Share-n-Learn: A Framework for Sharing Activity Recognition Models in Wearable Systems with Context-Varying Sensors

SEYED ALI ROKNI, Washington State University
HASSAN GHASEMZADEH, Washington State University

Wearable sensors utilize machine learning algorithms to infer important events such as behavioral routine and health status of their end-users from time-series sensor data. A major obstacle in large-scale utilization of these systems is that the machine learning algorithms cannot be shared among users or reused in contexts different than the setting in which the training data are collected. As a result, the algorithms need to be retrained from scratch in new sensor-contexts such as when the on-body location of the wearable sensor changes or when the system is utilized by a new user. The retraining process places significant burden on end-users and system designers to collect and label large amounts of training sensor data. In this article, we challenge the current algorithm training paradigm and introduce *Share-n-Learn* to automatically detect and learn physical sensor-contexts from a repository of shared expert models without collecting any new labeled training data. Share-n-Learn enables system designers and end-users to seamlessly share and reuse machine learning algorithms that are trained under different contexts and data collection settings. We develop algorithms to autonomously identify sensor-contexts and propose a gating function to automatically activate the most accurate machine learning model among the set of shared expert models. We assess the performance of Share-n-Learn for activity recognition when a dynamic sensor constantly migrates from one body-location to another. Our analysis based on real data collected with human subjects on three datasets demonstrate that Share-n-Learn achieves, on average, 68.4% accuracy in detecting physical activities with context-varying wearables. This accuracy performance is about 19% more than 'majority voting', 10% more than the state-of-the-art transfer learning, and only 8% less than the experimental upper bound.

CCS Concepts: • **Human-centered computing** → *Ubiquitous and mobile computing*; • **Computing methodologies** → *Machine learning*; • **Computer systems organization** → *Sensor networks*; • **Hardware** → *Emerging technologies*;

ACM Reference format:
Seyed Ali Rokni and Hassan Ghasemzadeh. 2018. Share-n-Learn: A Framework for Sharing Activity Recognition Models in Wearable Systems with Context-Varying Sensors. *ACM Trans. Des. Autom. Electron. Syst.* 1, 1, Article 1 (May 2018), 27 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

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 to meet objectives of the application [1–4]. At the heart of these human-centered IoT systems is *human monitoring* where physiological and behavioral state of the user are assessed using wearable sensors or those deployed in the environment. Smartphones [5, 6], wrist-band sensors [7–9], smart-home sensors [10], smart-watches [11], necklaces [12, 13],

DOI: 10.1145/nnnnnnn.nnnnnnn

smart-shoes [14, 15], smart-insoles [16], and sensors embedded in clothing are only few examples of the sensors utilized for human monitoring [13, 17]. These systems have proved effective in applications such as activity recognition [9, 18], elderly care [19–21], gait analysis[11, 15, 16], and anomaly detection [22]. Typically, the sensors acquire physical measurements, use 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.

The machine learning algorithms allow for continuous and real-time extraction of important physiological and behavioral information from sensor data. The generalizability of these algorithms, however, has remained a challenge to date, mainly due to the dynamically changing configuration of the system. In particular, the algorithms need to be reconfigured (i.e., retrained) upon any changes in configuration of the system, such as displacement/ misplacement/ mis-orientation of the sensors. For example, an activity recognition algorithm trained with a wrist-worn sensor achieves poor accuracy performance when used with a smartphone worn on the waist. Practically, the development of supervised machine learning algorithms requires model training using sufficiently large amounts of labeled training data, a process that is deemed to be time consuming, labor-intensive, and expensive [23]. Therefore, it is important that we develop new methodologies for sharing already trained machine learning models in order to prevent the tedious process of collecting labeled training data for every sensor-context.

Our pilot application in this article is *activity recognition* where a new wearable sensor is utilized by the end-user and an accurate activity recognition model needs to be trained without the requirement for collecting labeled training instances from the user or human expert. The main challenge arises from the fact that the new sensor exhibits various contextual dynamics such as constant displacement, misplacement, and mis-orientation in the body coordinate. Prior research [24–26] suggests that one can develop activity recognition algorithms that compensate for such sensor-context dynamics at the cost of collecting and labeling sensor data in new sensor-context settings. However, such algorithms are only reliable if we collect sufficient labeled training data for all possible sensor-contexts and with all users who adopt the system. Failure to do so results in directly reusing algorithms that were trained in a different setting, which leads to significant accuracy decline in new settings. For instance, it is shown that the accuracy of activity recognition algorithms drops more than 60% due to sensor displacements [27] or upon adoption of the system by new users [28]. The reason for such a sudden performance degradation is that the machine learning algorithms learn a model based on a set of training data that is collected in a particular sensor-context such as with a fixed body location. However, because the training and future unseen data are not in the same feature space and not have the same distribution [29], the outcome of the algorithms changes significantly when the system is utilized in a situation different than that of the training, such as when the on-body location of the sensor changes.

In this article, we present development and validation of *Share-n-Learn*, a novel framework that partners a set of shared activity recognition models with a dynamically context-varying wearable sensor to recognize human activities autonomously and in real-time with no need to collect ground truth labeled training data from the human expert. We focus on cases where multiple context-specific algorithms (i.e., expert models) are shared for use by the dynamic sensor where the dynamic sensor is worn/used on various body-locations during its operation. We propose an approach for learning a gating function to choose the most accurate expert model based on the observed sensor data. Share-n-Learn generates and autonomously labels a training dataset by examining observations of the dynamic sensor and associating those observations with synchronously sampled observations of a static sensor affixed to a particular body-location. After this initial setting, Share-n-Learn uses the automatically collected sensor data to learn a gating function for expert model activation/selection.

We compare the performance of Share-n-Learn with that of several mixture-of-experts models and a transfer learning technique and demonstrate that Share-n-Learn outperforms all the algorithms under comparison.

Our multi-view learning approach presented in this paper is a novel method of sharing activity recognition capabilities of several sensors, with already trained classifiers, for use by a dynamic sensor, which does not have any previously activity recognition models. 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 use in the dynamic view. The development of multi-view learning solutions that enable transfer of machine learning knowledge from previously trained models to new physical contexts in human-centered IoT applications is a new research direction. Our sensor-context learning approach contributes to the 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 research may open a new avenues in designing wearable and IoT systems of the future that are not only accurate but also autonomous in learning their underlying computational models without constant interaction with end-users to collect and label sensor data.

2 RELATED WORK

Our work in this article is related to three main research areas including sensor localization, transfer learning, and mixture-of-expert models. Share-n-Learn not only eliminates the tedious process of training supervised node localization algorithms but also combines transfer learning [30] and mixture-of-experts (MoE) concepts [29, 30] in a unified framework. In this section, we briefly discuss related studies in each of the three categories.

2.1 Sensor Localization

Several recent studies proposed techniques to detect on-body location of wearable sensors using machine learning algorithms. Authors in [24] proposed an approach to combine sensor localization within activity classification. The general idea is to train a machine learning algorithm capable of detecting on-body location of the sensor by examining sensor readings acquired from inertial sensors such as accelerometer and gyroscope sensors. As soon as the location of the sensor is identified, one can use an activity recognition model specifically trained for the detected wearing site in order to infer human activities. Authors in [31] proposed a probabilistic context modeling that uses ambient sensors to detect sensor context. Then, they used the context model to rank possible activity labels. However, the main limitation of such approaches is that they require a set of labeled training data to train the sensor localization or context recognition model. Furthermore, the set of possible sensor locations (on-body wearing sites) is limited by the training data. As discussed previously, collecting ground truth labeled data to train a supervised classifier for location/context detection is an expensive and time-consuming process. Share-in-Learn addresses this limitation by autonomously and implicitly learning sensor-context and activating a shared expert activity recognition model.

2.2 Transfer Learning

Transfer learning approaches are classified into instance transfer, feature representation transfer, parameter transfer, and relational knowledge transfer [30]. In order to transfer instances, TrAdaBoost [32], an extension of AdaBoost[33], was proposed to enable knowledge transfer from one domain to another by utilizing the training data collected in a source domain to incrementally construct a training dataset in a target domain. This approach, however, assumes that the data in both source and target domains have the same feature space, an assumption that is not applicable to wearable sensors with dynamically changing configurations and different modality.

In the area of transfer learning, multi-view learning approaches [34] are more desirable for knowledge transfer when source and target domains deal with similar data points (i.e., instances) but have different feature spaces. Several algorithms such as Manifold Alignment [35] attempt to align feature spaces of the two domains using un-informed methods assuming that labeled data is available in the source view/domain but not in the target view. The un-informed learning approach works only under certain assumptions on the underlying distributions of the data in the source and target views. Furthermore, this approach achieves only suboptimal accuracy results because its accuracy is upper bounded by the accuracy of the source view [36]. On the other hand, informed learning approaches such as Co-Training [37] and Co-Expectation-Maximization (Co-EM) [38] assume that labeled training data are available in the target domain and intend to use multi-view learning techniques to use a small amounts of the labeled data in each view to train two separate classifiers each for one view/domain [39].

In smart-home applications, [40] showed that it is possible to avoid data collection phase by transferring classifier models of activity recognition in one home to another home with similar activity recognition systems. To resolve the need for a common feature space, they proposed to compute “meta-features” as the common ground on which the knowledge transfer can occur. However, defining meta-features was an off-line process, which makes such an approach infeasible for real-time applications such as wearable-based health monitoring. Moreover, the proposed method was utilized only to binary sensors embedded in one’s home.

In contrast to informed and uninformed methods, Teacher / Learner (TL) transfer learning has being used when there is no direct access to training data [39]. Instead, to gather enough training data, source trained classifier operates simultaneously with the target learner and provides the labels of newly observed data points. Although noticeably less studies have been done using teacher/learner approaches, studies of using these approaches could increase the performance of transfer learning. Studies in [36, 41, 42] apply the teacher/learner model to develop opportunistic systems capable of performing reliable activity recognition in a dynamic sensor environment. The study in [42] showed that by synchronizing current sensor and new sensor, the existing sensor can provide the labels of future activities. This approach requires two sensors be worn for a long time until a sufficient amount of different activities be performed by the user. Furthermore, it needs constant rate of data transmission between existing sensor and new sensor. One of the main problems of this approach and many other teacher/learner approaches is that the accuracy of the learner’s training data is bounded by the accuracy of the teacher. Moreover, they rely on a reliable source classifier because the only source of a ground truth is the source sensor and thus the learner is completely reliant upon labels provided by teacher. Authors in [43, 44] suggested a calibration method for transferred labels by clustering observations in a compound feature space of source and target and incorporating observation of target sensor in knowledge of source sensor. However, they assumed that the location of the target sensor is fixed. We argue that such an assumption is unrealistic because in real-world applications the physical context of the sensor changes consistently. For example, the on-body location of a smartphone can change not only over time but also from one end-user to another. With the increasingly growing smartphone ownership [45], over 90% of the U.S. adults under 65 are anticipated to own a smartphone in 2018. Therefore, we need to develop new mechanisms for autonomous sharing, reuse, and adoption of computational models.

