M. A. Al Hafiz Khan and N. Roy, "Cross-Domain Unseen Activity Recognition Using Transfer Learning," *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2022, pp. 684-693, doi: 10.1109/COMPSAC54236.2022.00117.

https://doi.org/10.1109/COMPSAC54236.2022.00117

Cross-Domain Unseen Activity Recognition Using Transfer Learning

Md Abdullah Al Hafiz Khan
Department of Computer Science
Kennesaw State University
Marietta, United States
mkhan74@kennesaw.edu

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

***Abstract*—Activity Recognition (AR) models perform well with a large number of available training instances. However, in the presence of sensor heterogeneity, sensing biasness and variability of human behaviors and activities and unseen activity classes pose key challenges to adopting and scaling these pre-trained activity recognition models in the new environment. These challenging unseen activities recognition problems are addressed by applying transfer learning techniques that leverage a limited number of annotated samples and utilize the inherent structural patterns among activities within and across the source and target domains. This work proposes a novel AR framework that uses the pre-trained deep autoencoder model and generates features from source and target activity samples. Furthermore, this AR framework establishes correlations among activities between the source and target domain by exploiting intra- and inter-class knowledge transfer to mitigate the number of labeled samples and recognize unseen activities in the target domain. We validated the efficacy and effectiveness of our *AR* framework with three real-world data traces (Daily and Sports, Opportunistic, and Wisdm) that contain 41 users and 26 activities in total. Our AR framework achieves performance gains ≈ 5-6% with 111, 18, and 70 activity samples (20% annotated samples) for Das, Opp, and Wisdm datasets. In addition, our proposed AR framework requires 56, 8, and 35 fewer activity samples (10% fewer annotated examples) for Das, Opp, and Wisdm, respectively, compared to the state-of-the-art *Untran* model.**

***Index Terms*—Activity Recognition, Transfer Learning, HAR, Heterogeneous Learning,Imbalanced Activity Recognition, Pervasive Computing.**

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]. The proliferation of smart sensing devices (i.e., smartphone, smartwatch, etc.) integrated with various sensors (e.g., accelerometer, gyroscope, etc.) help develop applications related to health care monitoring, rehabilitation system, interactive gaming, etc., have constantly been evolving to improve the human-centric services in the smart living environments. These activity recognition systems are typically built to recognize a predefined and limited set of activities (i.e., sitting, standing) using annotated sensor signals of body-part movements in a similar environment. These AR systems' performance degrades while recognizing similar activities in different environments due to variations like the same sensor with heterogeneous devices, sensor biases, the user's daily lifestyle and activity pattern, etc. To maintain the similar performance of the AR system in the deployed environment without building a new AR system and mitigate variations of the new target environment, we incorporate transfer learning techniques. More specifically, we employ transfer learning-enabled maximum mean discrepancy (MMD) to mitigate the variations in the target environment.

Usually, the AR model is trained on handcrafted features, requires domain expert knowledge, and depends specifically on the performed activities and environment. On the other hand, the deep learning techniques help extracts features automatically. However, deep learning techniques [4] are data-hungry processes and require computationally expensive resources and a large volume of annotated activity samples. Moreover, these deep models overfit in the presence of a limited amount of target domain activity samples. Therefore, we overcome these challenges by leveraging the unsupervised deep learning techniques and extracting the sensor signals' inherent characteristics.

A major aspect of the deep model is that it learns generic features in the beginning layers and the deep layer closer to the classifier learns domain-specific features [5]. [6] uses the inherent characteristics of the deep model and transfers a few layers of the source domain that helps reduce the computation time and transfer domain-dependent features in the target domain. However, the partial network (few layers) transfers lose the source domain's activity pattern knowledge that can be incorporated during model learning and improve the activity recognition performance in the target domain. Therefore, to maximize the relationship of the activity classes with the represented features, we keep all the layers from the source trained autoencoder-based classifier (except the classifier layer) and generated features in the source domains. As a result, the finetuning process also reduces computational time and maximizes source domain knowledge in the target environment.

A scalable and adaptable AR model can recognize new activities in the target environment to meet the application scenarios and user needs. The AR model can ask the users to provide annotated activity samples to learn new activities in the target environment. However, collecting numerous annotated examples and training a new AR model is not feasible. Instead

of employing limited annotated samples and utilizing existing knowledge from the source domain help maintain the AR model performance in the target domain. Therefore, we utilize existing domain knowledge from the source domain, minimize domain discrepancy in the target domain in an unsupervised way, and use limited annotated activity samples to build our target classifier. We summarize the key contributions of our work below.

- **Maximizing Existing Knowledge:** To maximize domain knowledge utilization, we exploit transfer-learning techniques. First, we train a deep autoencoder using source domain annotated samples. Second, this autoencoder model is finetuned with unannotated samples in the target domain. The benefits of this approach are that the finetuning process is entirely unsupervised.
- **Minimizing Domain Discrepancy:** Structural pattern mapping techniques help minimize intra-class (i.e., same activity in both domains) distance and maximize inter-class (i.e., two different activities) distances during domain discrepancy minimization process using Maximum Mean Discrepancy.
- **Extensive Evaluation with Limited Annotation**: We study the problem of limited annotation in the target domain and their challenges. To demonstrate the effectiveness and efficacy of our proposed approach, we conducted experiments on three real-world datasets.

