2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation

UnTran: Recognizing Unseen Activities with Unlabeled data using Transfer Learning

Md Abdullah Al Hafiz Khan
Department of Information Systems
University of Maryland, Baltimore County
Email: mdkhan1@umbc.edu

Nirmalya Roy
Department of Information Systems
University of Maryland, Baltimore County
Email: nroy@umbc.edu

***Abstract*—The success and impact of activity recognition algorithms largely depends on the availability of the labeled training samples and adaptability of activity recognition models across various domains. In a new environment, the pre-trained activity recognition models face challenges in presence of sensing biasness, device heterogeneities, and inherent variabilities in human behaviors and activities. Activity Recognition (AR) system built in one environment does not scale well in another environment, if it has to learn new activities and the annotated activity samples are scarce. Indeed building a new activity recognition model and training the model with large annotated samples often help overcome this challenging problem. However, collecting annotated samples is cost-sensitive and learning activity model at wild is computationally expensive. In this work, we propose an activity recognition framework, *UnTran* that utilizes source domains' pre-trained autoencoder enabled activity model that transfers two layers of this network to generate a common feature space for both source and target domain activities. We postulate a hybrid AR framework that helps fuse the decisions from a trained model in source domain and two activity models (raw and deep-feature based activity model) in target domain reducing the demand of annotated activity samples to help recognize unseen activities. We evaluated our framework with three real-world data traces consisting of 41 users and 26 activities in total. Our proposed *UnTran* AR framework achieves $\approx$ 75% F1 score in recognizing unseen new activities using only 10% labeled activity data in the target domain. *UnTran* attains $\approx$ 98% F1 score while recognizing seen activities in presence of only 2-3% of labeled activity samples.**

I. Introduction

Activity recognition (AR) is a prolific research area in the era of Internet-of-Things (IoT), pervasive, wearable and smart computing [1][2][3]. With the proliferation of smart sensing devices, (i.e., smartphone, smartwatch etc.) various applications related to health care monitoring, obesity management, interactive gaming etc., have constantly been evolving to improve the human-centric services in the smart living environments. In contrast, AR models are typically built to recognize a predefined and limited set of activities, for example sitting, running, walking, exercising etc. In addition, the emerging diversity of wearable devices, their sensing capabilities and heterogeneities, and variations in human activities and their daily life-styles undermine the performance of known AR models. These traditional approaches solely rely on the specific environmental settings, heuristically selected handcrafted features and a predefined set of activities trained with a large set of annotated samples. Therefore, in general, obtaining reliable ground truth annotated activity samples is crucial to adapt AR systems in the target environment.

Scaling existing AR system is challenging due to the presence of handcrafted features that are dependent on domain knowledge and predefined environmental settings. Finding the optimal set of features across a predefined set of activities also requires domain expertise. To overcome these challenges, deep learning based unsupervised techniques have been proposed that help to choose an optimal set of domain dependent features [4]. However, training a deep network is computationally expensive, and requires a large set of activity samples tuning in the target domain. Moreover, limited amount of training samples in the target domain causes overfitting and network biasness that pose challenges for adapting AR models. Researcher proposed various domain adaptation techniques [5][6] that mostly involved co-training [7]. However, co-training a deep network requires both source and target domains' activity samples during the learning phase, which may not always be available. The most interesting insights of using multi-layered deep learning techniques is that it generates most generic features in the lower-layers and most specific features in the deeper layers [8]. Motivated by this, we bootstrap our AR framework by transferring the learned weight and bias parameters of a pre-trained deep network from the source to target domain. This partial layer knowledge transferring helps mitigate the need for domain dependent handcrafted features, reduces the computational cost, and minimizes the data distributions divergence between the source and target domains.

The presence of new activities in the target environment (domain) poses challenges when scaling an existing AR system. For example, AR model capable of recognizing exercising activities (such as push ups) can not correctly distinguish new activity like playing basket ball in the target domain. Evaluating the performance of existing AR models requires a large set of annotated activity samples in the target domain. However, it is not feasible to ask users to provide annotated samples for each of the activity instances and train a new activity model that can adapt the characteristics of a new environment. Our assumption is that the user can provide a small amount of annotated activity samples in the target domain. However, training new activity model with these small amount of labeled activity samples encounters two challenges: i) availability of

0-7695-6387-2/18/$31.00 ©2018 IEEE
DOI 10.1109/IoTDI.2018.00014

limited training data and ii) presence of imbalanced and unseen activities in the new domain. Therefore, it is difficult for the traditional AR model to cope with the new challenges while achieving the required activity recognition performance. In this work, we advocate to use the source domain activity recognition model in conjunction with two variants of target domain activity recognition models – statistical features based AR model and deep feature based AR model which help to mitigate the scarcity of label information in the target domain.

The key advantage of our activity recognition model is that it leverages the performance of existing AR model by learning the variability of the activity patterns using a small amount of labeled activity samples in the target domain. The key contributions of this work are summarized below.

- We exploit transfer learning enabled deep features representation techniques to mitigate the scarcity of activity samples in an unsupervised manner. We leverage our feature learning approach with a limited amount of training samples by transferring the first two layers of the source trained deep sparse autoencoder with deep learning classifier in the target domain. Our model helps percolate the existing weights and biases of the trained network in the target domain and constructs generalized feature space representation which help overcome the diversity across users' activities, environmental settings and sensing biasness.
- We transfer label information from source domain to target domain by utilizing source domain classifier and fuse decision with two target domain classifiers - handcrafted and deep networked generated feature based classifier. Fusing the knowledge of these three classifiers altogether helps solve the scarcity of label information and the imbalanced class problem in the target domain.
- We evaluate our AR framework, *UnTran* on three real-world activity data traces and demonstrate the effectiveness and efficacy of our proposed cross-domain activity recognition model.

II. Related Work

In this section, we review the existing work in three major areas: traditional machine learning approaches, deep learning and transfer learning in activity recognition.

A. Activity Recognition

In wearable pervasive computing, a plethora of research exists that recognizes human activities (i.e., playing basket ball, walking, standing etc.) [9][10][11][2]. Researchers employed various supervised machine learning algorithms (SVM, Decision Trees, Random Forest etc.) to classify human activity where these classifiers were trained with a large set of labeled activity samples. These traditional supervised machine learning algorithms are tuned to specific settings and do not perform well if deployed in a new environment where variations in users activity patterns, diversity of devices, and sensing biasness are omnipresent [12][2][3]. Traditional supervised AR models utilized handcrafted features. However, these features extraction process requires domain expert knowledge [13]. AR models trained with handcrafted features are not robust and scalable because of the existence of tightly bound feature space to a specific setting and the fixed number of activities in the source environment.