2.3 Mixture-of-Experts (MoE) Models

Our work in this article is also related to the mixture-of-experts models. Historically, the MoE methods exploit a divide-and-conquer strategy to build an ensemble classifier [46–48]. These methods partition the problem space into subspaces each dedicated to one expert model. Particularly, they train a gating function to coordinate the process of assigning input instances to different partitions. These methods could be classified into two categories [49] including implicit MoE and explicit MoE. In the implicit MoE methods [50], the gating function is trained simultaneously with the expert models. In other words, during the training phase of the experts, the gating function examines the efficiency and weakness of each expert in different sub-spaces and update weights of the gating function accordingly. On the other hand, in the explicit MoE methods [51, 52], the problem space is explicitly partitioned using techniques such as clustering and each expert is then assigned to a cluster.

Our approach in developing a MoE-based model in this paper has differences and similarities with both of the aforementioned MoE categories. Similar to explicit MoE, Share-n-Learn assumes that there exists an explicit unknown partitioning in problem spaces and there is an expert for each sub-domain. However, contrary to explicit MoE, the clustering algorithm based on similarity or density of data points cannot well partition the feature space in our problem under study. On the other hand, consistent with implicit MoE, we partition the data points based on performance of the experts. Yet, our expert models remain unchanged during gating function training phase. In other words, our approach combines the benefits of a Teacher/ Learner transfer learning method with those of a MoE paradigm to incrementally learn not only actual partitioning of the data but also the expert assignments.

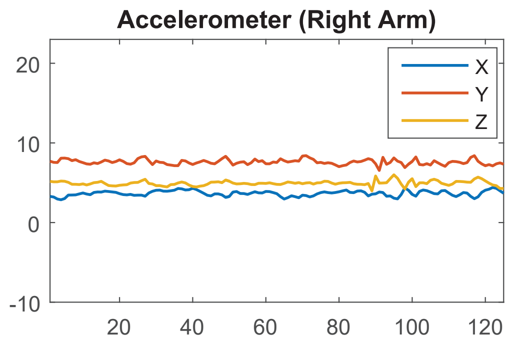
3 PRELIMINARIES

In this section, we briefly describe the process of activity recognition in wearable sensor systems and discuss challenges associated utilization of context-varying wearable sensors for activity recognition.

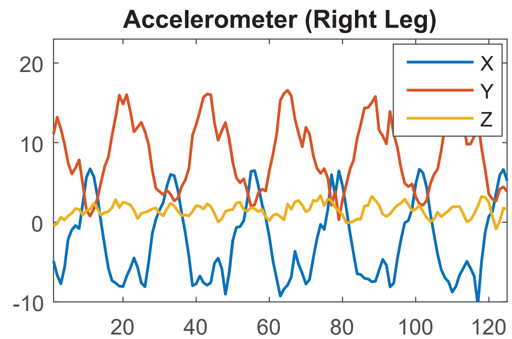
3.1 Activity Recognition

A sensing device typically has several sensors for capturing different states of the user (e.g., body motion), an embedded software module to perform signal processing, machine learning and information extraction, and a radio for data transmission. When a decision is made on the collected sensor data, the results can be used locally or forwarded to a back-end storage on the cloud for further processing, and to provide decision support. In this article, however, we limit our focus to the network around the user.

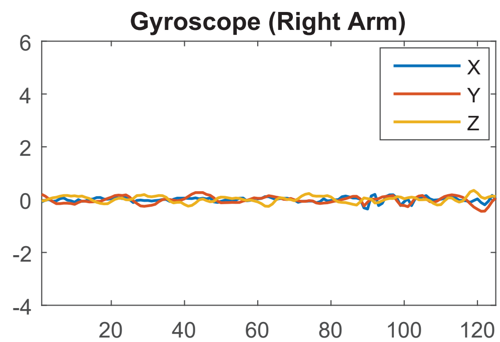
In activity recognition, readings from inertial sensors such as accelerometers, magnetometers, and gyroscopes undergo signal processing and machine learning to detect human movements such as ‘walking’, ‘running’, or ‘sitting’. Each sensing node processes sensor readings through a chain of embedded software for signal processing and machine learning. The signals that are sampled by each sensor node are first passed through a filter to reduce high frequency noise. The next phase is segmentation which is intended to identify ‘start’ and ‘end’ points of the movements being classified. Conventionally, segmentation is performed using a sliding time window with some overlap between consecutive windows. The next module is ‘feature extraction’ which is responsible for computing statistical characteristics of the signal segment. Features represent different attributes of the signal such as ‘peak-to-peak amplitude’, ‘standard deviation’, and ‘mean value’ [53]. Finally, a trained classification model uses the extracted features to determine the current activity of the user.



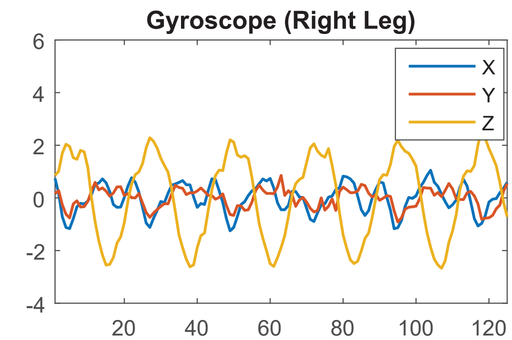
(a) Accelerometer readings of a sensor worn on Right-Arm



(b) Accelerometer readings of a sensor placed on Right-Leg



(c) Gyroscope readings of a sensor worn on Right-Arm



(d) Gyroscope readings of a sensor placed on Right-Leg

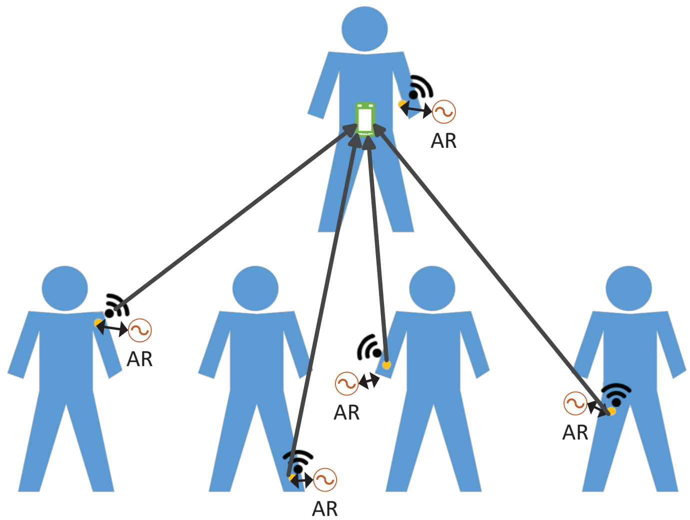
Fig. 1 A 5-second signal reading during 'cycling' activity as captured by accelerometer and gyroscope sensors worn on two different body locations, right-arm and right-leg. As shown in (a) and (c), 'cycling' is well pronounced by a sensor placed on right-leg, while the same activity appears stationary using a sensor node placed on right-arm as shown in (b) and (d).

In this article, we focus on the classification module when a new sensor without a trained classifier utilized and such sensor transitions from one body-location to another at the user's comfort.

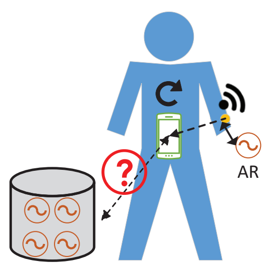
3.2 Context-Varying Sensors

Conventionally, activity recognition models are trained assuming that the sensors are worn on fixed (and known *a priori*) body locations. However, when the on-body location of the sensors changes, the activity recognition model fails to accurately classify human movements. This potentially limits scalability of human-centered IoT systems because the users are constrained to wear the sensors only on predefined locations on the body or use them according to the context or experimental protocol with which the data collection and activity recognition training has taken place. To extend wearability of the system and enhance the ability to detect activities of different body segments, it is reasonable to have a dynamic (i.e., context-varying) sensor that the user can wear around their body as desired.

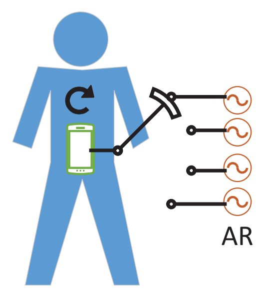
Advances in embedded sensor design and wearable electronics allow end-users to utilize new devices such as smartphones whose on-body location changes consistently. In such a realistic scenario, the dynamic sensor, however, is able to gather and store a repository of models of different



(a) Model Sharing Phase



(b) Gating Function



(c) Activity Recognition

Fig. 2 *Share-n-Learn* framework for autonomously learning to activate appropriate activity recognition model among a set of shared expert models each of which is expert in one particular sensor-context such as fixed body-location; (a) Model sharing: several networks with context-invariant machine learning models share their activity recognition model for use by a context-varying sensor; (b) Gating function training: the context-varying 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); and (c) Activity recognition: the trained gating function is used to switch between the shared models based on the detected sensor-context.

contexts/locations that it has not been trained to detect. The task of autonomously distinguishing among various sensor-contexts by examining sensor data is quite challenging because different body locations exhibit different signal patterns during the same activity. Figure 1 shows an example of sensor readings captured at 25 Hz during ‘cycling’ activity from two different body locations. Figure 1a and Figure 1c show 3D acceleration and 3D angular velocity as captured by accelerometer and gyroscope sensors placed on Right-Arm (RA), respectively. Figure 1b and Figure 1d illustrate the acceleration and angular velocity expressed by a Right-Leg (RL) sensor node. From these graphs, it is clear that ‘cycling’ is pronounced significantly differently by the two body segments (i.e., right-leg and right-arm). Share-n-Learn intends to automatically detect sensor-context and activate a machine learning model appropriate for the current context of the dynamic sensor.