II. Related Work

In the wearable pervasive computing era, the activity recognition (AR) models infer activities from various sensors using classical supervised machine learning approaches [7] [8] [2]. Classical AR are both data-driven and knowledge-based approaches. AR models heavily depend on domain knowledge-dependent handcrafted features and use machine learning algorithms [9]. Activity recognition models use annotated samples to train classifier algorithms in a supervised fashion and infer activities in the target environment. These supervised machine learning classifiers consider environment-specific settings and underperform in diverse environments where user activity patterns, sensing devices, and sensing biases are present [10] [2] [3]. These handcrafted features hinder the scalability of AR models. Therefore, automated features extraction is warranted. To automate the feature extraction process and reduce the dependency of domain expert knowledge, the researcher exploited the deep learning-based feature extraction process in the activity recognition domain [11] [12] [13]. These approaches learn hidden activity patterns from the sensor data traces and discover meaningful patterns without the human intervention [14] [15] [16]. Deep learning models are data-hungry and also require a lot of annotation. Researchers also explored unsupervised deep learning methods that demand a large set of unlabeled training samples [13] [15]. These methods are computationally expensive and require a significant training time to adjust the network parameters. None of these approaches work well in the presence of limited annotated activity samples. Various deep learning models such as CNN, LSTM, etc. used to infer activities [17] [18] [17]. [18] proposed convolutional neural network-based activity recognition approach to learn activity patterns in a semi-supervised fashion and infer activites. [17] proposed ensembles of deep LSTM (Long short-term memory) recurrent neural network model to infer human activities from sensor signals. None of these approaches work well with limited annotated activity samples and consider traditional AR settings. These methods did not consider dataset diversity, different activity styles, etc. Domain discrepancy needs to minimize to create a scalable activity recognition model.

Transfer learning approaches help improve adaptability and scalability issues and reduce domain discrepancy. Recently a limited number of aspects of transfer learning-enabled activity recognition have been investigated in AR domain [19] [20] [21]. [20] proposed an uninformed transfer learning algorithm that helps 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 the availability of a large set of unlabeled data samples with similar activities in the target domain. [22] addressed the versatility of sensor modality and sensor position independence by transferring a similar set of activity labels from an existing trained sensor node to a new sensor node without any user intervention. In transductive transfer learning, a.k.a Domain Adaptation settings, similar classes present both domains, and the AR model learns parameters during the training phase from both domain data. This technique reduces the required annotated data in the target domain [23] [24] [25] [26] [27]. [25] proposed cross-domain domain transductive transfer techniques and minimized domain data distributions using MMD (maximum mean discrepancy) techniques and predict unlabeled images in the target domain. [28] proposed majority voting-based cross-subject transfer learning techniques that minimize the intra-class distance between the source and the target domain and infer activities. In this work, we consider both intra- and inter-class distance minimization. These AR models consider similar activities in both domains and can not recognize new activities in the target domain.

Unseen activity recognition approaches investigated in the recent past. To address recognizing unseen activities using unannotated data, researchers proposed various attribute-based approaches [29] [7]. These attribute-based activity recognition models assume that each activity has a unique set of attributes. The performance of these models degrades in the presence of existing and new activities. Though fusion-based models combine the attributes- and features-based models and improve performance in the target environment, these models failed to consider sensing baisness, activity patterns, variations, and user diversity in the targeted domain. In addition, these approaches also depend on the expert domain knowledge. To alleviate the annotated samples, researchers also investigated semi-supervised methods and learned classifier parameters using both labeled and unlabeled activity samples [30] [31]. However, these methods are error-prone and typically unable

to replace the need for ground-truth annotated data from experts. In an attempt to bootstrap an existing trained activity model, in this work, we advocate using a small subset of unlabeled samples in addition to a small subset of labeled activity samples in the target domain.

This work exploits the benefits of the existing pre-trained deep sparse autoencoder-enabled activity recognition model in the source domain to reduce the required samples in the target domain. Our proposed method minimized both within- and cross-domain inter-and intra-class distances. As a result, our framework can infer unseen activities in the presence of a limited number of annotated samples in the target domain. This work reduces this effort by transferring the knowledge from the source to the target domain autonomously by using deep transfer learning techniques. Our proposed AR framework helps mitigate the scarcity of labeled activity samples by utilizing label information from the source domain to the target domain.

III. The Proposed Activity Recognition Framework

We propose and design an AR Model for recognizing unseen activities in the presence of user activity patterns diversity, sensing biases, and limited activity samples in the target domain. We assume that the source domain has a significant amount of labeled activities samples and a pre-trained AR model. The proposed AR model constructs a common feature space where similar activity samples help generate similar feature space to tackle this problem. Fig. 1 represents the overview of our activity recognition approach.

Problem Statement

Mathematically we define our problem as follows. Let source domain training data $D_s = \{x_i^{(s)}, y_i^{(s)}\}_{i=1}^{Ns} = \{\mathbf{X}^s, \mathbf{y}^{(s)}\}$, 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 $D_t = \{x_j^{(t)}, y^{(t)}\} = \{\mathbf{X}^{(t)}, \mathbf{y}^{(t)}\}$ 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 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, the target domain has two unseen activities, and the total number of activity categories is imbalanced.

Framework Architecture

Our AR framework feed sensor signals to the autoencoder to generate deep features. Furthermore, the latent structural pattern mapping module minimizes feature discrepancy between the domains, and finally, we used these discrepancy minimized features to predict activity in the target domain. Fig. 2 shows the overall architecture of our AR framework.

Data Processing: We pass the accelerometer sensor signals through a low-ass median filter to filter the noise. Next, we determine the band of the filter by applying FFT to the data. Finally, each frame is created from the filtered accelerometer sensor signals using a fixed-width sliding window with 50% overlap per frame. These activity frames then pass through the auto-encoder to train and generate deep features later.

Deep Feature Encoding (DFE): Autoencoder, a feed-forward neural network [32] [33] help learn unannotated sensor signal features automatically in two steps - i) encoding and ii) decoding. Our proposed four layers deep autoencoder discovers activity patterns by first compressing the given input sensor signals $\mathbf{x}$ in the encoder step. Then the decoding step generates a similar output vector $\bar{\mathbf{x}}$ by decompressing it.