Learning features from unlabeled activity samples has also been explored recently [14][15]. These methods learn feature spaces from a large set of activity samples in the target domain. Therefore, reemploying the existing AR models requires a lot of annotated activity samples in the target domain. Researchers have investigated semi-supervised methods that help learn parameters using both labeled and unlabeled activity samples [16][17]. These techniques alleviates ground-truth annotation problem with a smaller pool of labeled samples from a large set of unlabeled activity samples. However, these methods are error prone and typically unable to replace the need for ground-truth annotated data from experts. In addition, it is always not feasible to collect a large number of labeled data traces or make requests to the human annotators. In an attempt to bootstrap an existing trained activity model, in this work, we advocate to use a small subset of unlabeled samples in addition with a small subset of labeled activity samples in the target domain.

In recent times, unseen activity recognition approaches have also been investigated. NuActive [18] proposed outliers-aware attribute based unseen activity recognition method using unlabeled activity data traces and showcased classification performance by training AR model with selected activity samples using active learning. However, the performance of the attribute based activity recognition model degrades in presence of existing and new activities. The attribute based activity recognition models assume that each activity has a unique set of attributes. [9] proposed attribute and feature based fusion method to improve the performance of AR model with the help of labeled activity samples. Although [18], [9] achieved better performance in inferring new activities, the authors failed to consider the sensing biasness, activity patterns, variations, and user diversity in the targeted domain. Most of the existing work discovered new activities within the same domain and also relied on the unique set of manually defined attributes for each of the activities. Moreover, defining attributes of each activity is a time-consuming task, and requires a lot of efforts and domain knowledge. In this work, we reduce this effort by transferring the knowledge from the source to target domain in an autonomous way by using deep transfer learning techniques. Our proposed AR framework helps mitigate the scarcity of labeled activity samples by utilizing labels information from source domain to target domain.

B. Deep Learning

Various research have focused on learning features from sensor data traces using deep learning techniques [19][20][21]. Deep learning based feature extraction method has been applied for the activity recognition research problem. The main objective of this approach is to learn hidden activity patterns from the sensor data traces and discover meaningful patterns

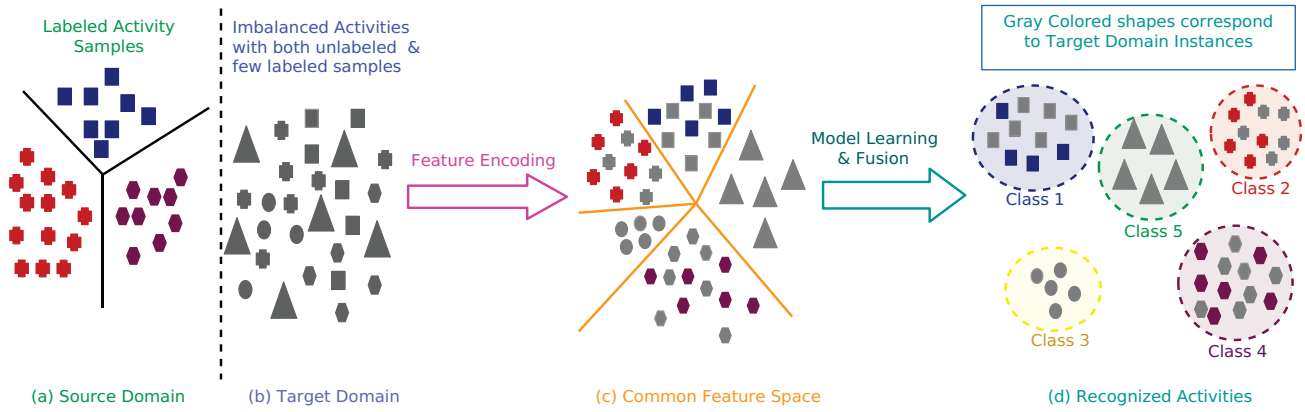


Fig. 1: Overview of our activity recognition approach. (a) Source domain labeled activity instances, (b) Target domain contains both unlabeled and few labeled activity instances, (c) Common feature space for classification, and (d) Resulting activities after classification. Note that different shapes correspond to different activities.

without the human intervention. These automatic hidden patterns can be discovered with two deep learning approaches – supervised and unsupervised [22][23][24]. Supervised deep feature learning approaches are computationally expensive and require a large set of annotated samples. On the other hand, unsupervised deep learning methods demand a large set of unlabeled training samples [21][23]. Both of these methods are computationally expensive and require a significant amount of training time to adjust the network parameters. None of these approaches work well in presence of scarce activity samples. In order to deploy these AR models, further tuning of parameters is necessary in the target environment. In this work, we exploit the benefits of existing pre-trained sparse deep autoencoder enabled activity recognition model in the source domain to reduce the required samples in the target domain.

C. Transfer Learning

Scalability and adaptability are the persistent research challenges in activity recognition application domain. Transfer learning based activity recognition techniques have been investigated recently [25][26][27]. Nonetheless, a limited number of aspects of transfer learning enabled activity recognition has been investigated. [26] proposed uninformed transfer learning algorithm that help minimize cross-subject variability to scale human activity recognition. The authors proposed to transfer label information from the source domain to recognize unlabeled activities in the target domain and assumed availability of a large set of unlabeled data samples with similar activities in the target domain. [28] addressed the versatility of sensor modality and sensor position independence by transferring a similar set activity labels from an existing trained sensor node to a new sensor node without any user intervention. In contrast, we propose a framework that is able to infer activities with a limited number of samples and in presence of new activities in the target domain.

III. Overview of UnTran Framework

We briefly outline the different algorithmic components of our proposed activity recognition framework, UnTran in this section.

A. Problem Settings

We design *UnTran* framework for recognizing unseen activities in presence of user activity patterns diversity, sensing biasness and limited activity samples in the target domain. We assume that the source domain has a significant amount of labeled activities and a pre-trained activity model. Our *UnTran* framework works with limited activity data, imbalanced and unseen activities. To tackle this, we propose to construct common feature space where similar activity samples help generate similar feature space. However, with a limited activity samples, AR model suffers from overfitting problem in the deployed target domain. To address this, we combine the inference decisions from multiple AR models (one source AR model and two target AR models) and deploy that to recognize activities in the target domain. Fig. 1 represents the overview of our activity recognition approach.