4 SHARE-N-LEARN FRAMEWORK

Figure 2 shows an overview of the Share-n-Learn framework that develops an integrated model of sensor-context detection and activity recognition for a context-varying (i.e., dynamic) sensor without collecting any labeled training data. Initially, as shown in Figure 2a, several context-specific activity recognition (AR) models each trained in a particular sensor-context are shared for use with a context-varying sensor for which there exist no trained activity recognition model. For the purpose of this paper, we limit sensor-context to on-body location of the sensor. That is, each shared model is associated with an activity recognition model that is trained with a sensor affixed to a specific body-location. The dynamic sensor may migrate from one body-location to another (e.g., a smartphone used on various locations on the body). In the example shown in Figure 2, 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 sensor has obtained a repository of different expert models. In the next phase, called *gating function training*, the dynamic sensor uses its local sensor observations and the knowledge of an assistive static sensor (e.g., 'left wrist') to learn an algorithm for sensor-context detection and expert model selection. Throughout this paper, we refer to the observations made by the dynamic sensor as 'dynamic view' and those observed 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 is utilized only during gating function training to facilitate model sharing. This sensor can be eliminated from the network once the gating function in trained. Alternatively, the static sensor may remain as part of the wearable network.

The dynamic sensor does not posses an inference model (e.g., sensor localization algorithm) to detect its current context (e.g., on-body sensor location). 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. The goal of our proposed framework is to detect dynamic context of the sensor without collecting labeled training data. As soon as the dynamic context is detected autonomously, the sensor will choose the corresponding shared model for activity recognition. During the gaiting function training, Share-n-Learn 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 as shown in Figure 2c.

4.1 Problem Statement

An observation X_i made by a wearable sensor at time 'i' can be represented as a 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.

Considering k different locations in the 'dynamic view', the probability distribution of an observation X_i is given by

$$P(X_i) = \sum_{l=1}^{k} P(X_i|l)P(l) \tag{1}$$

Assuming that observations of each location follow a specific distribution, we can consider the set of observations X as a mixture distribution where each X_i is generated in k different ways [54].

In particular, if all sensor observations associated with a body-location were generated from an unknown Gaussian distribution, we can assume that the sensor data follow a *Gaussian Mixture* where the parameter l in Equation 1 is a hidden variable. Therefore, Equation 1 can be rewritten as follows.

$$P(X_i) = \sum_{l=1}^{k} P(X_i|\mu_l;\sigma_l)P(l) \tag{2}$$

Unsupervised mixture models such as Gaussian Mixture Model (GMM) attempt to estimate a good probabilistic representation of the data. In general, if a good clustering of the sensor data exists, then one can use techniques such as Expectation Maximization (EM) to cluster the sensor observations. In presence of a good clustering of the data, one can attempt to develop an assignment problem where each cluster is assigned a shared expert model according to some assignment error. In the past, we have studied such an optimization problem for addition a static sensor to wearable network [43]. We used Hungarian algorithm [55] for minimum cost assignment of activity labels to clusters in context-invariant sensor views. Unfortunately, the problem under study in this article involves a dynamic sensor that migrates across on-body locations. Therefore, the assumption that the number of on-body locations is known in advance is unrealistic. As a result, hard assignment methods such as k-means cannot be used as they require that the number of clusters/contexts be known *a priori*. In case of probabilistic algorithms such as EM algorithm, the number of conditional probabilities for each data point is equal to the number of components in the mixture model (e.g., the number of clusters). Although when the number of clusters in unknown, these clustering approaches are usually paired with an optimization framework such as BIC (Bayesian Information Criterion) to find the optimal number of clusters [56]. On the other hand, even density-based clustering algorithms such as DBSCAN [57] or OPTICS [58] are not effective because from one on-body location, instances of different activities are not necessary close to each other in the feature space. For example, two instances of one high intensity activity such as "running" from two different locations could look more similar than two instances of different intensities such as "running" and "sitting" which are observed from the same location. In other word, we cannot use a similarity-based clustering algorithm to put all observations from one location to the same cluster. Therefore, mixture model paradigm with pre-clustering phase is not feasible to model sensor data in context-varying views.

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 resulting a set of available experts $AR = \{AR_1, \ldots, AR_K\}$ for context-specific activity recognition. 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 $\mathcal{X}=\{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\}$ 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 detect 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). We define the expertise domain of an expert model as follows.

Definition 4.1 (Expertise Domain). Let $\mathcal{X}$ be the set of all observations made by the dynamic sensor. For expert model $AR_i \in AR$, an *Expertise Domain* is defined as a subset $S \subset X$ where E_i, the mis-classification error associated with model AR_i on S, is significantly less than other expert models. In other words $E_i \ll E_j$, $j \neq i$.

When a sensor migrates from one location to another, its expertise domain transitions from one domain to another. That is, the domain of observations made by the dynamic sensor can be divided into regions based on expertise of the expert models in detecting activities. To decide among expertise domains, we propose to train a gating function to accurately select corresponding expert model.

Definition 4.2 (Gating Function). A *Gating Function* is a soft decision function, g, which gets current observation X_t as input and assigns a probability p_i to expert model AR_i based on its expertise on X_t. We note that $\sum_i p_i = 1$.

4.2 Mixture-of-Experts Modeling

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. As discussed in Section 4.1, modeling the problem with mixture models such as GMM which only rely on the nature of data and do not consider performance of the available experts is ineffective. In particular, ignoring the information of class label from experts may lead to an unbalanced partitioning [49]. On the other hand, the problem of partitioning based on observation-label relationship can be modeled using mixture-of-experts which encourages specialization of expert models [59]. Figure 3 shows a simple example of mixture-of-experts with two different experts and two class labels. In this example, *Expert 1* is responsible for detecting instances that reside above the gating line while *Expert 2* is in charge of predicting instances that fall below the gating line. This example shows why clustering algorithms fail to address the problem of model sharing. In fact, a clustering algorithm may assign those blue circle instances close to the gating boundary to the Expert 2. In addition, 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.

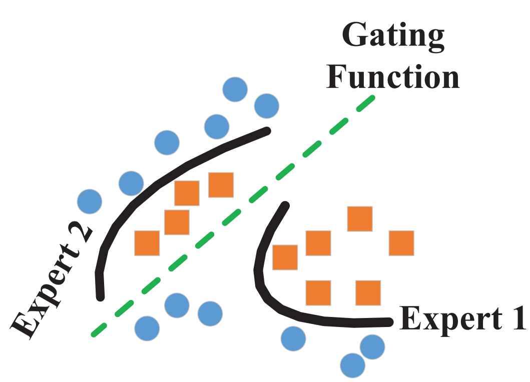


Fig. 3 An example of Mixture-of-Expert with two class labels (blue circles and orange rectangles) and two different detection models (Expert 1 and Expert 2). Each expert is responsible for accurate classification of the instances residing in one side of the gating function.

Let g be a gating function that assigns a probability p_i to expert AR_i based on the observation X_t made in the dynamic view at time t. The decision expressed by AR_i on X_t is a probability vector P^t over all possible activities such that $P^t_j = P(Y_t = j)$. Therefore, the probability of each activity label j can be computed by

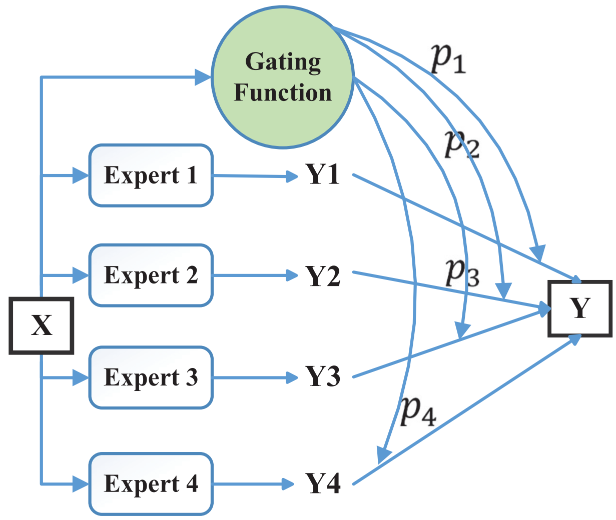


Fig. 4 An example of Mixture-of-Expert with four Experts. Each expert receives X, an observation made in dynamic view, and decides on the target activity as perceived by that expert. Simultaneously, the gating function assigns a probability p_i to the predicted outcome of each expert. The final outcome is computed using a combination of all experts' decisions.

$$
\begin{aligned}
P(Y_t = j|X_t, g) &= \sum_{i=1}^{K} P(Y_t, AR_i|X_t, g) \\
&= \sum_{i=1}^{K} g(AR_i|X_t)P(Y_t = j|X_t, AR_i) \\
&= \sum_{i=1}^{K} p_i \; P(Y_t = j|X_t, AR_i)
\end{aligned}
\tag{3}
$$

As the aforementioned formulation suggests, our proposed framework first chooses the context expert model AR_i with probability p_i based on the observation X_t (i.e $g(AR_i|X_t)$). It then computes the probability of label j with respect to AR_i and current observation: $P(Y_t = j|X_t, AR_i)$. Therefore, the Mixture-of-Expert training algorithm aims to maximize the likelihood probability in Equation 3 on the training data to learn the parameters of the gating function g. Figure 4 illustrates an example of the gating function with four experts. A new observation X is fed to all four experts as well as the gating function. Then, the gating function aggregates the predictions of all experts based on assigned probabilities. To find maximum likelihood of a probability model, Bayes classifier or Bayes Gaussian classifier are commonly used in the literature as an iterative approach where the problem consists of several observed random variables (e.g. activity label) as well as hidden random variables [29, 60, 61]. The challenging task in our framework, however, is that there is no labeled training data available for the dynamic sensor to learn a model. The only source of knowledge in dynamic view is noisy predictions made by the static sensor. To overcome this challenge, we introduce a novel approach using Teacher/Learner transfer learning to train the gating function .

4.3 Gating Function Training

Figure 5 shows how sensors and the user interact to realize SSCL. As the user wears both static and dynamic sensors while performing daily activities, the static sensor assists the dynamic sensor to train its 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 the dynamic sensor. The dynamic sensor then compares the prediction vector of 'Teacher' with the predictions made by all expert models in its repository. The dynamic sensor learns from the information provided by 'Teacher' and selects the closest decision as the candidate of sensor-context for that particular observation. The data gathered in this fashion over time are used as training data for learning the gating function.