The encoder works similarly as PCA [34]. This compression (encoding) of the signals helps capture inherent features when the dimensions of the hidden layers are smaller than the input sensor signals. As a result, reconstructing similar output as the raw sensor signals in the decoding process becomes challenging. We, therefore, construct a sparse hidden layer as the first hidden layer by adding sparsity constraint and feed sensor signals into this layer. This technique helps maintain a larger dimension of our sparse layer to get the meaningful feature representation after the encoding step. Furthermore, the additional three layers of our encoder help capture non-linear correlations among the activities. We named this modified autoencoder as Deep Sparse Autoencoder (DSAE). Our DSAE learns the hidden layers' weights matrices and bias vectors by minimizing the following reconstruction error.

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

$$\hat{\rho_i} = \frac{1}{m} \sum_{i=1} a_i^{L_1} x_i \quad (2)$$

The first term of Eqn. 1 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. 1, represents Kullback-Leibler (KL) divergence between the sparsity constraint ρ (empirically set to 0.05 for all the neurons in the first layer) and average activation $\hat{\rho}$ of the first hidden layer. The average activation of a hidden unit, j is computed using Eqn. 2, where m denotes the number of sensor signal samples, and $a_j^{L_1}$ represents activation unit j of first layer (denoted as L_1). We employ the mini-batch gradient descent (MGD) [35] method to determine the changes of weights and biases and update the network parameters accordingly. This automatic feature extraction process learns features space without considering the distribution of activity labels.

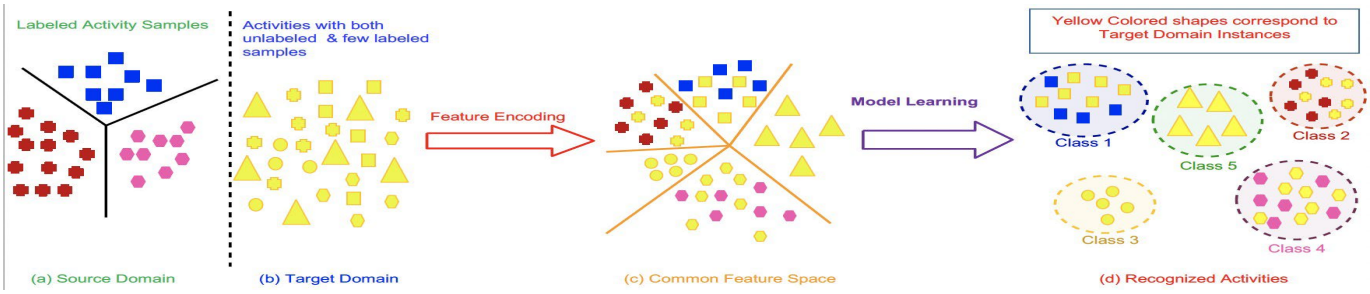


Fig. 1: Overview of our activity recognition approach. (a) Source domain labeled activity instances, (b) Target domain contains both unlabeled and few labeled activity instances, (c) Common feature space for classification, and (d) Resulting activities after classification. Note that different shapes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

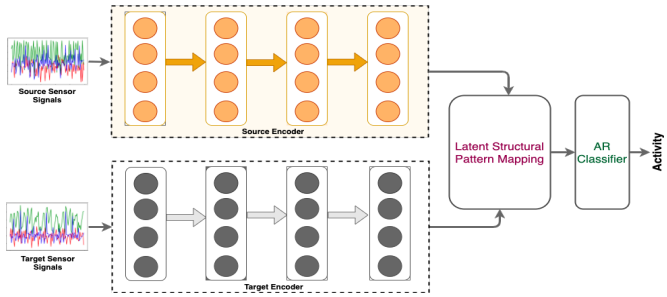


Fig. 2: Overall Framework Architecture.

Source Domain Knowledge Encoding: DSAE helps learn inherent activity characteristics unsupervised. However, establishing the correlation between the activity and corresponding features requires tuning the network parameter for the activity classes. Since the source domain has annotated activity samples, we append a softmax classifier at the encoding layer's end to encode class labels in the source domain and fine-tune our DASE model parameters. We use the following cross-entropy objective function to optimize this source domain classifier model.

$$\min_{\theta} \; -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k} \mathbf{1}\{y_i = j\} \log \frac{e^{\theta_j^T x_i}}{\sum_{l=1}^{k} e^{\theta_l^T x_i}} \tag{3}$$

where $\mathbf{1}(.)$ is an indicator function and provides 1 when the condition is true otherwise 0. θ denotes softmax classifier parameters - weights and biases. We employ the mini-batch gradient descent [36] method to tune the network parameters.

Target Domain Feature Extraction: The performance of the source trained classifier degrades while deploying in the target domain due to the marginal distributions of the data between two domains and unseen activity samples. We discard the softmax classifier portion, keep the source trained encoder (four layers), and generate features for the source and target domain sensor signals. The feature generation shown in Fig. 2. The generated target domain features show a discrepancy with the source for the similar activities. Therefore, we explicitly emphasize source domain knowledge and maximize this knowledge transfer by maximizing inter-class distance and minimizing intra-class distance by employing the structural pattern mapping technique and reducing the discrepancy between these two domains' feature space.

