

Cross-Domain Similarity in Domain Adaptation for Human Activity Recognition

Samra Kasim
 Whiting School of Engineering
 Johns Hopkins University
 Baltimore, USA
 skasim3@jhu.edu

John W. Sheppard
 Gianforte School of Computing
 Montana State University
 Bozeman, USA
 john.sheppard@montana.edu

Abstract—Human Activity Recognition (HAR) is a difficult machine learning problem, even for state-of-the-art deep learning models, due to HAR data’s within-domain and cross-domain heterogeneity. Our research addresses the challenge of closed-set domain adaptation in heterogeneous, parameter-based, and transductive transfer learning on HAR datasets. We use a Bidirectional Long Short Term Memory (BLSTM)-based model that, in addition to training for classification accuracy using only labeled data from the source domain, also jointly trains on source and unlabeled target datasets to reduce the discrepancy between source and target domains using cross-domain similarity as an additional loss function. Our work contributes to existing research in the area of domain adaptation for HAR by evaluating the performance of the following cross-domain similarity metrics as loss functions in improving model classification accuracy: 1) Maximum Mean Discrepancy (MMD), which uses feature means to measure similarity between two domains; 2) Kernel Canonical Correlation Analysis (KCCA), which utilizes canonical correlations for similarity determination; and 3) Cosine Similarity, a metric that uses the cosine of the angle between two vectors as similarity measure. Our results demonstrate that MMD as a cross-domain similarity metric not only outperforms KCCA and Cosine Similarity in domain adaption, but also results in mean F1 score improvement of 45% over results where a model is trained solely on the target dataset.

I. INTRODUCTION

Transfer learning is a class of machine learning that aims to transfer knowledge via a learner from one domain (source) to a different, but related, domain (target). Transfer learning can be subdivided into homogeneous or heterogeneous. The former is for situations where the source and target domains share the same feature space while the latter is a type of transfer learning that is used when the feature spaces differ between the domains. Transfer learning can also be grouped by the type of learning: 1) transductive, where only the labeled source data and unlabeled target data are available; 2) inductive, where only the labeled source data are available; and 3) unsupervised, where neither source nor target data are labeled [1]. In addition, transfer learning can be grouped by four different approaches: 1) instance-based approaches that utilize instance weighting strategies; 2) parameter-based approaches that transfer knowledge at the model or parameter level; 3) relational-based approaches that focus on the relational domain; and 4) feature-based approaches that transform or augment features [2].

Domain adaptation is a type of transfer learning that minimizes the gap between source and target domains and can be divided as follows: 1) closed-set, i.e., the label set in the source and target dataset is the same and there are no unknown classes between the domains; 2) partial domain adaptation, i.e., the source dataset is much larger than the target dataset, so it is assumed that the source label set will contain the target label set; 3) open-set, i.e., the source and target dataset share some common labels and differ in some unknown class labels; and 4) universal domain adaptation, which involves a labeled source dataset and a target dataset for which little is known [2, 3].

The focus of our research is heterogeneous, parameter-based transductive transfer learning with closed-set domain adaptation. More formally, a domain \mathcal{D} consists of three parts: an input feature space \mathcal{X} , an output feature class space (i.e., the classification labels) \mathcal{Y} , and a learned prediction function f , where $f := P(\mathbf{y}|\mathbf{x})$ [4]. Therefore a domain is defined as $D := \{\mathcal{X}, \mathcal{Y}, P(\mathbf{y}|\mathbf{x})\}$. Given source domain (\mathcal{D}^s) and target (\mathcal{D}^t) and their respective feature spaces \mathcal{X}^s and \mathcal{X}^t , in heterogeneous, transductive transfer learning $\mathcal{D}^s \neq \mathcal{D}^t$, it follows that $\mathcal{X}^s \neq \mathcal{X}^t$, and that the probability distributions are also $P(\mathbf{X}^s) \neq P(\mathbf{X}^t)$, where \mathbf{X}^s and \mathbf{X}^t are instance sets of their respective feature spaces. In addition, for heterogeneous transfer learning, the label sets may not be equal, i.e., ($y^s \neq y^t$). However, for our research, we assume the label sets are equal ($y^s = y^t$) as we are conducting closed-set domain adaptation. Last, task \mathcal{T} is defined as $\mathcal{T}^s = \{y^s, f^s\}$ and $\mathcal{T}^t = \{y^t, f^t\}$ for source and target, respectively, and in our research the task, human activity recognition (HAR), remains the same (i.e., $\mathcal{T}^s = \mathcal{T}^t$).