ALGORITHM 1: Synchronous Sensor-Context Learning (Static View)

repeat
  '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;
until *all activity have been observed*;

ALGORITHM 2: Synchronous Sensor-Context Learning (Dynamic View)

Import expert models;
sensor-context training data $\leftarrow \{\}$;
repeat
  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;
until *static sensor sends prediction*;
construct a sensor-context classifier based on training data;

Algorithm 1 and Algorithm 2 show our multi-view sensor-context learning[1] approach from the static view and dynamic view perspectives, respectively, to train the gating function. Figure 6 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 of the user 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 the 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 wirelessly. At the same time, the dynamic sensor queries its expert models to make a prediction on the current observation made in the dynamic view. The dynamic sensor selects the expert with the closest

[1]The implementation for the sensor-context learning algorithm is available at https://github.com/ali-rokni/SSCL

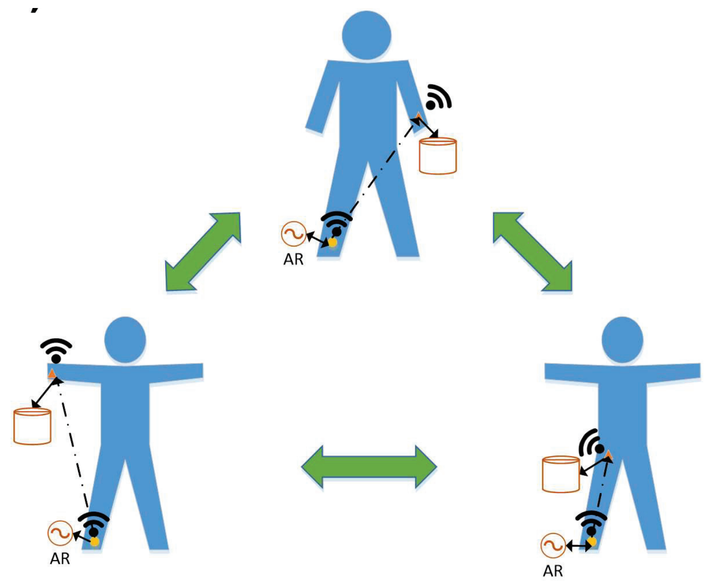


Fig. 5 The SSCL framework in action: while static and dynamic sensors are synchronized, the user performs physical activities and the static and dynamic sensors collaborate. Meanwhile, the location of the dynamic sensor might change and the process continues.

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 article are independent of the type of 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 physical activities independently.

4.4 Measuring Decision Disagreement

To compute disagreement between predictions made by the static sensor and that of each expert model associated with the dynamic view, we need a comparison metric among the experts. In our experiments, we use *Euclidean* distance function between the decision vector in the static sensor and that of each expert model in the 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 in the static sensor and P' is the decision vector for one expert in the dynamic view, we can compute their distance using Minkowsly distance function give by

$$\Delta(P, P') = \sqrt[r]{\sum_{j=1}^{M} (P_j - P'_j)^r} \tag{4}$$

Using Euclidean ($r = 2$) we place a higher weight than Manhattan distance ($r = 1$) on larger differences in any dimensions.

4.5 Handling Uncertainty

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

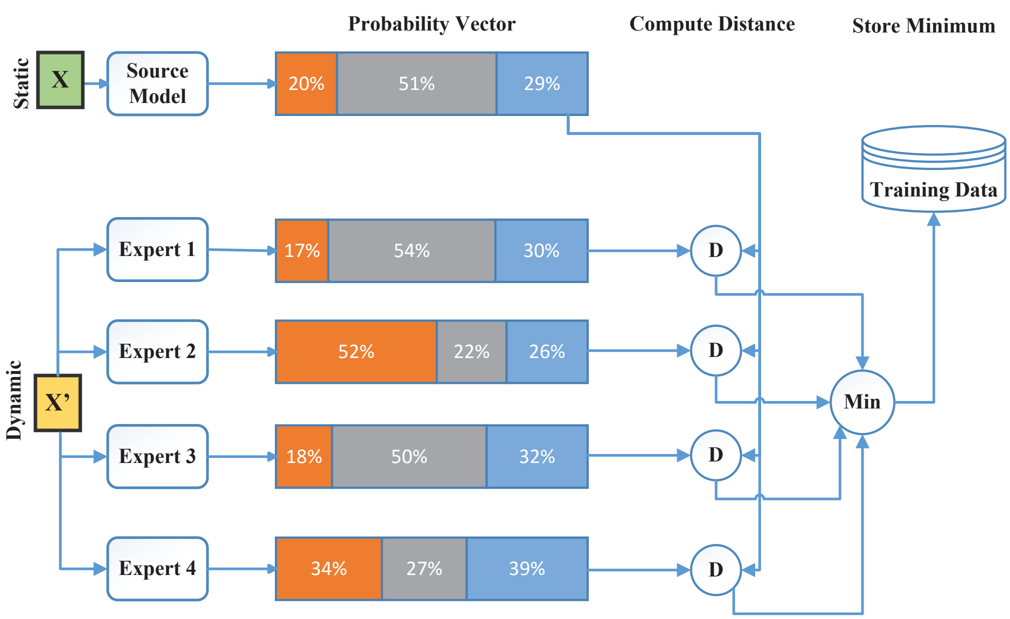


Fig. 6 The process of training: when an activity occurs, the static model and all experts of the dynamic model make their predictions. Next, distances of experts decisions and static model prediction are compared and the index of the closest expert is selected as sensor-context label for the current dynamic observation.

5 EXPERIMENTAL RESULT

In this section, we demonstrate the effectiveness of our multi-view learning algorithms using real data collected with human subjects. In particular, we assess the preformance of our algorithm on three datasets.

5.1 Datasets

In the first dataset, called Sport and Daily Activity (SDA), 8 human subjects including 4 female and 4 male subjects between the ages 20 and 30 performed 19 different physical activities for 5 minutes each [62–64]. The subjects performed the activities at Bilkent University Sports Hall while wearing 5 motion sensor nodes on five different body locations. Each sensor node is an Xsens MTx

[65] inertial sensor unit with a 3-axis accelerometer, a 3-axis gyroscope and a 3-axis magnetometer. The sampling frequency is set to 25Hz and the 5-min signals are divided into 5-sec segments so that $480 (= 60 \times 8)$ signal segments are obtained for each activity. The collected dataset contains over $9,120,000$ samples of acceleration, angular velocity and magnetic field. According to [62], subjects performed the activities on their own style which led to an obvious inter-subject variation in the speed and amplitude of the same activity across subjects and trials. This dataset is publicly available through the UCI Repository[2].

The second dataset, which is also publicly available[3], is referred to as OPPORTOUNITY dataset (OPP) [66]. In the OPP dataset, four subjects operated in a room simulating daily life activities by interacting with real objects while 5 inertial measurement units (IMUs) were placed on their back, and (Right/Left) (Upper/Lower) Arms. Each subject performed 5 runs of each activity of daily living (ADL) following a given scenario as *Grooming, Relaxing, Preparing Coffee, Drinking Coffee, Preparing Sandwich, Eating Sandwich, Cleaning up, wrapped in a Starting and Breaking*. The activities are annotated in five different levels and in this paper we used locomotion (i.e., sit, stand, lie and walk) annotations.

The third dataset, referred to as as TRA (Transitional Activities), includes data collected with 3 human subjects who performed 12 different transitional activities while wearing 5 wireless motion sensor nodes on their 'Left Wrist' (LW), 'Right Arm' (RA), 'Left Thigh' (LT), 'Right Ankle' (RK) and 'Waist' (WA) [67]. Each sensor node has a 3-axis accelerometer and a 2-axis gyroscope. The physical activities include *Stand to Sit*, *Sit to Stand* , *Sit to Lie*, *Lie to Sit*, *Jump*, *Turn Clockwise*, *Bend to Grasp*, *Step Backward*, *Look Back*, *Kneeling Right*, *Rise from Kneeling* and *Return from Looking back*. 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.

Table 1 and Table 2 briefly summarize statistics and on-body sensor locations for the three dataset used in this article.

Table 1 Datasets overview

Dataset	# subjects	# sensor	# activities	Type of Activities
OPP	8	5	19	Sport and Daily Activities
SDA	4	5	4	Locomotion Activities
TRA	3	5	12	Transitional Activities

Table 2 Sensor Locations

	SDA		OPP		TRA	
No	On Body Location	Abbr.	On Body Location	Abbr.	On Body Location	Abbr.
1	Torso	TO	Back	BAK	Waist	WA
2	Right Arm	RA	Right Upper Arm	RUA	Left Wrist	LW
3	Left Arm	LA	Right Lower Arm	RLA	Right Arm	RA
4	Right Leg	RL	Left Upper Arm	LUA	Left Thigh	LT
5	Left Leg	LL	Left Lower Arm	LLA	Right Ankle	RK

[2]https://archive.ics.uci.edu/ml/datasets/daily+and+sports+activities
[3]http://archive.ics.uci.edu/ml/datasets/OPPORTUNITY+Activity+Recognition

5.2 Data Analysis

We first performed a Gaussian smoothing [68] as a preprocessing step to filter out instrumental noises and to partially deal with inter-subject variations. Then, from each segment of the individual sensor streams, we extracted 9 statistical features. Potentially, there are many different features that can be extracted from human activity signals [69]. However, as shows in Table 3, these features aim to capture both shape and amplitude of the signals. For example, features such as AMP and MNVALUE are useful to capture intensity of the signal while STD or NMI intend to capture morphology of the signal. For general activity recognition purpose, these features should be able to recognize different activities without distinguishing among styles that different subject may perform [44].

Table 3 Feature list

Label	Description
AMP	Amplitude of the signal
MNV	Mean of the signal
P2P	Peak to peak amplitude
STD	Standard deviation of the signal
RMS	Root mean square power
NMI	Number of local minima
NMX	Number of local maxima
MMI	Mean local minima
MMX	Mean local maxima

To design a comprehensive experiment, we choose one of sensor locations in Table 2 as the location of ‘static’ node (local expert) and assume that the dynamic sensor could freely transition from one location to another among the 4 remaining wearing sites or on-body locations. We continue this procedure by alternating the ‘static’ location with another body location and consider all remaining locations for the dynamic view. This strategy allows us to construct a static view with one affixed sensor node as well as a dynamic view with a sensor node consistently relocating among four wearing sites.