Latent Structural Pattern Mapping:

Our proposed DSAE generated features may maximize the intra-class distance and minimize inter-class distances in the target domain. Nevertheless, this problem space becomes complex in the presence of new activities. Therefore, we employ intra-class compactness and maximize inter-class separability that helps reduce the cross-domain discrepancy. During mapping the structural pattern between the domains, we also minimize the within-domain intra-class distance, cross-domain intra-class distance and maximize within-domain and cross-domain inter-class distances. Fig. 3 shows the detailed schematic diagram of our proposed approach. These DSAE generated feature spaces that are unable to establish the relationship between the source and target domain class labels and cannot utilize source domain class information to infer target domain activities. We explicitly incorporate the correlation among activities within and between the domains using intra-class compactness and inter-class separability information by projecting the DSAE generated features to separate kernel space by deriving the following matrix that considers these distances into account.

$$S_\phi = S_c - \gamma S_s = \begin{matrix} \\ \\ \end{matrix} \begin{matrix} \begin{matrix} \mathbf{I} & \mathbf{I} \end{matrix} \\ \begin{matrix} S^{(ss)} & S^{(st)} \\ S^{(ts)} & S^{(tt)} \end{matrix} \end{matrix} \tag{4}$$

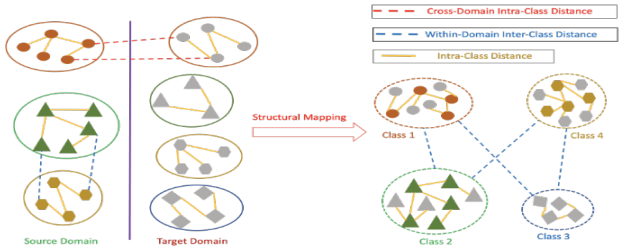


Fig. 3: Schematic diagram of structural pattern mapping. The blue dotted line represents within-domain inter-class distances. The deep yellow color line and the red dotted line depict intra-class class distances and intra-class distances, respectively. Note that different shapes correspond to different activities. After applying the structural pattern mapping techniques, we assume that both the source and target domain share similar feature spaces that minimize the intra-class distance and maximize the inter-class distance.

where γ is the model parameter that helps balances

between the intra-class compactness (S_c) and inter-class separability (S_s). The S_ϕ matrix contains both within ($S^{(ss)}$, $S^{(tt)}$) and cross-domain ($S^{(st)}$, $S^{(ts)}$) information and $S \in R^{(n_s+n_t)\times(n_s+n_t)}$. The major benefits of this matrix is that it ables to handle imbalanced classes between the domains and also considers the relationship among the instances explicitly. We define individual component $S_{c_{i,j}}$ and $S_{s_{i,j}}$ of S_c and S_s, respectively for every pair of instances $(\hat{x}_i, \hat{x}_j)$ for within-domain and across-domain. These individual component $S_{c_{i,j}}$ and $S_{s_{i,j}}$ is a scalar and defined as follows.

$$S_{c_{i,j}} = {}^{(c)}_{i,j}\, g(\phi(\hat{x}_i), \phi(\hat{x}_j)) \quad (5)$$

$$S^{s}_{i,j} = W^{(s)}_{i,j}\, g(\phi(\hat{x}^s_i), \phi(\hat{x}^s_j)) \quad (6)$$

where $W^{(c)}_{i,j}$ and $W^{(s)}_{i,j}$ represent the intra-class and inter-class nearest graph. We define these graph as follows.

$$W^{(c)}_{i,j}(\phi(\hat{x}_i), \phi(\hat{x}_j)) = \begin{cases} 1, & \text{if } y_i = y_j \\ 0, & \text{otherwise} \end{cases}$$

$$W^{(c)}_{i,j}(\phi(\hat{x}^s_i), \phi(\hat{x}^t_j)) = \begin{cases} 1, & \text{if } y_i = y_j \\ 0, & \text{otherwise} \end{cases}$$

We construct the within- and across-domain neighbor graph using the k-nearest neighbor classifier. Every instances of S matrix is generated using equation 5and 6. Source domain annotated activity samples are used to construct the neighbor graph by following the standard K-nearest algorithm and generating all instances of $S^{(ss)}$ matrix. Each of these matrices, $S^{(st)}$, $S^{(ts)}$, $S^{(tt)}$, require knowing both source and target domain annotation. The target domain pseudo labels are generated using similar techniques as joint distribution adaptation algorithm (JDA) [37]. Both pseudo-labels and provided limited annotation of the target activities and annotated sources activity samples are used to construct the nearest neighbor graph matrix. After constructing these neighbor graphs, we apply softmax function (g) over the cosine similarity distance (d_c) within- and across-domain to measure the similarity for each instances pair. We define the similarity function $g(\phi(\hat{x}^s_i), \phi(\hat{x}^t_j))$ as follows.

$$g(\phi(\hat{x}^s_i), \phi(\hat{x}^t_j)) = \frac{\exp^{(d_c(\phi(\hat{x}^s_i), \phi(\hat{x}^t_i))}}{\sum_{j=1}^{n_t} \exp^{(d_c(\phi(\hat{x}^s_i), \phi(\hat{x}^t_i))}} \quad (7)$$

The function g is normalized and computes the similarity score using the cosine distance function for each pair of instances. This function helps distinguish similar and dissimilar instances. S_ϕ matrix is formulated without explicitly minimizing conditional data distribution.

Minimize conditional data distribution: We adopt Maximum Mean Discrepancy (MMD) techniques to reduce the cross-domain conditional data discrepancy, embed cross-domain latent structures, and generate common feature subspace. Hence our optimization problem becomes as follows.

$$\sum_{i,j} \frac{S^{st}_{ij}}{S^{ss}_k S^{tt}_l} \left\| \phi(\hat{x}^s_i) - \phi(\hat{x}^t_j) \right\|_2 = tr(A^T K M_c K A) \quad (8)$$

where the trace of a matrix is represented by $tr(.)$ and $X \in R^{(n_s+n_t)}$ is the cross-domain activity data. The kernel matrix is represented as $K = \phi(\hat{X})\phi(\hat{X})$ and $A \in R^{n\times k}$ is the projection matrix. We adopt the presented technique in [25] and construct the MMD matrix M_c that involves the latent class structure (The matrix, S is constructed from intra- and inter-class separability information) as follows.

$$M_c = \sum_{i,j} s_{ij}\, \mathbf{m}_{ij}\mathbf{m}_i^T \quad (9)$$

where $\mathbf{m}_i = \frac{(s^{ss}_i)^T}{\|\mathbf{s}^{ss}_i\|_1}, \frac{(s^{tt}_j)^T}{\|\mathbf{s}^{tt}_j\|_1}$. The i and jth column vectors of S^{ss} is denoted as s^{ss}_i and s^{tt}_j, respectively. objective is to minimize the class-conditional distribution between the domains and incorporate the inherent structural activity patterns.