Activity prediction tasks, such as HAR, are classification tasks that use probabilistic models to categorize human activities (e.g., running, cooking, sitting, etc.) based on data generated by sensors (e.g., gyroscope, accelerometer). By their nature, these tasks have a temporal component as the datapoints that make up the performance of a particular task (e.g., the change in input to an accelerometer during a one minute run) are collected successively over a time interval. Bidirectional Long Short Term Memory (BLSTM) models for HAR [5]–[8] have demonstrated effectiveness in identifying patterns contained within time-series data.

Even so, HAR is a difficult problem to solve for even state-of-the-art deep learning models [9]. The domain variability across datasets and even within datasets can be extensive. This variability can result from wearing diversity, e.g., the placement of the wearable sensor on the body [9] or from cross-user diversity; how users perform the activities can differ significantly. Known as the population diversity problem [10], this variability in a dataset increases as the number of users in the dataset increases and causes classification accuracy to degrade quickly and significantly [11]. Another source of variability is device-instance diversity, e.g., the differences among the sensors from one phone model to another or among different brands of sensors. These sensor heterogeneities demonstrably impair HAR classification. Stisen et al. [12] showed that, for some devices, sensor biases accounted for an 8% deviation in exerted force – a bias that, they claim, was large enough to “account for the acceleration of a fast train”.

Because our research is focused on domain adaptation and evaluating the effectiveness of cross-domain similarity metrics in minimizing the gap between source and target domains, we do not mitigate for heterogeneities within a dataset. Nor do we mitigate for cross-dataset heterogeneities that may result from use of different smartphone models in one dataset or the use of different smartphone models or sensors across datasets. Our focus is training models from labeled (source) and unlabeled (target) HAR domains jointly using a cross-domain similarity metric that minimizes the domain discrepancy between source and target domains. This results in model parameters that are optimized for classification accuracy (based on the source data) and also shared source and target domain features. Then this jointly trained model is used to classify human activities in the target domain. Unlike other research in HAR domain adaptation, which evaluated cross-domain adaptation at the user level [13, 14] or by comparing only two datasets for particular activities [15], our research focused on evaluating the effectiveness of cross-domain similarity metrics across multiple domains.

We propose a BLSTM-based model for joint training with which we conduct parameter-based transfer learning by adjusting model parameters based on optimizing a loss function that jointly trains to improve HAR classification accuracy on the source domain and minimize discrepancy between source and target domains. We test three cross-domain similarity metrics as loss functions for their effectiveness in domain adaptation: 1) Maximum Mean Discrepancy (MMD); 2) Kernel Canonical Correlation Analysis (KCCA); and 3) Cosine Similarity. For completeness, we also train and test the model on the target domain to provide an upper-bound target domain classification accuracy measure (the ground truth or GT) and train the model on the source domain and test on the target domain without performing explicit knowledge transfer or domain gap minimization to provide a lower-bound target domain classification accuracy measure (naïve transfer or NT). We hypothesize that 1) similarity-based loss metrics will outperform NT due to the additional information learned by the models via knowledge transfer during joint training, but will underperform

GT models; and 2) incrementally increasing the size of the source dataset will improve model performance on the target dataset with joint training.

The rest of this paper is organized as follows. In Section II, we present related work. In Section III, we describe the open-source datasets used in our experiments. In Section IV, we describe our implemented BLSTM-based model and detail the loss functions used for training optimization. In Section V, we discuss our experiment design. In Section VI, we outline and analyze our results. In Section VII, we detail our conclusions and consider future work.

II. RELATED WORK

In a 2019 survey, Wilson and Cook [16] identified activity prediction using time-series data as an area of research due to the challenges posed by large differences in feature space and/or class labels between source and target domains. Recently, there has been additional research specifically addressing domain adaptation in activity prediction. In 2018, Ding et al. [13], after comparing Domain Adversarial Neural Network, Wasserstein Distance, and MMD as loss functions, found the MMD loss method to be well-suited for HAR. They combined MMD with Center Loss, which they used to decrease the inter-class differences in their datasets. The authors randomly selected four users from UCI-HAR dataset and trained on these users with the purpose of transferring between the users. That is, they trained on one user (source) and transferred and tested on another user (target). They found that MMD outperformed the two other widely-used algorithms in their experiments. Our paper builds on their research by comparing MMD to KCCA and Cosine Similarity and evaluating transfer learning at the level of the domain and not the individual.

In a 2019 paper, Hosseini et al. [15] utilized wireless health sensor data and a deep sequential domain adaptation model for health signal classification. Specifically, they employed a BLSTM model to extract feature representations from sensor input signals and jointly trained the BLSTM prediction model on the source and target domains while minimizing MMD loss to successfully show the transfer of HAR from adults to children in five activities. Our paper adds to this research by evaluating transfer learning at the domain level without regard to the specific qualities of the source and target domains and by comparing the effectiveness of three distinct cross-domain similarity metrics.