To ensure arbitrary relocation of the dynamic sensor, we randomly shuffle observations associated with both sensors. In order to maintain the sensors synchronized, we use the same random sequence for both static and dynamic views. We then divide the obtained sequence of the observations into three chunks including ’experiment preparation’, ’train the gating function’, and ’test’. Since the Share-n-Learn framework assumes that there is an available repository of expert models, we use the first chunk of the data for preparing the experiment to apply our SSCL algorithms. In particular, we develop 5 expert models each of which corresponding to a particular location listed in Table 2. Note SSCL starts after this phase and treats each expert model as a black-box. Having a repository of expert models for each location, we train the gating function on the second chunk of the data. We choose one of the sensors as static and assume that the target sensor is added to the system and relocates arbitrarily among the remaining locations. We follow the SSCL algorithm to train the the gating function using similarity between likelihood probabilities provided by the static sensor and the decision of each expert model. After training the gating function , we evaluate the performance of SSCL on the third chunk of the data.

Table 4 Sensor Locations

SDA		OPP		TRA	
Source/Static Node	Accuracy	Source/Static Node	Accuracy	Source/Static Node	Accuracy
TO	61.05	BAK	76.25	WA	66.44
RA	57.89	RUA	72.74	LW	45.68
LA	70.0	RLA	65.72	RA	52.59
RL	66.32	LUA	74.85	LT	59.48
LL	74.74	LLA	74.42	RK	44.45
Average	66.0	Average	72.79	Average	53.74

5.3 Performance of Static Sensor

Before analyzing the performance of our Share-n-Learn framework, we first evaluate the performance of each expert model. The goal of this analysis is to assess the robustness of a single static sensor node in recognizing all experimental activities. We note that the static sensor is used as a local expert and the source of knowledge for training the gating function. Therefore, it is important to gauge the level of robustness of the predictions made by a static sensor. As explained in Section 5.2, all context-specific models are generated in the experiment preparation phase. For this purpose, a random forest classifier is used to develop context-specific models for each one of the sensor nodes. The evaluation study of 179 classifiers from 17 families on the entire UCI repository has shown that "the classifiers most likely to be the bests are Random Forests" [70].

As shown in Table 4, the accuracy of the activity recognition model ranged from 72.8% for the OPP dataset to 53.4% for the TRA dataset on average. The highest accuracy (i.e., 76.25%) belonged to the 'BAK' sensor in the OPP dataset. In contrast the 'RK' sensor in the TRA dataset achieved the lowest accuracy among all others sensors.

These results suggest that not all activity instances are reliably distinguishable using a single view by the static sensor. For instance, more than 23% of the instances are mis-classified by the 'BAK' sensor in the OPP dataset. Therefore, the assumption that the source of knowledge (i.e., static sensor in this case) is perfect is not realistic. This observation suggests that solely relying on predictions of a static sensor or each of the experts while learning the gating function can result in obtaining a poor recognition model by over-fitting on the noise of static sensor (local expert). 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.

5.4 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. Although prior studies combine sensor localization or context detection within activity classification [24, 31], they require a set of labeled data to train the localization/context-learning model. To the best of our knowledge, there is no prior research that addresses the problem of mixture-of-experts model considering no available ground truth and only relying on the performance of available experts for wearable sensors. Thus, we developed several baseline and intuitive MoE-based 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. The other natural method of dealing with the problem of mixture-of-experts is voting. In the majority voting approach, we query all experts and perform activity recognition based on majority votes of the experts. In addition to these two mixture-of-experts-based methods, we compare our approach with the experimental upper bound, which is obtained from decisions of the correct classifier corresponding to the current sensor location (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 several transfer learning algorithms proposed for wearable computing. In these approaches, however, the only source of knowledge will be the local expert. 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 [42], which are applicable to the synchronous teacher/ learner approach studied in this article. We note, however, that such algorithms do not incorporate the knowledge provided by external/shared classifiers. Additionally, these approaches are originally designed and analyzed when the location of the sensor remains unchanged. Calatroni et. al proposed the Naive approach as reusing the 'source' (i.e., local expert) classifier in 'target' (i.e., dynamic sensor in this case). They emphasized that this method only works when 'source' and 'target' are highly similar (e.g., sensors are co-located on the body and are homogeneous). The *System-Supervised* method refers to the case where 'target' assigns labels to its observations based on labels predicted by 'source'.

5.5 Performance Metrics

We evaluate the efficiency of our gating function by comparing the performance of the dynamic sensor in activity recognition using the context-detection gating function with all the competing algorithms using the following performance metrics.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{6}$$

where

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

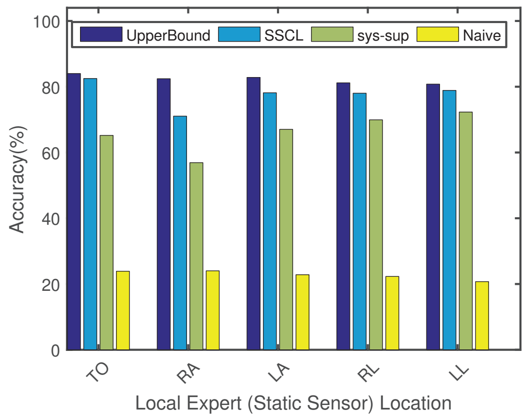
$$\tag{9}$$

and TP, TN, FP and FN denote true-positive, true-negative, false-positive, and false-negative, respectively. Each metric captures different aspect of the classifier performance. In other words, only the accuracy of the classifier is not enough for assessing the performance of a classifier and we should make sure that while we increase the recall, we do not loose much of precision.

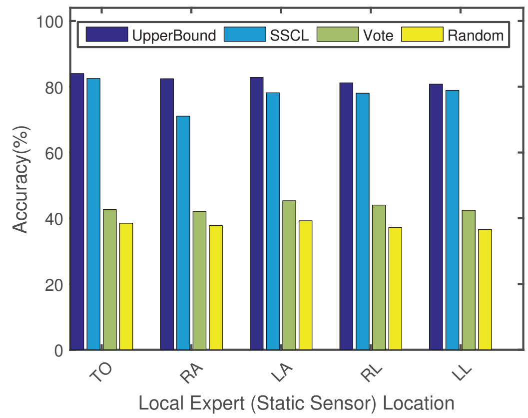
Our analysis compares the activity recognition performance using all the discussed algorithms (i.e., random, majority voting, naive, system-supervised, and upper-bound) in the next section.

5.6 Comparative Analysis

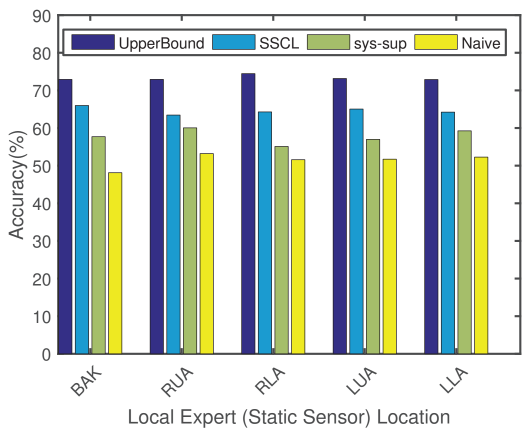
For each dataset, we study different scenarios where the local expert could be any of the five sensors mentioned in Table 2. For each scenario, we consider each one of the other four locations as possible locations of the dynamic sensor. In other word, while the static sensor is fixed in one of the five locations, the dynamic sensor is continuously being relocated among all other four locations. For example in SDA dataset, when 'TO' is considered as the location of 'local expert', we use 'RA', 'LA', 'RL', and 'LL' as on-body locations of the dynamic sensor.