Optimization: Our DSAE generated features are non-linear, we employ kernel K instead of X and incorporate regularization term in our optimization function and rewrite the optimization problem as follows.

$$\min_{\mathbf{A^T K H K^T A = I}} \text{tr}\ (\mathbf{A^T K M_c K^T A}) + \beta\ ||\mathbf{A}|| \quad (10)$$

We solve the Equation 10 using Lagrange multiplier technique. We denote the Lagrange multiplier as $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_k) \in R^{k\times k}$ and derive Lagrange function from equation 10 as follows.

$$L = \text{tr}\ (\mathbf{A^T(KM_cK^T} + \beta\mathbf{I)A}) + \text{tr}\ ((\mathbf{I} - \mathbf{A^T KHK^T A})\mathbf{\Lambda}) \quad (11)$$

We take the partial derivative with respect to $\mathbf{A}$ and set the derivative $\frac{\delta \mathbf{L}}{\delta \mathbf{A}} = 0$ then Equation 11 becomes generalized eigen-decomposition and can be written as follows.

$$\mathbf{(KM_cK^T} + \beta\mathbf{I)A = KHK^T A\Lambda} \quad (12)$$

We apply iterative optimization to find k smallest eigen vectors and compute the projection matrix A. The transformed feature space then computed as $Z = AX$. Any classifier can be used to train with the projected cross-domain activity data and infer activities.

Activity Classification:

The proposed method optimizes the intra- and inter-class distance between the source and target domain instances. Furthermore, the generated features space segregate different class instances farther and agglomerate similar class instances. Therefore, the k-nearest neighbor classifier works well to infer activities in the target domain. In this work, we deploy a k-nearest neighbor classifier trained with the cross-domain data to infer target domain activities. To compare the performance of our proposed AR framework, we consider Transfer Component Analysis (TCA) [38], Joint Distribution Adaptation (JDA) [37], and Untran [6]. During the embedding (distribution difference minimization) step, we apply linear kernel for TCA, JDA, our proposed method to construct kernel matrix as suggested by [38] and [37].

IV. Experimental Evaluation

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

A. DataSets Description

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

i) Opportunity dataset (Opp) [39] [40] 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) [41] 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) [42] 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. Implementation Details

We implemented our framework using python based deep learning platform, Tensorflow [43]. Accelerometer sensor data was segmented into 128, 200 and 125 samples with 50% overlap for Opp, Wisdm, and Das. Frames were filtered with a low-pass median filter to remove noises. These frames were then fed into the DSAE to extract features from the sensor signals. We implemented transfer learning baseline methods, TCA, and JDA with python. Our DSAE is comprised of four layers. We concatenated the softmax function at the end of the encoder to build the DSAE classifier and fine-tuned the classifier parameter for the source domain annotated data. We ran our AR framework on a server equipped with four NVIDIA GTX 1080-Ti GPUs and 64 GB memory with an Intel Core i7-6850K processor.

C. Performance Metrics

We evaluated and compared the performance of our AR 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.

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

Datasets	Number of Unseen Activities				
	1	2	3	4	5
Das	86.77	81.70	76.11	72.45	68.02
Opp	82.94	70.98	68.70	63.05	60.67
Wisdm	80.09	68.18	65.25	-	-
Avg.	83.58	73.62	70.02	67.75	64.35

TABLE II: AR framework performance accuracy (%) on a varying number of unseen activities in the target domain while maintaining an equal number of activities in the source and target domains.

D. Experimental Results

Each dataset is partitioned into two groups, and each group contains distinct users. Users are selected randomly to create source and target domain. Table I shows the number of users in the source and target domain for each of the datasets. We evaluated our model performance in presence of varying number of unseen activities and varying number of annotation.

E. Performance under the varying number of unseen activities:

Two experimental settings are considered to showcase the effectiveness of the proposed model, *i)* an equal number of activities in both domains, and *ii)* an increasing number of activities in the target domain.

i) An equal number of activity classes with varying unseen activities: In this experiment, both the source and target domain contain an equal number of activities while varying the number of unseen activities in the target domain. In the case of the Opp and Das dataset, both the source and the target domain contain five activities, and in the Wisdm dataset, both domains contain three activities. In the first step of this experiment, the target domain contains one unseen activity, and the rest of the activities are similar to the source activities. We report the average result by conducting this experiment for all the unseen activities. In the next step, the target domain contains two unseen activities. In this way, we repeat this process until the target domain contains all the unseen activities. The classifier is trained with 20% annotated target domain activities. On average, 111, 17, and 70 annotated activity samples (activity samples distribution shown in table IV) are selected from Das, Opp, and Wisdm, respectively.