Mathematically we define our problem as follows. Let source domain training data $D_s = \{x_i^{(s)}, y_i^{(s)}\}_{i=1}^{N_s} = \{\mathbf{X}^s, \mathbf{y}^{(s)}\}$, where $x_i^{(s)} \in \mathbf{R}^d$ denotes d-dimensional source-domain instance and $y_i^{(s)}$ denotes the corresponding label of C_s categories. We assume that the target domain contains d-dimensional unlabeled data instances and target domain data are represented as $D_t = \{x_j^{(t)}, y_j^{(t)}\} = \{\mathbf{X}^{(t)}, \mathbf{y}^{(t)}\}$ where $\mathbf{y}^{(t)}$ is the class label to infer. We also assume that target domain constitutes both seen and unseen activities and contains activity categories, $C_t = \{C_{un} \cup C_{sn}\}$, where seen activities categories, C_{sn} and C_{un} represents unseen activity categories. Due to the heterogeneity in the target domain, marginal probability distributions of data between these two domains are different ($P(\mathbf{X}^s) \neq P(\mathbf{X}^t)$). It is worth to note here that transfer learning based approach works when both

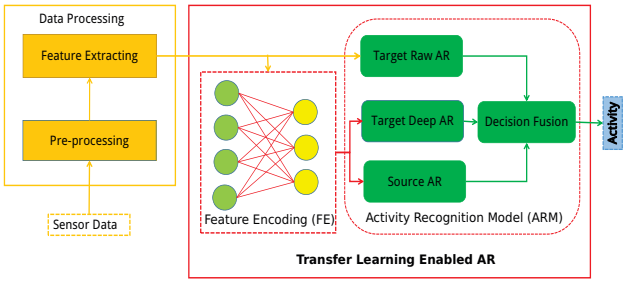


Fig. 2: Overall System Architecture

the source and target domains are related, which implies that the generated feature space between two domains has explicit or implicit relationship to each other.

For example, the source and target domains activity sets are {*'Sitting', 'Standing', 'Cooking', 'Eating'*} and {*'Sitting', 'Standing', 'Cooking', 'Biking', 'Jogging'*}, respectively and both the domains contain accelerometer sensor signal traces. In this scenario, target domain has two unseen activities and the total number of activity categories are imbalanced.

B. System Architecture

The overall *UnTran* architecture is shown in Fig. 2. *UnTran* consists of three main components.

Data Processing: This module filters raw sensor signals and then extracts low-level features from these pre-processed raw sensor data.

Feature Encoding: This module uses first two layers of source trained autoencoder and generates common features in the target domain. Data distribution differences are minimized by transferring the two layers from source domain that helps generate generic features. These features are then used in the next module.

Activity Recognition Model: In this module, we fuse the knowledge of one source domain AR model and two target domain AR models. Source domain labeled samples are passed through the feature encoder and then encoded features are used to train classifiers. In the target domain, one classifier is trained with deep features and other classifier is trained with low-level raw features. Finally, we fuse the knowledge of these three AR models to infer activities in the target domain.

IV. System Design and Algorithm

In this section, we discuss the details of our activity recognition framework.

A. Data Processing

Our framework is agnostic and works with any kinds of sensor signals. In this work, we use accelerometer sensor signals to demonstrate the effectiveness of our proposed framework. The collected sensor signals for the activities are noisy and need to be processed before the activity recognition process. We processed our sensor signals in two steps -*i)* Data Preprocessing, and *ii)* Feature Extracting. In the data processing step, collected raw sensor data is filtered using a low-pass median filter. We determine the band of the filter by applying FFT on the data. This filtered data is then used to create frames. We created each frame in a fixed-width sliding window having a length of 50% overlap per frame. In the feature extracting step, the previously generated frames are used to compute various statistical and frequency domain features. Time domain features like mean, standard deviation etc., and frequency domain features like energy, entropy etc., of the signals, are calculated using Fast Fourier Transform (FFT) on each frame. We normalized the computed features which is then fed into the feature encoding process, that helped reduce the training time of the autoencoder.

B. Feature Encoding (FE):

Autoencoder (AE) is a feed-forward neural network that contains an input layer, an output layer, and one or more intermediate hidden layers between them [29] [30]. Autoencoder contains two processes - i) encoding, and ii) decoding. Given an input $\mathbf{x}$, autoencoder encodes this input through four layers encoding process and then feed this encoded output as input to the decoding process to generate an output $\bar{\mathbf{x}}$. In this work, we use four layers deep autoencoder for feature encoding. Mathematically, the encoding and decoding processes of the deep autoencoder are represented as follows.

Encoding Layers:

$$\begin{aligned}\mathbf{h}_i^{(1)} &= f(\mathbf{W_1}\mathbf{x}_i^{(1)} + \mathbf{b}_1),\\ \mathbf{h}_i^{(2)} &= f(\mathbf{W}_2\mathbf{h}_i^{(1)} + \mathbf{b}_2),\\ \mathbf{h}_i^{(3)} &= f(\mathbf{W}_3\mathbf{h}_i^{(2)} + \mathbf{b}_3)\\ \mathbf{h}_i^{(4)} &= f(\mathbf{W}_4\mathbf{h}_i^{(3)} + \mathbf{b}_4)\end{aligned} \tag{1}$$

Decoding Layers:

$$\begin{aligned}\bar{\mathbf{h}}_i^{(4)} &= f(\mathbf{W}'_4\mathbf{h}_i^{(4)} + \mathbf{b}'_4),\\ \bar{\mathbf{h}}_i^{(3)} &= f(\mathbf{W}'_3\bar{\mathbf{h}}_i^{(4)} + \mathbf{b}'_3),\\ \bar{\mathbf{h}}_i^{(2)} &= f(\mathbf{W}'_2\bar{\mathbf{h}}_i^{(3)} + \mathbf{b}'_2),\\ \bar{\mathbf{h}}_i^{(1)} &= f(\mathbf{W}'_1\bar{\mathbf{h}}_i^{(2)} + \mathbf{b}'_1)\\ \bar{x}_i &= \bar{h}_i^{(1)}\end{aligned} \tag{2}$$

where $f(.)$ is a nonlinear activation function. We use sigmoid function as a nonlinear activation function.

