Training newcomers to recognize locomotion. *Eighth international conference on networked sensing systems (INSS'11), Penghu, Taiwan*, 6, 2011.

[11] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, Oct 2010.

[12] Daniel Roggen, Kilian FÃűrster, Alberto Calatroni, and Gerhard TrÃűster. The adarc pattern analysis architecture for adaptive human activity recognition systems. *Journal of Ambient Intelligence and Humanized Computing*, 4(2):169–186, 2013.

[13] Marc Kurz, Gerold Holzl, Alois Ferscha, Alberto Calatroni, Daniel Roggen, Gerhard Troster, Sagha, et al. The opportunity framework and data processing ecosystem for opportunistic activity and context recognition. *International Journal of Sensors Wireless Communications and Control*, 1(2):102–125, 2011.