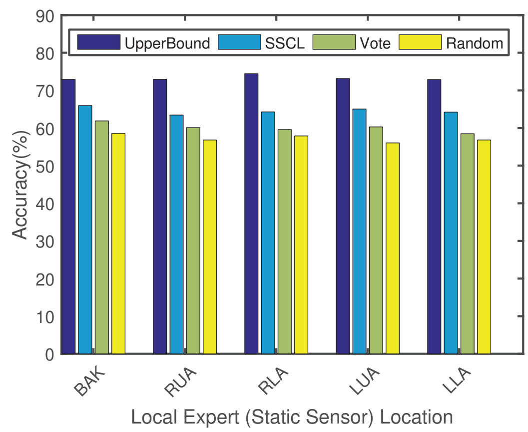
(a) SDA Local Expert Accuracy



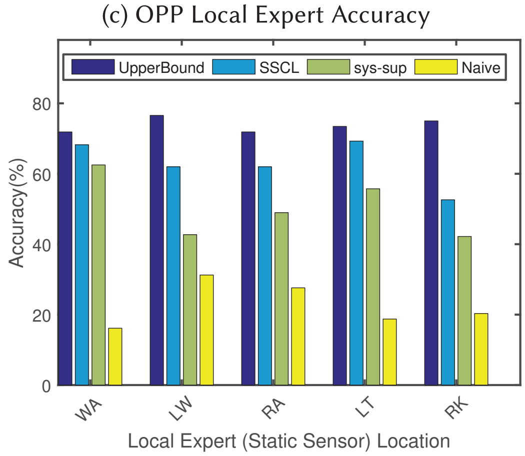
(b) SDA Mixture of Expert Accuracy



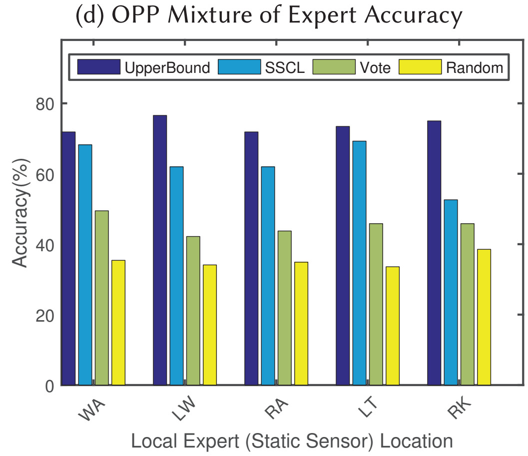
(c) OPP Local Expert Accuracy



(d) OPP Mixture of Expert Accuracy

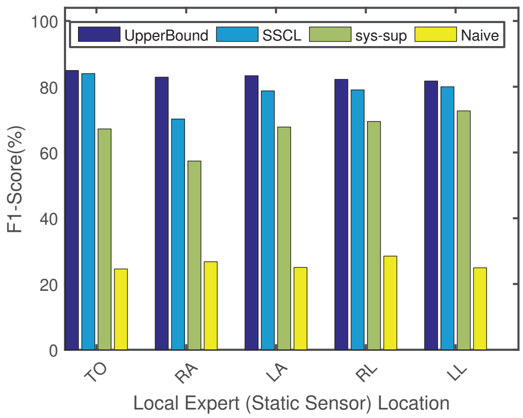


(e) TRA Local Expert Accuracy

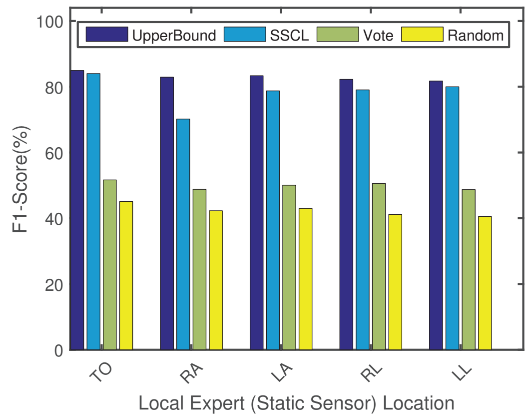


(f) TRA Mixture of Expert Accuracy

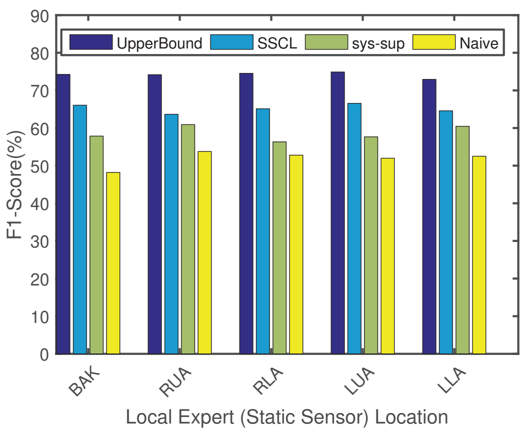
Fig. 7 Accuracy of activity recognition using different mixture-of-experts and local expert methods. (a), (c), (e) shows the accuracy of local expert methods while (b), (d), (f) of mixture-of-experts approaches under comparison including randomly selected expert (Random), majority voiding (Vote), our approach (SSCL), and experimental upper bound (UpprBnd)



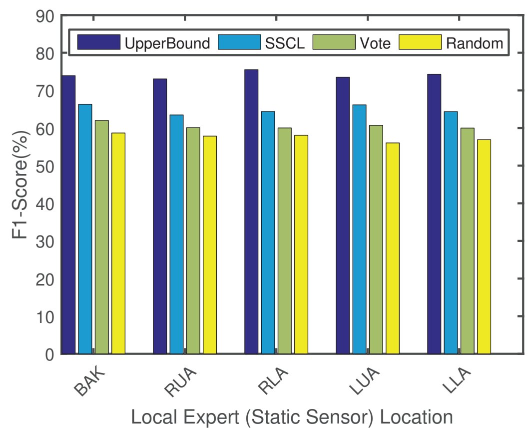
(a) SDA Local Expert F1



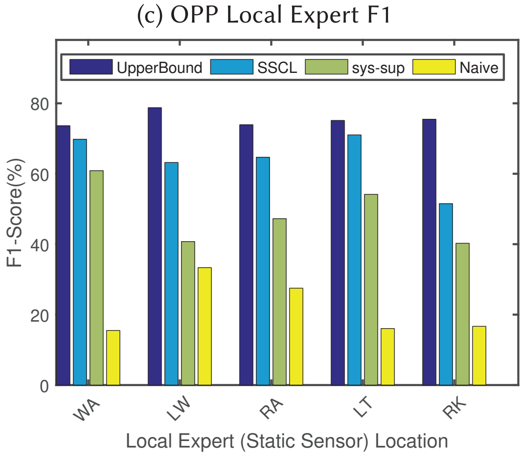
(b) SDA Remote Expert F1



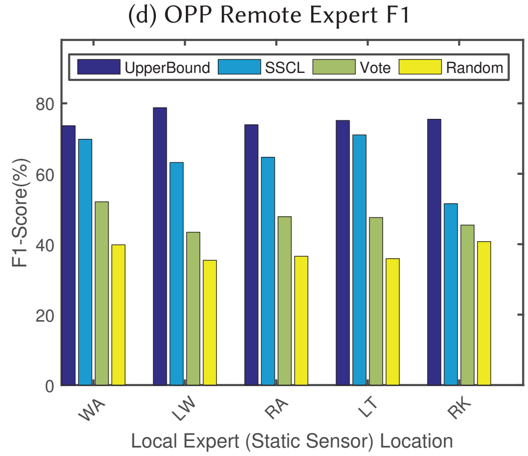
(c) OPP Local Expert F1



(d) OPP Remote Expert F1



(e) TRA Local Expert F1



(f) TRA Remote Expert F1

Fig. 8 F1 score of activity recognition using different mixture-of-experts and local expert methods. (a), (c), (e) shows the F1 of local expert methods while (b), (d), (f) of mixture-of-experts approaches under comparison including randomly selected expert (Random), majority voiding (Vote), our approach (SSCL), and experimental upper bound (UpprBnd)

5.6.1 Accuracy of Dynamic Sensor. According to the Section 5.4, we compare our approach with methods in two paradigms: mixture of experts and local experts. Figure 7 shows that accuracy of Share-n-Learn in comparison with mixture of expert methods. The result of experiment on SDA dataset shows that the accuracy of the activity recognition algorithm based on our automatic sensor-context detection approach (SSCL) ranges from 82.5% for 'Torso' to 71.1% for 'Right Arm'. On average, SSCL-based activity recognition achieves 77.7% accuracy. This accuracy is 34.4% and 39.8% higher than the accuracy of 'Majority Voting' and 'Random' respectively.

Comparing to local expert approaches based on Teacher/ Learner transfer learning in SDA dataset, SSCL-based activity recognition achieves 11.4% higher accuracy than 'system-supervised' method in detecting the 19 experimental activities. Furthermore, our approach outperforms the 'naive' approach showing more than 54% higher accuracy. Comparing to experimental upper bound, SSCL-based activity recognition works very well and is only 4.5% less accurate than activity recognition model built using ground truth labels (i.e., activity recognition model trained with known sensor-context labels).

For OPP dataset, our experiment shows that the average accuracy of the activity recognition algorithm using SSCL is 64.6% which is 4.5% and 7.4% higher than the accuracy of 'Majority Voting' and 'Random' respectively. An interesting observation here is the close accuracy of the 'Majority Voting' and 'Random' methods. It shows that for detecting the locomotion activities, classifiers of different locations could use similar discriminating features such that each of them could be able to distinguish observation of other locations. Using local expert approaches on OPP dataset, SSCL-based activity recognition achieves 6.8% higher accuracy than 'system-supervised' Furthermore, our approach outperforms the 'naive' approach showing more than 13.2% higher accuracy.

The accuracy of using SSCL for activity recognition in TRA dataset ranges from 69.2% for 'LT' to 52.6% for 'RK'. On average, SSCL-based activity recognition achieves 62.8% accuracy. This accuracy is 17.4% and 27.5% higher than the accuracy of 'Majority Voting' and 'Random' respectively. Similar to two other datasets, the SSCL shows higher activity recognition performance comparing to the local expert teacher/learner approaches. Particularly, the accuracy of activity recognition using SSCL is 12.4% and 40% higher than 'system-supervised' and 'Naive'. Note that the accuracy of the local expert and the overall performance of shared experts are two important factors which contribute to the final accuracy of SSCL method.

An interesting observation is very low accuracy of 'naive' method from local expert paradigm even comparing to 'Random' method in MoE approaches. As in Section 5.4 introduced, the 'Naive' method does not leverage shared models and just reuses the model of local expert (i.e. static sensor). However, according to the experiment setup, this model is not trained for any of the dynamic sensor contexts. While the 'Random' method randomly chooses one of shared models which one of them are correct expert model and provides highly accurate labels. Another observation in Figure 7 is the higher accuracy of the 'system-supervised' though lower than SSCL, among other rival methods. As discussed in Section 5.4, in the 'system-supervised' method, labels are assigned to the observations of the dynamic sensor based on the prediction of local expert. The higher accuracy of 'system-supervised' compared to 'Majority Voting' shows that relying on the prediction local expert leads to more accurate labels compared to the majority voting among non-expert models from different locations.

5.6.2 F1 Score of Dynamic Sensor. While, precision and recall try to assess false positive rate and false negative rate of the classifier, F1-score combines these two measures and provides a unique metric. Figure 8 compares F1-score of mixture-of-expert methods. The F1-score of SSCL in SDA dataset ranges from 84% to 70.2% with average of 78.4%. This shows an improvement of 28.4% and 36% comparing to 'Majority Voting' and 'Random' methods respectively. In comparison to experimental upper bound, SSCL achieves only 4.6% less F1-score. In addition to the MoE methods, the evaluation of local expert methods illustrated in Figure 8a. It shows that in comparison to 'system-supervised' method, SSCL achieves 11.5% higher F1-score and the 'Naive' method has the minimum F1-score with average of 52.4%.

For OPP dataset, the F1-score of SSCL-based classifier ranges from 67.1% to 63.9% with average of 65.5%. This shows an improvement of 3.8% and 7.2% comparing to 'Majority Voting' and 'Random' methods, respectively. However, this F1 is 8.6% less than F1-score of experimental upper bound. Furthermore, it shows 15.4% higher F1-score in comparison to 'system-supervised' method and 21.6% comparing to the 'Naive' method. Similarly, the SSCL activity recognition classifiers has an average 64% F1-score on TRA dataset. This score is 16.8% and 26.4% higher than 'Majority Voting' and 'Random' methods, respectively. Additionally, comparing to the local expert methods illustrated in Figure 8, it shows 15.4% and 42.2% higher performance than 'system-supervised' and 'Naive' method, respectively.