Autoencoder helps discover activity patterns by compressing the sensor signals (x) in the encoder then decompress the output of the encoder to generate an output which is similar to the sensor signal $(\vec{x})$. However, this compression process generates low-dimensional features which is similar to PCA [31]. The disadvantage of this feature discovering process is that the hidden layers' dimension must be kept smaller than the encoder input dimension. As a result, reconstructing similar output as the raw sensor signals in the decoding process becomes challenging. Therefore, we employ low-level features as an autoencoder input. However, this extracted features reduces the input dimension, hence implicitly restricts the number of neurons in each hidden layer that results to

a low-dimensional PCA feature that hinders in finding the generalized feature space. We, therefore, use a sparse hidden layer as the first hidden layer and feed low-level features into this layer. The dimension of our sparse layer is larger than the raw features dimension and in order to get the meaningful feature representation, we add sparse constraints in this layer. Additional three layers are used to establish non-linear correlation among the activities. We named this modified autoencoder as Deep Sparse Autoencoder (DSAE). Our DSAE learns weights matrices and bias vectors of the hidden layers by minimizing the following reconstruction error.

$$J_{aen}(W,b) = \min_{\mathbf{W},\mathbf{b}} ||\mathbf{x} - \bar{\mathbf{x}}||^2 + \alpha \sum_{i=1}^{N_{L_1}} \Phi_{kl}(\rho||\hat{\rho}_i) \quad (3)$$

The first term of Eqn. 3 represents the reconstruction cost of our DSAE where $\mathbf{W}$ and $\mathbf{b}$ denote weights and biases of encoding and decoding layers, respectively. the second term, of Eqn. 3 represents Kullback-Leibler (KL) divergence between the sparsity constraint ρ and average activation $\hat{\rho}$ of the first hidden layer. The average activation of a hidden unit, j is computed as follows.

$$\hat{\rho_j} = \frac{1}{m} \sum_{i=1}^{m} \left[a_j^{L_1} x_i \right] \quad (4)$$

where m denotes the number of low-level feature inputs. We employ stochastic gradient descent (SGD) [32] method to determine the changes of weights and biases and update the network parameters accordingly.

We assume that our source domain has a large number of labeled activity samples. DSAE helps learn inherent activity characteristics in an unsupervised fashion. Establishing the correlation between the activity and corresponding features requires tuning the network parameter with respect to the activity class. Hence, we append a softmax layer at the end of the encoding layer to encode class labels in the source domain. To train this source domain classifier, we use the following cross-entropy objective function.

$$\min_{\theta} \left(-\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \mathbf{1}\{y_i = j\} \log \frac{e^{\theta_j^T x_i}}{\sum_{l=1}^{k} e^{\theta_l^T x_i}} \right) \quad (5)$$

where $\mathbf{1}(.)$ is an indicator function and provides 1 when the condition is true otherwise 0. We employ stochastic gradient descent (SGD) [32] method to tune the network parameters.

The performance of the source trained classifier degrades while deploying in the target domain due the marginal distributions of the data between two domains and unseen activity samples. Lower layers of this source domain network produce most generic features and higher layers (closer to classifier layer) generates most domain-specific features [8]. We transfer the first two layers of the source trained network to produce the common feature space in the target domain. We choose the first two layers due to its capability of generating generic features and preserving domain information. Selection of the number of layers can also be performed empirically. We also see that our assumption holds empirically (details are presented in section V-I). Target domain activity instances are then fed into this partial network to produce most common features space. Fig. 3 represents this transfer learning enabled feature encoding process.

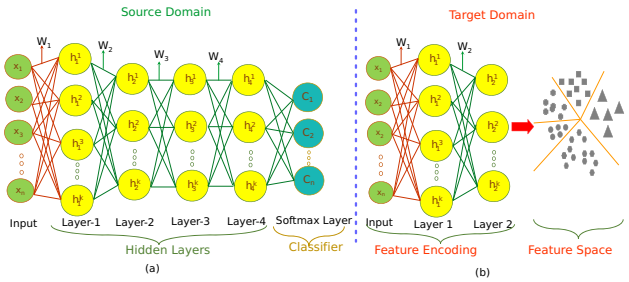


Fig. 3: Common Feature Space Generation

C. Activity Recognition Model (ARM)

Our activity recognition model consists three classifiers- one source domain classifier and two target domain classifiers. Source domain annotated activity instances are fed into the feature encoding layers (first two layers as shown in Fig. 3 (a)) to produce common features. These generated features can be used to train any standard classifier. Due to the optimal implementation, we use support vector machine (SVM) [33]. SVM finds the maximum margin hyperplane $w.x - b = 0$, that maximizes the distance between the activity instances and hyperplane by minimizing the following problem.

$$\begin{aligned} \underset{(w,b)}{\text{minimize}} \quad & \frac{1}{2} w w^T + C \sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, and\ \xi_i \geq 0, \forall_i. \end{aligned}$$

where x_i and y_i represents the ith the feature vector and activity label, respectively. $\phi(.)$ corresponds to kernel function that transforms the separable feature space. The parameter, ξ_i represents the degree of false classification. Capacity constant, C controls overfitting and error of the classifier.

Generated source domain deep features are used to train a SVM classifier and we named it as 'Source AR'. Target domain low-level features are also fed into the feature encoder to produce deep features and which will be used to train a SVM classifier in the target domain and we named it as 'Target Deep AR'. We also train a SVM classifier with low-level features in the target domain and named it as 'Target Raw AR'. Since the target domain has a few number of labeled activity samples, we fuse these three models to overcome the insufficient labeled data problem in the target domain. Before fusing these classifiers, we determine whether an activity sample belongs to an existing class or a new class. To determine if an activity sample belongs to a new class, we train one-class SVM with all the source domain labeled samples as the seen class. If a sample is outside of the source

activity distributions then it is detected as a new class. This novelty detector helps formulate fusion function.

The target domain classifier model trained with a few number of labeled activity samples underestimates the class conditional probability, $P(y_i|x)$, because a limited sample covers a smaller feature subspace compared to the true feature subspace that can be covered by all the activity labeled samples in the target domain. Due to this smaller probability, AR model fails to detect correct activity class for many instances. The class probabilities of our AR model is computed as follows.

$$P(y_i|x) = \frac{1}{1 + \exp(Af(x) + B)} \tag{6}$$

where $f(x)$ denotes the signed distance of the input feature vector to the hyperplane. The parameters, A and B are estimated using maximum likelihood estimation from the labeled training activity instances during the training period of the classifier.