Datasets	Number of Unseen Activities				
	1	**2**	**3**	**4**	**5**
Das	93.39	89.89	84.14	85.27	83.59
Opp	72.72	67.69	63.16	60.00	56.87
Wisdm	96.96	89.21	81.98	-	-
Avg.	87.69	82.23	76.43	72.64	70.23

TABLE III: AR framework performance accuracy (%) in the presence of a varying number of unseen activities in the target domain while the target domain contains all the source activities and additional unseen activities

Table II represents the performance of our AR framework. We see that our AR framework achieves accuracy ≈ 87%, 83% and 80% for Das, Opp, and Wisdm dataset, respectively, for single unseen activity in the target domain. The proposed AR framework achieves lower accuracy because the Opp dataset consists of more diverse user activity styles, sensing biases, and device heterogeneity. In addition, this dataset also contains a small number of annotated activity samples (17 samples) and missing values. We notice that our AR framework performance decreases with the increased number of unseen activities in the target domain. This result is expected because we add more unseen activities in the target domain and treat these new activities samples as noise. Our classifier fetches difficulties transferring knowledge from source to target with more new instances.

ii) An increased number of activities in the target with varying unseen activities: In this experimental setting, the target domain contains all the activities present in the source domain. In addition, the target domain has a varying number of unseen activities. The proposed AR framework is trained with 20% annotated data in the target domain; on average, 159, 24, and 131 annotated activity samples (activity samples distribution shown in table V) are selected from Das, Opp, and Wisdm, respectively. Both seen and unseen activities in the source and target domain help establish the minimized cross-domain and in-domain intra- and inter-class distance minimization. Table III represents AR framework performance. For two unseen activities in the target domain, our AR framework achieves 89.89%, 67.69%, 89.21% for Das, Opp, and Wisdm datasets, respectively. Similar source domain activities in the target domain help reduce the cross-domain intra- and inter-class distances between the domains. Our AR framework performance depends on the pseudo labels for the unannotated samples to construct the neighbor graph that explicitly participates in optimization. Therefore, more common activities between the domains help improve the accuracy of our model.

F. Performance under varying labeled activities

We investigate our model performance in the presence of a variable number of annotated data in the target domain for the following scenarios - *i)A* an equal number of activity classes in the target domain, and *ii)* an increased number of activity classes in the target domain.

An equal number of activities in target domain: In this experiment, we vary the number of annotated activity samples in the target domain while maintaining an equal number of activities in both environments. The average activity annotation distributions shown in table IV. In addition, the target domain varies the number of unseen activities. The average performance of our AR framework is shown in figure 4. We observe that our AR framework performance increases with the increasing number of annotated samples in the target domain for all three datasets. Furthermore, we see that the Opp and Das dataset performs better than the Wisdm dataset because the Wisdm dataset contains fewer activities in both domains. Our AR framework creates a neighbor graph composed of intra- and inter-class distances within- and cross-domains. We infer that more activities help create this neighbor graph, minimizing discrepancies and maximizing knowledge transfer. Therefore, the less common activities in both domains degrade our model performance.

Datasets	Percentage of annotated activity samples								
	10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
Das	55	111	167	223	278	334	390	445	500
Opp	8	18	26	35	45	53	62	71	80
Wisdm	35	70	105	140	176	211	246	281	316

TABLE IV: The number of activity samples in percentage distribution for an equal number of activities in target domain.

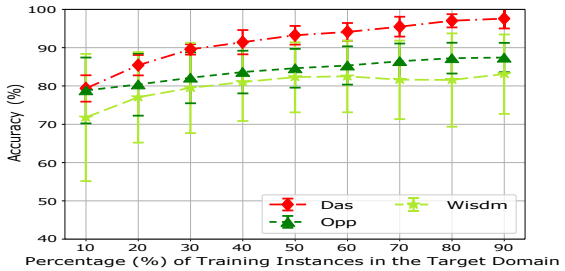


Fig. 4: Our AR framework Performance (Accuracy(%)) on varying labeled data in the target domain. Note that both domain contains same number of activities

Datasets	Percentage of annotated activity samples								
	10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
Das	80	160	239	320	400	480	560	639	720
Opp	12	24	37	49	62	74	86	99	111
Wisdm	65	131	196	262	328	394	459	524	590

TABLE V: The number of activity samples in percentage distribution for an increased number of activities in the target domain.

An increased number of activities in the target domain: In this experimental setting, we vary the number of labeled activities from 10% to 100% while maintaining similar activities and additional unseen activities in the target. The average activity annotation distributions shown in table V. Furthermore, we increase unseen activities by adding one other unseen activity in the target in each step. The average results are reported for each corresponding percentage of annotated data in Figure 5. With the varying labels, the performance of our classifier follows similar trends as with the equal number of classes in the target domain. However, the AR framework fetches difficulties constructing neighbor graphs for the Opp dataset as the total number of activity samples is smaller. In addition, it contains missing values and idle body part movement.

G. Individual Activity Recognition Performance

We investigate the individual activity recognition performance of our AR framework. In this experiment, we varied the number of unseen activities from 1 to n in the target domain for each activity while using 20% annotated target

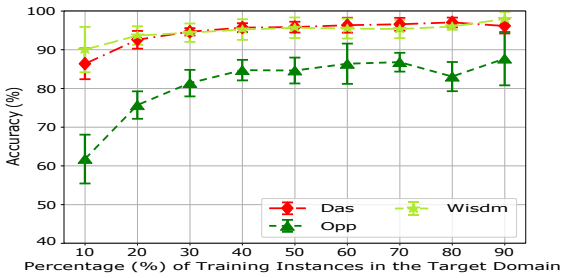


Fig. 5: Our AR framework Performance (Accuracy(%)) on varying labeled data in the target domain. Note that both domain contains increased number of unseen activities

Dataset	Our AR	UnTran	JDA	TCA
Das	**81.70**	74.75	51.23	45.24
Opp	63.15	**65.86**	50.92	26.77
Wisdm	**80.68**	69.51	42.61	34.68
Avg.	**75.18**	70.07	48.25	35.56

TABLE VI: Baseline performance comparison (Accuracy (%)) for an equal number of activities in the target domain

domain samples. The average individual activity recognition performance for each dataset is shown in Figure 6a, 6b, and 6c, respectively. From figure 6a, we see that our model achieves F1 scores ≈65%, 60%, 62% for activities 'Running', 'Lying', and 'Exercising', respectively for Opp dataset. In the case of the 'Running' activity, a few instances are detected as 'Lying' or 'Exercising.' Similarly, 'Lying' and 'Exercising' activities are also falsely detected as 'Exercising' or 'Running' and vice-versa. This misclassification happens as a result of similar features present in those activities. Therefore, we can infer that activities with distinct patterns improve our model's performance. In the case of the Opp dataset, we notice that 'Clean Table,' 'Open Fridge' shows F1 scores ≈ 43% and 48%, respectively. Opp dataset has missing values and also contains non-movement-related activities. These activities share less knowledge during neighbor graph generation, and hence performance degrades our AR framework. In the case of the Wisdm dataset, we see that our model achieves F1, precision, and recall values of ≈ 86%, 83%, 89%, respectively. This dataset contains no missing values and less heterogeneous instances in the source and target domain that help improve the performance of our model.