6 DISCUSSIONS

The proposed work in this article is related to two interconnected challenges, reliability and scalability, in sensor-based computer systems. The vision is that by seamless integration of sensor systems, we can create more reliable systems whose interoperability and knowledge exchange will improve reliability of the obtained measurements and will therefore result in large-scale adoption of these technologies. We define reliability as the ability of the system to maintain its performance (measured by established metrics such as accuracy, precision, recall, and f-measure) regardless of the environment in which the system is deployed. The current utility of wearable systems is limited to controlled environments such as laboratory settings and clinics. When used in uncontrolled environments, their performance drops dramatically due to various forms of uncertainty such as on-body displacement, misplacement, and mis-orientation of the sensor, adoption of the system by a new user, addition of a new sensor to the network, removal of a sensor from the network, and sensor platform dynamics such as changes in sensor modality and sampling rate. We define scalability as the ability of the system to maintain its usability regardless of the environment in which the system is deployed. It is possible to collect and label sensor data for a particular type of uncertainty. For example, when the system is adopted by a new user, one can collect sensor data with the new user, label the data, and train a new activity recognition classifier in which case the classifier will achieve a high accuracy as long as distribution of the sensor data in source and target is the same. However, this approach is unrealistic and un-scalable because we will need to collect and label sensor data for every user, setting, sensor, and platform. Furthermore, the user's environment, setting, context, and behavior changes consistently. Therefore, if we are to develop scalable computer systems deployable in large scales, we will need to develop novel approaches for the computational models to reconfigure automatically and autonomously without human supervision.

In this paper, we introduced Share-n-Learn for reusing already trained machine learning models with an autonomous sensor-context detection algorithm. The algorithm identifies the best expert model and is trained with no ground truth label 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. In particular, Teacher/ Learner (TL) transfer learning [30] has been used when there is no direct access to training data. When the location of the target sensor is fixed, several studies [36, 41, 42] apply the teacher/ learner model to develop an opportunistic system capable of performing reliable activity recognition. Authors in [43] showed that by combining the observations of source and target sensors, the system can achieve a performance higher than that of the source itself assuming the location of the target sensor remains unchanged. However, the assumption of a 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 without access to ground truth labeled data in dynamically changing wearable sensor environments.

In reality, models learned for different body location might come from different feature spaces. Although in our experiments we considered all of the experts work on the same feature space, the same approach can be utilized for the cases when feature spaces of the two body locations are different. This way, when an observation is passed through an expert, a prior step of feature extraction projects the raw sensor observation to the corresponding feature space. Even in case of known number of clusters and the same feature space, [71] showed that clustering algorithms are not efficient when observations are mixed from different on-body locations.

Share-n-Learn requires involvement of a static sensor during training of the gaiting function. We, however, note that the labels provided by the static sensor are generated automatically and according to a previously trained machine learning algorithm for the static sensor. Therefore, inclusion of a static sensor and generation of automatically generated labels by such a sensor is still consistent with our definition of computational autonomy, which refers to the science of developing and reconfiguring computational models, without human supervision, in dynamically changing environments. Note that the labels transmitted by the static sensor do not necessarily represent ground truth data. As shown through our results, predictions of the static sensor are not perfect and consequently, labels provided by the static sensor are not always accurate. In contrast, ground truth data often requires human supervision to ensure collection of highly accurate labels.

7 FUTURE WORK

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 by the accuracy of the shared models. Dynamic attributes of sensor-context are not limited to real-time changes in on-body 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 addition, leveraging the active learning [72] or the optimal experimental design [73], we can efficiently reduce the number of required label transfers and optimize the power consumption of learning phase. In particular, [74] studies active learning methods for activity recognition and claims one can achieve up to 80% reduction in required data points. According to active learning hypothesis, the learning algorithm could choose the data point from which it wants to learn. However, the source/ static sensor is not perfect and sometimes provides other sensors with the noisy information. Having more than one static sensor probably in different locations, we can leverage 'crowd sourcing' methods to 'average out' noisy information base on expertise of each source sensor [75–77]. However, still the question remains how to use active learning querying methods when the source of knowledge is noisy. Authors in [78] suggest that model could be improved by selective repeated labeling rather than a new instance. Particularly, in our problem due to the temporal aspect of observations, those data points do not repeat exactly. Therefore, the learner should decide instances similar to previous ones or synthesis some close observations. Additionally, recent studies on representation learning using deep neural networks [79] show that using learned features could lead to more efficient knowledge transfer.

In this study, we only focused on activity recognition applications using homogeneous sensors. 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. More specifically, we plan to research how to take advantage of ambient sensors to increase the robustness of source labels in an Internet-of-Things

In this project, we performed all our data analysis and algorithm develop off-line using data collected. Because all sensor measures are time-stamped, we had access to sensor readings from all sensor nodes within the wearable network. Our future work also involves implementation of Share-n-Learn algorithm on the sensor node for real-time training of a dynamic algorithm. In such scenario, the static and dynamic sensors need to be synchronized for joint monitoring of human activities. We expect that many existing synchronization algorithms can be adopted for use in our final real-time training framework.

8 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 Share-n-Learn that uses the knowledge of existing sensors and a repository of context-specific models to adapt with on-body sensor relocation. We used activity recognition task as our pilot application and develop an framework that enables to share the machine learning algorithms across different sensor contexts. We introduced a multi-view learning 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. The result of our experiments on 3 datasets demonstrates that Share-n-Learn on average achieves an activity recognition accuracy 68.4% for context-varying sensors which is only 8% less than the experimental upper bound performance.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation, under grants CNS-1566359 and CNS-1750679. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations.

REFERENCES

[1] A. Whitmore, A. Agarwal, and L. Da Xu, "The internet of things a survey of topics and trends," *Information Systems Frontiers*, vol. 17, no. 2, pp. 261–274, 2015.

[2] R. Want, B. N. Schilit, and S. Jenson, "Enabling the internet of things," *Computer*, no. 1, pp. 28–35, 2015.

[3] J. A. Stankovic, "Research directions for the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, 2014.

[4] S. Mann, J. Nolan, and B. Wellman, "Sousveillance: Inventing and using wearable computing devices for data collection in surveillance environments." *Surveillance & Society*, vol. 1, no. 3, pp. 331–355, 2002.

[5] A. Mehrotra, R. Hendley, and M. Musolesi, "Towards multi-modal anticipatory monitoring of depressive states through the analysis of human-smartphone interaction," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. ACM, 2016, pp. 1132–1138.

[6] R. Xiao, G. Laput, Y. Zhang, and C. Harrison, "Deus em machina: On-touch contextual functionality for smart iot appliances," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 2017, pp. 4000–4008.

[7] W. Qi and Y. Zhai, "The study on the life signs of clinical patients monitored by electronic wrist band," in *Computational Science and Engineering (CSE) and Embedded and Ubiquitous Computing (EUC), 2017 IEEE International Conference on*, vol. 2. IEEE, 2017, pp. 356–359.

[8] B. Fang, N. D. Lane, M. Zhang, A. Boran, and F. Kawsar, “Bodyscan: Enabling radio-based sensing on wearable devices for contactless activity and vital sign monitoring,” in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services.* ACM, 2016, pp. 97–110.
[9] M. Gjoreski, H. Gjoreski, M. Luštrek, and M. Gams, “Continuous stress detection using a wrist device: in laboratory and real life,” in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct.* ACM, 2016, pp. 1185–1193.
[10] A. Benmansour, A. Bouchachia, and M. Feham, “Multioccupant activity recognition in pervasive smart home environments,” *ACM Computing Surveys (CSUR)*, vol. 48, no. 3, p. 34, 2016.
[11] W. Xu, Y. Shen, Y. Zhang, N. Bergmann, and W. Hu, “Gait-watch: A context-aware authentication system for smart watch based on gait recognition,” in *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation.* ACM, 2017, pp. 59–70.
[12] M. A. U. Alam, N. Roy, A. Gangopadhyay, and E. Galik, “A smart segmentation technique towards improved infrequent non-speech gestural activity recognition model,” *Pervasive and Mobile Computing*, vol. 34, pp. 25–45, 2017.
[13] K. Isbister, K. Abe, and M. Karlesky, “Interdependent wearables (for play): A strong concept for design,” in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems.* ACM, 2017, pp. 465–471.
[14] H. Oh and M. D. Gross, “Awareable steps: functional and fashionable shoes for patients with dementia,” in *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers.* ACM, 2015, pp. 579–583.
[15] K. Baousi, N. Fear, C. Mourouzis, B. Stokes, H. Wood, P. Worgan, and A. Roudaut, “Inflashoe: A shape changing shoe to control underfoot pressure,” in *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems.* ACM, 2017, pp. 2381–2387.
[16] H. Ma and W.-H. Liao, “Human gait modeling and analysis using a semi-markov process with ground reaction forces,” *IEEE transactions on neural systems and rehabilitation engineering*, vol. 25, no. 6, pp. 597–607, 2017.
[17] M. Chen, Y. Ma, J. Song, C.-F. Lai, and B. Hu, “Smart clothing: Connecting human with clouds and big data for sustainable health monitoring,” *Mobile Networks and Applications*, vol. 21, no. 5, pp. 825–845, 2016.
[18] J. A. Ward, G. Pirkl, P. Hevesi, and P. Lukowicz, “Towards recognising collaborative activities using multiple on-body sensors,” in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct.* ACM, 2016, pp. 221–224.
[19] F. Nunes, N. Verdezoto, G. Fitzpatrick, M. Kyng, E. Grönvall, and C. Storni, “Self-care technologies in hci: Trends, tensions, and opportunities,” *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 22, no. 6, p. 33, 2015.
[20] S. Pinto, J. Cabral, and T. Gomes, “We-care: An iot-based health care system for elderly people,” in *Industrial Technology (ICIT), 2017 IEEE International Conference on.* IEEE, 2017, pp. 1378–1383.
[21] C. Ogonowski, K. Aal, D. Vaziri, T. V. Rekowski, D. Randall, D. Schreiber, R. Wieching, and V. Wulf, “Ict-based fall prevention system for older adults: Qualitative results from a long-term field study,” *ACM Trans. Comput.-Hum. Interact.*, vol. 23, no. 5, pp. 29:1–29:33, Oct. 2016. [Online]. Available: http://doi.acm.org/10.1145/2967102
[22] B. Arrington, L. Barnett, R. Rufus, and A. Esterline, “Behavioral modeling intrusion detection system (bmids) using internet of things (iot) behavior-based anomaly detection via immunity-inspired algorithms,” in *Computer Communication and Networks (ICCCN), 2016 25th International Conference on.* IEEE, 2016, pp. 1–6.
[23] D. Zakim and M. Schwab, “Data collection as a barrier to personalized medicine,” *Trends in pharmacological sciences*, vol. 36, no. 2, pp. 68–71, 2015.
[24] T. Sztyler and H. Stuckenschmidt, “On-body localization of wearable devices: An investigation of position-aware activity recognition,” in *Pervasive Computing and Communications (PerCom), 2016 IEEE International Conference on.* IEEE, 2016, pp. 1–9.
[25] P. Alinia, R. Saeedi, A. Rokni, B. Mortazavi, and H. Ghasemzadeh, “Impact of on-body sensor localization on met calculation using wearables,” in *Body Sensor Networks (BSN), 2015 IEEE International Conference on*, June 2015.
[26] K. Kunze and P. Lukowicz, “Dealing with sensor displacement in motion-based onbody activity recognition systems,” in *Proceedings of the 10th international conference on Ubiquitous computing.* ACM, 2008, pp. 20–29.
[27] R. Saeedi, J. Purath, K. Venkatasubramanian, and H. 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.
[28] R. Fallahzadeh and H. Ghasemzadeh, “Personalization without user interruption: Boosting activity recognition in new subjects using unlabeled data,” in *Proceedings of the 8th International Conference on Cyber-Physical Systems*, ser. ICCPS ’17. New York, NY, USA: ACM, 2017, pp. 293–302. [Online]. Available: http://doi.acm.org/10.1145/3055004.3055015
[29] C. M. Bishop *et al.*, *Pattern recognition and machine learning.* springer New York, 2006, vol. 4, no. 4.
[30] S. J. Pan and Q. Yang, “A survey on transfer learning,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 10, pp. 1345–1359, Oct 2010.