Our DSAE implicitly reduces the distribution distances between source and target domain, therefore we use both source and target domain classifier to infer activities. We propose a scoring function that combines the source and target domain classifiers' inference knowledge and helps overcome the biased probability estimates. Our novelty detector helps determine new activities in the target domain and also helps formulating our fusion function. Our fusion function is formulated as follows.

$$\phi(y|x) = \begin{cases} P_s(y|x) + P_d(y|x), & \text{if } (y_d = y_r) \\ max(P_s(y|x), P_d(y|x)), & \text{else if } (y_s = y_d) \\ P_s(y|x) \times P_r(y|x) \times P_d(y|x), & \text{otherwise} \end{cases}$$

In case of new activities detected by novelty detector, we define the following fusion function.

$$\phi(y|x) = \begin{cases} P_r(y|x) + P_d(y|x), & \text{if } (y_d = y_r) \\ max(P_d(y|x), P_r(y|x)), & \text{otherwise} \end{cases}$$

where $P_s(y|x)$, $P_d(y|x)$ and $P_r(y|x)$ represent source trained classifier probability, target domain deep feature trained classifier probability and raw feature trained classifier probability, respectively. y_d, y_r and y_s denotes the output class of 'Source AR', 'Target Deep AR' and 'Target Raw AR' models predicted classes respectively. Feature encoding process does not minimize data distributions explicitly. Hence, novelty detector may also falsely classify few existing activity samples as new activity or vice versa. We overcome this challenge by adding P_r and P_d together in both existing or new activity detection process and this help improving the prediction probability even though they have lower individual probabilities. We multiply source, and target domain prediction probabilities when all classifiers predict similar activity classes because source AR model prediction probability is usually much higher than that of target domain AR model. Upon determining the combined probability using fusion model, class-labels with the highest probability represents the activity class and it is represented as follows.

$$y^* = \arg\max_y \phi(y|x) \tag{7}$$

V. Experimental Evaluation

In this section, we discuss the details of our experiments.

A. DataSets Description

We validate our proposed activity recognition framework, *UnTran* with three publicly available datasets traces. We use accelerometer sensor signal traces from these datasets. The dataset descriptions are discussed below.

i) Opportunity dataset (Opp) [34][35] contains naturalistic 17 activities of daily living (ADL) from four participants. The activities include drinking, cleaning table, eating sandwich etc. Data was recorded at 64 Hz for about 6 hours of recording from 5 Inertial Measurement Unit (IMU) on the upper limbs and torso comprising of 3D accelerometers, 3D gyroscope and 3D magnetic field sensor. We consider 10 activities and use only accelerometer sensors data to evaluate our framework.

ii) WISDM Actitracker dataset (Wisdm) [36] contains 6 distinctive human activities including walking, jogging, sitting etc. belongs to 29 users. Data was collected at 20 Hz using a smartphone accelerometer sensor kept on front pants leg pocket.

iii) Daily and Sports dataset (Das) [37] containing 19 activities performed naturally by 8 subjects. Data was collected at 25 Hz sampling frequency. Each activity duration was 5 min for each subject. The activity set includes sitting, playing basketball, cycling etc. Five motion tracker (MTx) units were used to collect the activity dataset where each MTx unit contains 3D accelerometer, 3D gyroscope, and 3D magnetometer sensors. MTx units were placed on the torso, right arm, left arm, right leg and left leg.

B. Baseline Methods

We compare our proposed *UnTran* framework with the state-of-the-art transfer learning based classifiers such as Transfer Component Analysis (TCA) [38], and Joint Distribution Adaptation (JDA)[39].

C. Implementation Details

We implemented our framework using python based deep learning platform, Tensorflow [40]. Accelerometer sensor data was segmented into 128 samples frames, with 50% overlap between successive frames. Frames were filtered with low-pass median filter to remove noises. We extracted various statistical time- and frequency-domain features, which were then fed into the classifier in batches, with a batch size of 32. We kept the frame length and batch size consistent across all datasets and experiments. We implemented transfer learning baseline methods, TCA with python and JDA using MATLAB. Our DSAE comprised of four layers. In addition, a softmax layer was added to encode the class labels in the source domain. We used the first two layers of the source tuned network to

Dataset	Source Domain	Target Domain
Opp	3	1
DAS	6	2
WISDM	21	8

TABLE I: Number of users in the source and target domain

build our classifier. We ran our *UnTran* framework on a server equipped with four NVIDIA GTX 1080-Ti GPUs and 64 GB memory with Intel Core i7-6850K processor.

D. Evaluation Methodology

We evaluated our *UnTran* framework with standard leave-two-sample-out cross validation method [18]. We train the target domain model with $(n-2)$ samples and rest of the two samples are used to test against the trained model. We repeat this process $\binom{n}{2}$ times and report the average result.

E. Performance Metrics

We evaluated and compared the performance of our framework based on the following metrics. *i)* Precision $P = (\frac{TP}{TP+FP})$, *ii)* Recall $R = (\frac{TP}{TP+FN})$, *iii)* F-1 Score $= \frac{2\times P\times R}{P+R}$ and, *iv)* Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$, where TP, FP, TN, and FN are the number of instances of true positive, false positive, true negative and false negative, respectively.

F. Experimental Results

In this work, we partition each dataset into two groups and each group contains distinct users. Table I shows the number of users in the source and target domain for each of the dataset. We randomly choose users to generate the source and target domain. We evaluate the performance of our proposed *UnTran* framework for the following settings *i)* Influence of balanced classes (i.e., how *UnTran* performs if both the source and target domains have same number of activities), and *ii)* Influence of imbalanced classes (i.e., how *UnTran* performs if target domain contains larger number of activities).

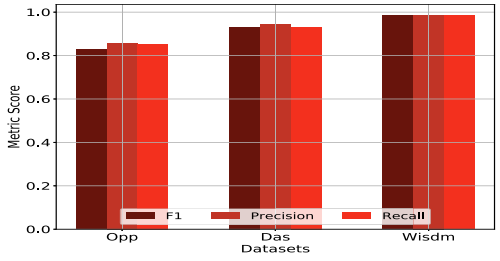


Fig. 4: *UnTran* performance on seen activities

G. Influence of Balanced Activities

In this experiment, both the source and target domain contain equivalent number of activities but target domain contains unseen activities too.