H. Baseline Performance Comparison

To show the efficacy and effectiveness of our AR framework, we compare the performance of our model with the following baseline methods *UnTran*, *JDA* and *TCA*. We follow similar experimental settings as *UnTran* to compare the performance with the state-of-the-art algorithm. The experiment was conducted with two unseen activities in the target domain for both cases - *(*i)) an equal number of activities and *ii)* an increased number of unseen activities in the target domain. These two unseen activities are randomly selected from the target domains. We repeated these random unseen activity selections for all the unseen activities and reported our AR framework's average performance in tables VI and VII.

Dataset	Our AR	UnTran	JDA	TCA
Das	**89.89**	78.92	69.51	68.25
Opp	67.69	**74.57**	69.96	36.26
Wisdm	**89.21**	81.32	66.67	67.03
Avg.	**82.63**	78.27	68.71	57.18

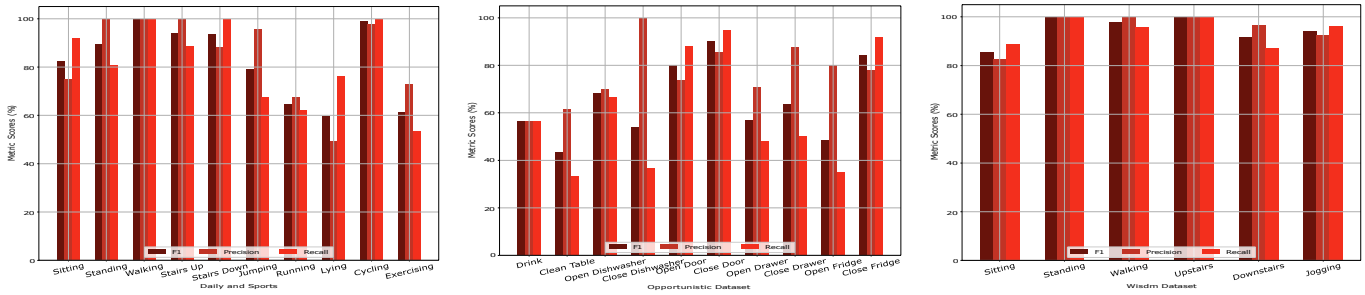
TABLE VII: Baseline performance (Accuracy (%)) comparison for an increased number of unseen activities in the target domain

With both an equal number of activities and an increased number of activities scenarios, the proposed AR is trained with 20% labeled data in the target domain. In contrast, other baseline methods are trained with 30% annotated target domain activity samples. We choose 30% annotated target domain data because the state-of-the-art *UnTran* model works well with 30% labeled data. Table VI represents the performance comparison of our AR framework for balanced activities between the source and target domain in the presence of two unseen activities in the target domain. From Table VI we see that our AR framework achieves superior performance compared to other baselines. Our AR framework achieves an average performance accuracy of 75%. Compared to *UnTran* our model performance gain is ≈ 5-6% with only 20% labeled activity samples in the target domain. Our AR framework achieves performance gain ≈ 26-39% compare to other methods (JDA, TCA). These results clearly show the effectiveness of explicit structural mapping between the source and target domain. It is noted that our proposed AR framework requires 10% lesser annotated activity samples to achieve performance gain 5-6% accuracy compared to the state-of-the-art *Untran* method. In other words, the proposed AR framework requires 56, 8, and 35 fewer activity samples for Das, Opp, and Wisdm datasets to achieve the performance gain of 5-6% accuracy.

Table VII shows the detailed comparison results of our AR framework with the existing methods for an increased number of unseen activities in the target domain. Our AR framework achieves performance gain ≈ 4-5% compared to *UnTran* model with 10% less annotation. In other words, the proposed AR framework requires 79, 13, and 65 fewer activity samples for Das, Opp, and Wisdm datasets to achieve a performance gain of 4-5% accuracy. We also notice that *UnTran* model performs better for the Opp dataset. We suspect that decision fusion from multiple classifiers helps better performance gain for more diverse, heterogeneous, and noisy environments. This experimental setup helps understand additional activities in the target domain capable of learning from the existing activities in the source domain.

I. Parameter Sensitivity

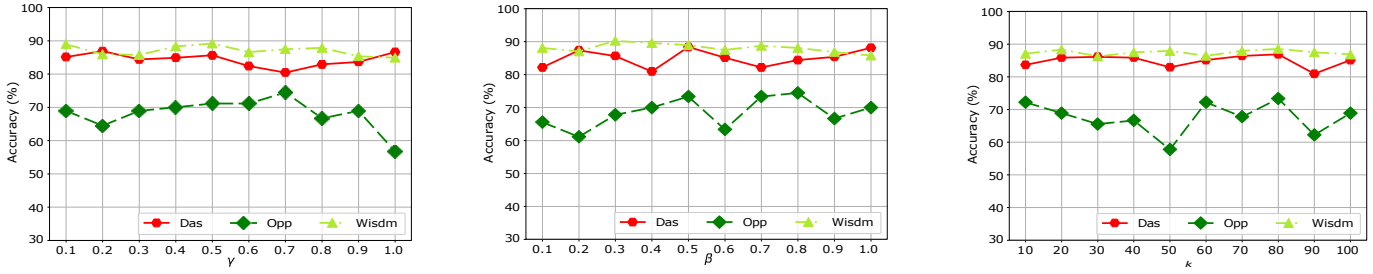
In this section, we investigate our AR framework performance on the following model parameters setting- *i)* similarity balancing parameter (γ), *ii)* regularizing parameter (β), and *iii)* number of kernel subspace (k). We change one parameter and keep the other two parameters unchanged during the experiment. We sample the value of the γ and β from $\{0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$. We also sample the parameter $k \in [10, 100]$ and report the accuracy of