[31] J. Wen, J. Indulska, and M. Zhong, "Adaptive activity learning with dynamically available context," in *Pervasive Computing and Communications (PerCom), 2016 IEEE International Conference on.* IEEE, 2016, pp. 1–11.
[32] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th international conference on Machine learning.* ACM, 2007, pp. 193–200.
[33] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119 – 139, 1997.
[34] S. Sun, "A survey of multi-view machine learning," *Neural Computing and Applications*, vol. 23, no. 7-8, pp. 2031–2038, 2013. [Online]. Available: http://dx.doi.org/10.1007/s00521-013-1362-6
[35] C. Wang and S. Mahadevan, "Manifold alignment using procrustes analysis," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 1120–1127. [Online]. Available: http://doi.acm.org/10.1145/1390156.1390297
[36] M. Kurz, G. Holzl, A. Ferscha, A. Calatroni, D. Roggen, G. 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*, vol. 1, no. 2, pp. 102–125, 2011.
[37] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, ser. COLT' 98. New York, NY, USA: ACM, 1998, pp. 92–100. [Online]. Available: http://doi.acm.org/10.1145/279943.279962
[38] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training," in *Proceedings of the Ninth International Conference on Information and Knowledge Management*, ser. CIKM '00. New York, NY, USA: ACM, 2000, pp. 86–93. [Online]. Available: http://doi.acm.org/10.1145/354756.354805
[39] D. J. Cook, K. D. Feuz, and N. C. Krishnan, "Transfer learning for activity recognition: a survey," *Knowl. Inf. Syst.*, vol. 36, no. 3, pp. 537–556, 2013.
[40] T. van Kasteren, G. Englebienne, and B. J. Kröse, "Transferring knowledge of activity recognition across sensor networks," in *Pervasive computing.* Springer, 2010, pp. 283–300.
[41] D. Roggen, K. FÃűrster, A. Calatroni, and G. TrÃűster, "The adarc pattern analysis architecture for adaptive human activity recognition systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 4, no. 2, pp. 169–186, 2013.
[42] A. Calatroni, D. Roggen, and G. 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*, vol. 6, 2011.
[43] S. A. Rokni and H. Ghasemzadeh, "Plug-n-learn: automatic learning of computational algorithms in human-centered internet-of-things applications," in *Proceedings of the 53rd Annual Design Automation Conference.* ACM, 2016, p. 139.
[44] S. A. Rokni and H. Ghasemzadeh, "Autonomous training of activity recognition algorithms in mobile sensors: A transfer learning approach in context-invariant views," *IEEE Transactions on Mobile Computing*, 2018.
[45] A. Berenguer, J. Goncalves, S. Hosio, D. Ferreira, T. Anagnostopoulos, and V. Kostakos, "Are smartphones ubiquitous?: An in-depth survey of smartphone adoption by seniors," *IEEE Consumer Electronics Magazine*, vol. 6, no. 1, pp. 104–110, 2017.
[46] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.
[47] S.-P. Kim, J. C. Sanchez, D. Erdogmus, Y. N. Rao, J. Wessberg, J. C. Principe, and M. Nicolelis, "Divide-and-conquer approach for brain machine interfaces: nonlinear mixture of competitive linear models," *Neural Networks*, vol. 16, no. 5, pp. 865–871, 2003.
[48] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and systems magazine*, vol. 6, no. 3, pp. 21–45, 2006.
[49] S. Masoudnia and R. Ebrahimpour, "Mixture of experts: a literature survey," *Artificial Intelligence Review*, pp. 1–19, 2014.
[50] R. A. Jacobs, M. I. Jordan, and A. G. Barto, "Task decomposition through competition in a modular connectionist architecture: The what and where vision tasks," *Cognitive Science*, vol. 15, no. 2, pp. 219–250, 1991.
[51] M. H. Nguyen, H. A. Abbass, and R. I. Mckay, "A novel mixture of experts model based on cooperative coevolution," *Neurocomputing*, vol. 70, no. 1, pp. 155–163, 2006.
[52] R. Ebrahimpour, E. Kabir, and M. R. Yousefi, "Improving mixture of experts for view-independent face recognition using teacher-directed learning," *Machine Vision and Applications*, vol. 22, no. 2, pp. 421–432, 2011.
[53] H. Ghasemzadeh, N. Amini, and M. Sarrafzadeh, "Energy-efficient signal processing in wearable embedded systems: An optimal feature selection approach," in *Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design.* ACM, 2012, pp. 357–362.

[54] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.
[55] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
[56] C. Fraley and A. E. Raftery, "How many clusters? which clustering method? answers via model-based cluster analysis," *The computer journal*, vol. 41, no. 8, pp. 578–588, 1998.
[57] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.
[58] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: ordering points to identify the clustering structure," in *ACM Sigmod Record*, vol. 28, no. 2. ACM, 1999, pp. 49–60.
[59] S. A. Rokni and H. Ghasemzadeh, "Autonomous sensor-context learning in dynamic human-centered internet-of-things environments," in *Proceedings of the 35th International Conference on Computer-Aided Design*. ACM, 2016, p. 75.
[60] V. N. Vapnik and V. Vapnik, *Statistical learning theory*. Wiley New York, 1998, vol. 1.
[61] C. K. Williams and D. Barber, "Bayesian classification with gaussian processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1342–1351, 1998.
[62] K. Altun and B. Barshan, "Human activity recognition using inertial/magnetic sensor units," in *International Workshop on Human Behavior Understanding*. Springer, 2010, pp. 38–51.
[63] D. Barnan, N. C. KRISHNAN, and J. Diane, "Automated activity interventions to assist with activities of daily living," *Agents and Ambient Intelligence: Achievements and Challenges in the Intersection of Agent Technology and Ambient Intelligence*, vol. 12, p. 137, 2012.
[64] K. Altun, B. Barshan, and O. Tunçel, "Comparative study on classifying human activities with miniature inertial and magnetic sensors," *Pattern Recognition*, vol. 43, no. 10, pp. 3605–3620, 2010.
[65] X. M. S. U. Manual, "Xsens technologies bv," 2009.
[66] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. FÃűrster, G. TrÃűster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, H. Sagha, H. Bayati, M. Creatura, and J. d. R. MillÃąn, "Collecting complex activity data sets in highly rich networked sensor environments," in *Seventh International Conference on Networked Sensing Systems*, 2010. [Online]. Available: http://www.inss-conf.org/2010/
[67] H. Ghasemzadeh, N. Amini, R. Saeedi, and M. Sarrafzadeh, "Power-aware computing in wearable sensor networks: An optimal feature selection," *IEEE Transactions on Mobile Computing*, vol. 14, no. 4, pp. 800–812, 2015.
[68] A. Witkin, "Scale-space filtering: A new approach to multi-scale description," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'84.*, vol. 9. IEEE, 1984, pp. 150–153.
[69] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.
[70] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems," *J. Mach. Learn. Res*, vol. 15, no. 1, pp. 3133–3181, 2014.
[71] S. A. Rokni and H. Ghasemzadeh, "Synchronous dynamic view learning: a framework for autonomous training of activity recognition models using wearable sensors," in *Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks*. ACM, 2017, pp. 79–90.
[72] B. Settles, "Active learning literature survey," *University of Wisconsin, Madison*, vol. 52, no. 55-66, p. 11, 2010.
[73] V. Fedorov, "Optimal experimental design," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 5, pp. 581–589, 2010.
[74] H. Alemdar, T. L. van Kasteren, and C. Ersoy, "Using active learning to allow activity recognition on a large scale," in *International Joint Conference on Ambient Intelligence*. Springer, 2011, pp. 105–114.
[75] M. Sugiyama and N. Rubens, "Active learning with model selection in linear regression," in *Proceedings of the 2008 SIAM International Conference on Data Mining*. SIAM, 2008, pp. 518–529.
[76] D. C. Brabham, "Crowdsourcing as a model for problem solving: An introduction and cases," *Convergence*, vol. 14, no. 1, pp. 75–90, 2008.
[77] A. Carlson, J. Betteridge, R. C. Wang, E. R. Hruschka Jr, and T. M. Mitchell, "Coupled semi-supervised learning for information extraction," in *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 2010, pp. 101–110.
[78] V. S. Sheng, F. Provost, and P. G. Ipeirotis, "Get another label? improving data quality and data mining using multiple, noisy labelers," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 614–622.
[79] S. A. Rokni, M. Nourollahi, and H. Ghasemzadeh, "Personalized human activity recognition using convolutional neural networks," in *AAAI*, 2018.