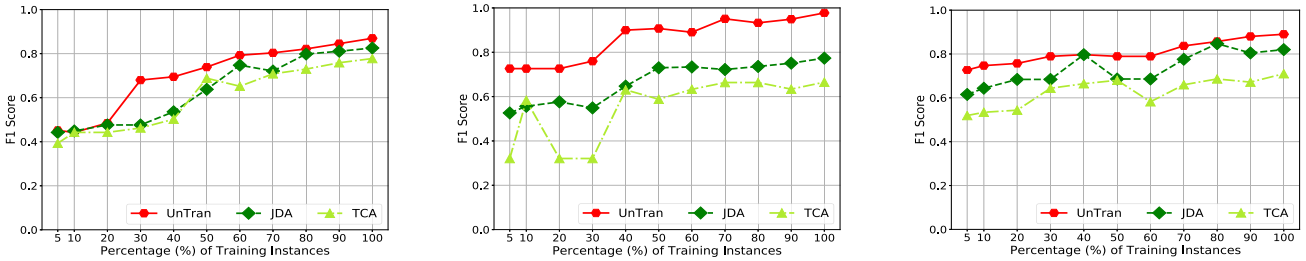
1) Seen Activities: We conduct this experiment to demonstrate our frameworks' efficacy while both source and target domain contain similar activities. Our framework comprises of supervised SVM model, therefore we encode label information with only 2-3% labeled activity samples in the target domain. We evaluated our framework with leave-two-sample-out cross-validation technique as stated before. Fig. 4 represents our framework's performance on three datasets. We see that our framework achieves F1 score of 0.82, 0.85, 0.98 for Opp, Das and Wisdm dataset, respectively. Note that Opp dataset achieves lower f1 score compared to other dataset due to the larger data distributions difference which is caused by the diverse set of activity classes and sensing biases. On the other hand, Wisdm dataset shows higher F1 score because it contains smaller number of similar type of activities (only six) hence this dataset has less data distributions divergence.

2) Unseen Activities: In this settings, we evaluated our model performance in the presence of new activities in the target domain. We vary the number of new activities in the target domain while maintaining the constraint of same number of activities in both domain. We also study how the performance of *UnTran* framework is affected by varying the number of labeled data, in the target domain.

Varying amount of labeled data: To study this, we systematically varied the amount of labeled activity data in the target domain and computed the average F1 score of our *UnTran* framework. We varied the labeled data of our (n-2) activity samples in the target domain and rest of the remaining two samples were used to test the performance of our framework. Percentage of labeled activity samples were chosen at random from the (n-2) samples. We also varied the number of unseen activities from one to five for OPP and DAS datasets and one to three for WISDM dataset. We computed the average and reported the results. In this case, the alternative classifier (TCA, JDA) also undergoes the same techniques and is trained with the equivalent amount of labeled training data in the target domain.

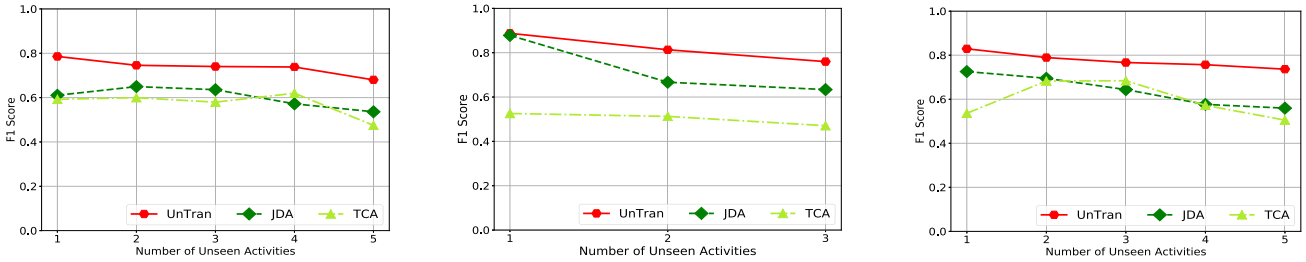
Fig. 5 reports the average results of the varying amount of labeled data while the source and target domain contains equivalent number of activity classes. We notice that our framework performance improves with the increase in the labeled activity samples. Our framework shows reasonable performance with only 20-30% of labeled samples compared to TCA and JDA. In case of opportunistic dataset, the performance is closer to JDA due to larger data distributions between source and target. Fig. 5b shows that our model achieves a performance gain of 12-15% because the wisdm dataset has a lesser number of closely related unseen activities. Our feature encoder was able to establish a better correlation among the extracted features and activities. In the presence of a large number of heterogeneous activities and diverse settings, our framework achieved a performance gain 2-4%. Fig. 5a , and 5c reports the impact of activity and environmental setting heterogeneities.

Varying the number of unseen activities: We evaluated our models efficacy in the presence of varying number of



(a) Opportunistic (b) Wisdm (c) Daily and Sports

Fig. 5: *UnTran* performance (Varying amount of labeled data) in presence of equivalent number of activities in both source and target domain



(a) Opportunistic (b) Wisdm (c) Daily and Sports

Fig. 6: *UnTran* performance (Varying number of unseen activities) in presence of equivalent number of activities in the source and target domain

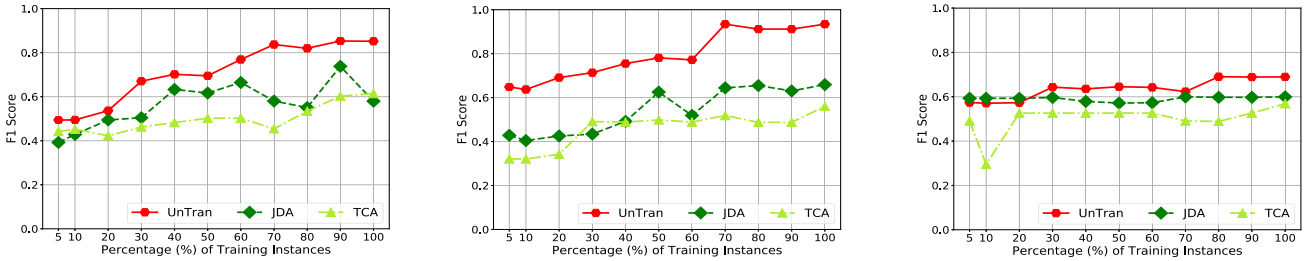
unseen activities in the target domain. We varied the number of unseen activities from one to A (number of activities) and followed the same leave-two-samples-out cross validation. Our framework performs reasonably well with 20-30% labeled activity samples. Therefore, we use 30% labeled activity samples to train our model in the target domain. Fig. 6 represents the experimental results for all three dataset. We noticed that our models overall performance drops 5-12% with the increasing number of unseen activities in the target domain. On the other hand, our model achieves performance gain of 10-13% compared to TCA and JDA because of the capability of utilizing label information from the source to the target domain. Both TCA and JDA minimizes the data distribution divergence explicitly and when TCA model is trained with the labeled data it performs similarly as JDA. Hence, the performance of TCA and JDA are close to each other. However, our framework shows supremacy due the knowledge fusion across the source and target domain.