(a) Our AR framework Performance (Accuracy(%)) for Das Dataset

(b) Our AR framework Performance (Accuracy(%)) for Opp Dataset

(c) Our AR framework Performance (Accuracy(%)) for Wisdm Dataset



(a) The parameter influence of γ

(b) The parameter influence of β

(c) The parameter influence of k

Fig. 7: Influence of learning parameter on our AR framework

our AR framework. Note that we train our classifier in the presence of 20% labeled data in the target domain. Both the source and target domains contain eight activities in common. In addition, our target domain contains two unseen activities. Figure 7 showcase the parameter impact of our AR framework performance for all three datasets. Our model approximately performs best at $\gamma = 0.5$ therefore, we choose $\gamma = 0.5$ for all three datasets throughout our experiments. Note that (Fig. 7a) our model achieves accuracy $\approx$ 85%, 89% and 71% for $\gamma = 0.5$ for Das, Wisdm and Opp dataset, respectively. We also notice that the performance of our model for the parameter $\beta = 0.5$ is $\approx$ 88%, 89%, and 74% for Das, Wisdm, and Opp datasets, respectively. Therefore, we choose $\gamma = 0.5$ throughout our experiments. From Figure 7c, we see that our model performs better for $k = 80$. Theoretically, $k = 0$ means an ill-defined problem, and $k = \infty$ means cross-domain adaptation has not been performed.

V. Discussion

Our proposed activity recognition framework addresses a significant promising problem of unseen activities with the help of limited annotation in the target environment. There are, however, a few limitations of our AR framework. First, we evaluated our framework with wearable accelerometer sensors signal only. Though performance examination against three public datasets implicitly attests to the efficacy of our framework against users, environmental heterogeneity, and sensing biasness. However, additional investigations are required to understand the effect of heterogeneous sensors (i.e., camera, PIR, etc.) and devices (smartwatch, smart-necklace, etc.). Furthermore, our AR helps reduce the amount of annotation required in the target domain but could not diminish the required annotation efforts. In this work, we focus on average performance for recognizing unseen activities. However, additional investigation is needed to understand the impact of the presence and absence of specific activities in the target domain.

VI. Conclusion

Human activity recognition techniques help develop various smart applications in different domains such as healthcare, obesity management, sports analytic, etc. Our transfer learning-enabled activity recognition technique helps infer unseen activities with limited annotated activity samples in the target environment. This paper proposes a novel activity recognition framework to minimize within- and cross-domain intra-class distance, maximize inter-class distance, and recognize seen and unseen activities in the target domain. First, we exploit the deep sparse autoencoder to generate source and target domains features. We then utilized the MMD distance minimization techniques to reduce the discrepancy between the domains and recognize activities in the target domain. Finally, we evaluated our proposed framework performance with several state-of-the-art transfer learning baseline models. The recognition performance of our framework suggests that our AR framework is scalable and adaptable in large-scale, diverse target environments.

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] 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.

[6] Md Abdullah Al Hafiz Khan and Nirmalya Roy. Untran: Recognizing unseen activities with unlabeled data using transfer learning. In *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 37–47. IEEE, 2018.

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

[13] Thomas Plo¨tz, 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.

[14] 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.

[15] 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.

[16] 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.

[17] Yu Guan and Thomas Plo¨tz. Ensembles of deep lstm learners for activity recognition using wearables. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(2):11, 2017.

[18] Ming Zeng, Tong Yu, Xiao Wang, Le T Nguyen, Ole J Mengshoel, and Ian Lane. Semi-supervised convolutional neural networks for human activity recognition. In *Big Data (Big Data), 2017 IEEE International Conference on*, pages 522–529. IEEE, 2017.

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

[20] 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.

[21] 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.

[22] Alberto Calatroni, Daniel Roggen, and Gerhard Tro¨ster. 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.

[23] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. Deep transfer metric learning. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 325–333. IEEE, 2015.

[24] Lihang Liu, Weiyao Lin, Lisheng Wu, Yong Yu, and Michael Ying Yang. Unsupervised deep domain adaptation for pedestrian detection. In *European Conference on Computer Vision*, pages 676–691. Springer, 2016.

[25] Tzu Ming Harry Hsu, Wei Yu Chen, Cheng-An Hou, Yao-Hung Hubert Tsai, Yi-Ren Yeh, and Yu-Chiang Frank Wang. Unsupervised domain adaptation with imbalanced cross-domain data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4121–4129, 2015.

[26] Gabriela Csurka. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*, 2017.

[27] Yu Zhang and Dit-Yan Yeung. Transfer metric learning by learning task relationships. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1199–1208. ACM, 2010.

[28] Jindong Wang, Yiqiang Chen, Lisha Hu, Xiaohui Peng, and Philip S Yu. Stratified transfer learning for cross-domain activity recognition. *arXiv preprint arXiv:1801.00820*, 2017.

[29] 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.

[30] 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.

[31] 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.

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

[33] 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.

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

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

[36] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012.

[37] 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.

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

[40] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Fo¨rster, Gerhard Tro¨ster, 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.

[41] 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.

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

[43] Mart´ın Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.