H. Influence of Imbalanced Activities

We examine the performance of our framework when target domain contains both the existing source activities and the new activities. Hence, in this setting, our target domain model contains larger number of activity classes. Basically, we are interested to see whether our framework is able to find and learn any relationship from the existing activities and recognize new activities in the target domain. Our DSAE model is trained with five activities (five for opp and das, three for wisdm) in the source domain. We used the first two layers to produce common features in the target domain. Generated features are then used to train SVM model in the target domain.

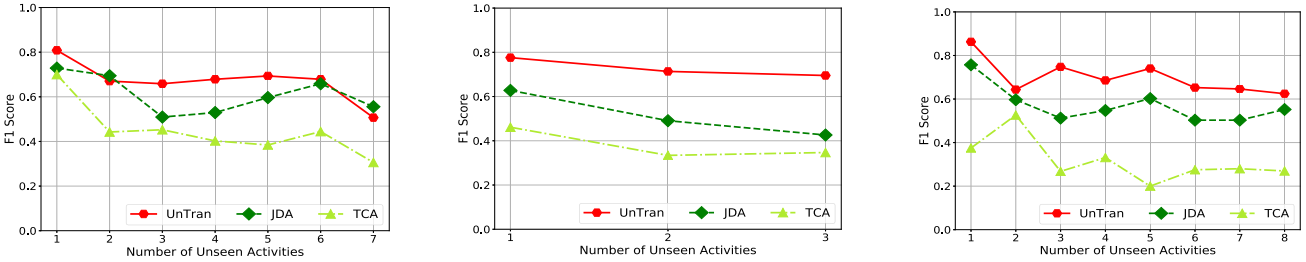
Varying labeled data: In this experiment, we vary the amount of labeled activity samples to train the target domain AR model. We choose (n-2) samples to train our 'Target raw AR' and 'Target deep AR' model and rest of the two samples test against this trained model. Percentage of labeled data of the (n-2) trained samples are chosen at random while training the target domain AR model. We used 30% labeled activity samples in this experiment. Further, we used leave-two-class-out cross validation technique, our (A-2) activity classes participated in the model training and rest of the two new activity classes participate in the test phase. It is worth noted that our target domain activity set contains all the existing activities of the source domain, and in addition, it also contains new activities.

Fig. 7 represents the performance of this experiment. We observe that our framework achieves performance gain of 3-5% for opportunistic and daily and sports dataset compared to the standard state-of-the-art transfer learning classifier. Our framework achieves 10-12% performance improvement on wisdm dataset compared to other TCA and JDA. For wisdm dataset (exercising activities), source domain contains three activities (i.e., *'Sitting', 'Standing', 'Walking'* and the target domain contains one or more new activities like *'Upstairs', 'Downstairs' and 'Jogging'*. These new activities are closely related with the source domain activities. For example, 'Jog-



(a) Opportunistic (b) Wisdm (c) Daily and Sports

Fig. 7: *UnTran* performance (Varying amount of labeled data) in presence of imbalance activities in the target domain



(a) Opportunistic (b) Wisdm (c) Daily and Sports

Fig. 8: *UnTran* performance (Varying the number of unseen classes) in presence of imbalance activities in the target domain

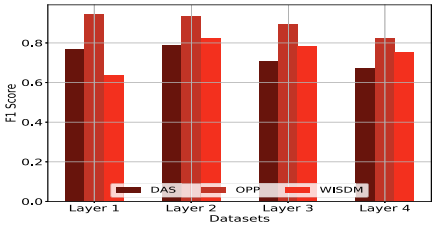


Fig. 9: *UnTran* performance on different layers

ging' and 'Walking' have both hand and leg movements. Therefore, our feature encoding model able to find more correlated features and establish a non-linear correlation among the activities and help improve the performance of our AR framework.

Varying the number of unseen activities: We extended our previous experiment, by varying the number of unseen activities in the target domain. In this setting, 30% annotated activity samples were used to train the 'Target raw AR' and 'Target deep AR' model. Fig. 8 represents our models performance in the presence of imbalance activities in the target domain. We observed that the performance dropped 15-20% with the increasing number of unseen activities. Our fusion framework incorrectly classifies a few number of new activities as existing activities because our feature encoding module encounters difficulties to generate distinct, separable common feature space for these activity samples. From Fig. 8, we notice that our framework achieve F1 score about 70% on average even in the presence of a large number of new activities in the target domain.

I. Performance Analysis of Deep Features

In this section, we examine, how our *UnTran* framework performs while we use features from different layers. The performance of our model is evaluated with a fixed number of unseen activities in the target domain for all the three real-world datasets. We use source trained classifiers' different layers to generate deep features in the target domain. Fig. 9 shows the leave-two-sample-out cross-validation result. Deep neural network generates most generic features in the lower layers and domain specific features in the upper layers. Fig. 9 reflects this characteristics. Note that the performance of our model decreases as approaches the upper layer of the network. Most generic features (Layer 1) are unable to distinguish activities while most specific features (Layer 4) are unable to generate a common feature space in the target domain. Therefore, we choose the first two layers to generate features and recognize activities in the target domain.

VI. Discussion

Our proposed deep sparse autoencoder based transfer learning enabled activity recognition framework, *UnTran* addresses a significant promising problem of unseen activity detection. There are however additional issues that need to be investigated.

Device and Sensor Diversity: We evaluated our framework, with only using wearable accelerometer sensors data. Though performance examination against three public datasets implicitly attests efficacy of our framework against users,

environment heterogeneities and sensing biasness. However, additional investigations are required when activity signals are collected through heterogeneous sensors (i.e., camera, PIR, etc.) and devices (smartwatch, smartnecklace, etc.).

Explicit Structural Patterns Mapping: Deep sparse autoencoder learns inherent latent features and is able to establish a correlation among the activities automatically. Transferring first few layers from the source to target domain helps generate common features space. We assume that features generated by the source trained layer implicitly reduces domain divergence automatically. However, for a large number of unseen and non-correlated activities, this implicit domain divergence minimization may be minimal and the performance of our framework degrades. To further improve the performance of our *UnTran*, a potential research direction, and our ongoing work is to incorporate structural correlation mapping among the intra- and inter-activities between the source and target domain.

Annotation Effort: We utilize source domain activity labels in our *UnTran* fusion AR framework to reduce labeled data in the target domain. However, we assume that users provide a few annotated samples in the target domain at random. To improve the performance and reduce the annotation costs associated with the number of activity samples can be further studied. One possible future direction is to employ active-learning based annotation technique.

VII. Conclusion

Human behavior and activity recognition in the smart environment have versatile application in healthcare, sports analytics, physical and cybersecurity domains. In this paper, we propose transfer learning enabled activity recognition approach that helps to infer new activities in the new environment. We envision that future smart environment will be very diverse in terms of users activity patterns, new sensing devices and their communication mediums. One of the most challenging task of activity learning is to recognize new activities in the target environment. Therefore, we advocate a novel activity recognition framework, *UnTran* to learn and recognize new activities in the target domain. We exploit the adaptability and scalability of deep sparse autoencoder in the target domain and fuse the deep and raw activity models both from source and target domain to deal with limited training samples. We attest the efficacy of our proposed *UnTran* framework with real data traces and compare its performance with several state-of-the-art transfer learning methods. We believe that our proposed adaptable and scalable activity recognition framework, UnTran will help advance human behavior and activity inference in large-scale diverse environments.

VIII. Acknowledgment

This research is partially supported by the ONR under grant N00014-15-1-2229.

References

[1] Nicky Kern, Bernt Schiele, and Albrecht Schmidt. Multi-sensor activity context detection for wearable computing. Springer.

[2] Oscar D Lara and Miguel A Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys and Tutorials*, 15(3):1192–1209, 2013.

[3] Xing Su, Hanghang Tong, and Ping Ji. Activity recognition with smartphone sensors. *Tsinghua Science and Technology*, 19(3):235–249, 2014.

[4] Jianbo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiao Li Li, and Shonali Krishnaswamy. Deep convolutional neural networks on multichannel time series for human activity recognition. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[5] Emmanuel Munguia Tapia, Stephen S Intille, and Kent Larson. Activity recognition in the home using simple and ubiquitous sensors. In *Pervasive*, volume 4, pages 158–175. Springer, 2004.

[6] Francisco Javier Ordóñez Morales and Daniel Roggen. Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers*, pages 92–99. ACM, 2016.

[7] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014.

[8] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3320–3328. Curran Associates, Inc., 2014.

[9] Le T. Nguyen, Ming Zeng, Patrick Tague, and Joy Zhang. Recognizing new activities with limited training data. In *Proceedings of the 2015 ACM International Symposium on Wearable Computers*, ISWC '15, pages 67–74, New York, NY, USA, 2015. ACM.

[10] Jorge-L Reyes-Ortiz, Luca Oneto, Albert Sama, Xavier Parra, and Davide Anguita. Transition-aware human activity recognition using smartphones. *Neurocomputing*, 171:754–767, 2016.

[11] Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten, and Paul JM Havinga. Complex human activity recognition using smartphone and wrist-worn motion sensors. *Sensors*, 16(4):426, 2016.

[12] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, 46(3):33, 2014.

[13] Yoshua Bengio. Deep learning of representations: Looking forward. In *International Conference on Statistical Language and Speech Processing*, pages 1–37. Springer, 2013.

[14] Sourav Bhattacharya, Petteri Nurmi, Nils Hammerla, and Thomas Plötz. Using unlabeled data in a sparse-coding framework for human activity recognition. *Pervasive and Mobile Computing*, 15:242–262, 2014.

[15] Yongjin Kwon, Kyuchang Kang, and Changseok Bae. Unsupervised learning for human activity recognition using smartphone sensors. *Expert Systems with Applications*, 41(14):6067–6074, 2014.

[16] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[17] Young-Seol Lee and Sung-Bae Cho. Activity recognition with android phone using mixture-of-experts co-trained with labeled and unlabeled data. *Neurocomputing*, 126:106–115, 2014.

[18] Heng-Tze Cheng, Feng-Tso Sun, Martin Griss, Paul Davis, Jianguo Li, and Di You. Nuactiv: Recognizing unseen new activities using semantic attribute-based learning. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 361–374. ACM, 2013.

[19] Song Cao and Ram Nevatia. Exploring deep learning based solutions in fine grained activity recognition in the wild. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 384–389. IEEE, 2016.

[20] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. Deep learning for sensor-based activity recognition: A survey. *arXiv preprint arXiv:1707.03502*, 2017.

[21] Thomas Plötz, Nils Y. Hammerla, and Patrick Olivier. Feature learning for activity recognition in ubiquitous computing. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, pages 1729–1734. AAAI Press, 2011.

[22] Charissa Ann Ronao and Sung-Bae Cho. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Systems with Applications*, 59:235–244, 2016.

[23] Bandar Almaslukh, Jalal AlMuhtadi, and Abdelmonim Artoli. An effective deep autoencoder approach for online smartphone-based human activity recognition. *International Journal of Computer Science and Network Security (IJCSNS)*, 17(4):160, 2017.

[24] Wenchao Jiang and Zhaozheng Yin. Human activity recognition using wearable sensors by deep convolutional neural networks. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1307–1310. ACM, 2015.

[25] Kyle D. Feuz and Diane J. Cook. Collegial activity learning between heterogeneous sensors. *Knowl. Inf. Syst.*, 53(2):337–364, November 2017.

[26] Ramin Fallahzadeh and Hassan 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*, ICCPS '17, pages 293–302, New York, NY, USA, 2017. ACM.

[27] Diane Cook, Kyle D Feuz, and Narayanan C Krishnan. Transfer learning for activity recognition: A survey. *Knowledge and information systems*, 36(3):537–556, 2013.

[28] Alberto Calatroni, Daniel Roggen, and Gerhard Tröster. Automatic transfer of activity recognition capabilities between body-worn motion sensors: Training newcomers to recognize locomotion. In *Eighth International Conference on Networked Sensing Systems (INSS'11)*, Penghu, Taiwan, June 2011.

[29] Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends in Machine Learning*, 2(1):1–127, 2009.

[30] Fuzhen Zhuang, Xiaohu Cheng, Ping Luo, Sinno Jialin Pan, and Qing He. Supervised representation learning: Transfer learning with deep autoencoders. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 4119–4125. AAAI Press, 2015.

[31] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[32] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[33] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[34] Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digumarti, Gerhard Tröster, José del R Millán, and Daniel Roggen. The opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters*, 34(15):2033–2042, 2013.

[35] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, et al. Collecting complex activity datasets in highly rich networked sensor environments. In *Networked Sensing Systems (INSS), 2010 Seventh International Conference on*, pages 233–240. IEEE, 2010.

[36] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

[37] Kerem Altun, Billur Barshan, and Orkun Tunçel. Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recognition*, 43(10):3605–3620, 2010.

[38] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.

[39] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2200–2207, 2013.

[40] Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.