The Cosine Similarity metric has been used widely in Information Retrieval, Natural Language Processing, and Computer Vision tasks to determine similarity between texts and images, but its use has been somewhat limited in HAR. However, in 2019, Feng and Duarte [17] focused on performing few-shot transfer learning for HAR utilizing a stacked Long Short Term Memory (LSTM)-based model for feature extraction and classification with parameter-based transfer-learning for knowledge transfer. One of their experiments used the exponential value of Cosine Similarity to measure cross-domain sample relevance between the source and target datasets. Specifically, they trained their model on the source domain,

TABLE I
SUMMARY OF DATASETS

Dataset	Accelerometer Unit Measure	Gyroscope Unit Measure	Sampling Frequency	Sample Length	Device
PAMAP2	m/s^2	rad/s	100Hz	0.01s	sensor
UCI-HAR	g	rad/s	50Hz	2.56s	smartphone
KU-HAR	m/s^2	rad/s	100Hz	3s	smartphone

then calculated the cross-domain measure between the source and target datasets, and used this measure (in addition to other measures such as parameters from the classifier) to update the target model’s parameters. Our research also uses a parameter-based transfer-learning approach, but compares the effectiveness of the Cosine Similarity cross-domain similarity metric to MMD and KCCA.

In 2022, Hamidi et al. [14] utilized Deep Canonical Correlation Analysis (DCCA), which is an alternative to KCCA. Like KCCA, DCCA also learns the representations of the source and target views, but DCCA does not use the inner product in its calculations. For their research, they utilized phone sensor data collected from three individuals who placed a smartphone on their hip, torso, bag, and hand. Then they sought to accurately classify the human activity independent of the user or the placement of the phone on the body. They used DCCA to learn a joint representation of the data and concluded that sensors placed on different parts of the body provided a more comprehensive view of how different signals contribute to activity recognition. They also found that learning a joint representation of the source and target domains via DCCA made their models robust to sensor failure and data loss. Their state-of-the-art research built a model to evaluate HAR in three individuals from a single dataset while our research seeks to evaluate the impact of three cross-domain similarity metrics in improving HAR across three datasets comprising 129 individuals.

III. HUMAN ACTIVITY RECOGNITION DATASETS

Each of the datasets (Table I) chosen for our study is a distinct domain and our experiments evaluate domain adaptation between the domains. UCI-HAR and KU-HAR data were generated from smartphones, while PAMAP2 utilized sensors placed on different parts of the body. The datasets are from different sources, but all use an accelerometer and a gyroscope for measurements. As our experiments are concerned with closed-set domain adaptation, the experiments will consider four activities (sit, stand, stairs up, stairs down), which are shared by all of the datasets.

For PAMAP2 and KU-HAR, the features are normalized to $[-1, 1]$. The UCI-HAR dataset was already normalized to $[-1, 1]$ by its creators. We further modified the original datasets by: 1) imputing missing values in the KU-HAR dataset via K-Nearest Neighbor imputation where $k=3$; 2) descaling the KU-HAR and PAMAP2 datasets to a sampling frequency of 50Hz to match the sampling frequency of the

UCI-HAR dataset; 3) converting the accelerometer unit measure, meters per second squared (m/s^2), in KU-HAR and PAMAP2 datasets to gravitational acceleration (g) to match the accelerometer unit measure in the UCI-HAR dataset; and 4) mapping the class labels in each dataset to the same integer value for the same label, i.e., sit is zero, stand is one, stairs up and down are two and three respectively across the datasets.

The KU-HAR raw dataset is the largest dataset by the number of rows (729,956), but the smallest in terms of number of features (6). The UCI-HAR raw dataset is the smallest in terms of rows (5,331), but the largest in terms of number of features (561). Last, the PAMAP2 raw dataset has 261,698 rows and 39 features. Specifically:

- The PAMAP2 dataset contains 18 different physical activities performed by nine subjects wearing three sensors worn over the wrist, on the chest, and on the dominant hand and a heart rate monitor. The sensory data include temperature, 3D-acceleration data (ms^2), 3D-gyroscope data (rad/s), and 3D-magnetometer data (μT). The sampling frequency is 100Hz. [18].
- The UCI-HAR dataset comprises six activities generated from 30 volunteers wearing a Samsung Galaxy S II smartphone on the waist. Using the smartphone’s accelerometer and gyroscope, data were captured at a rate of 50Hz and are composed of three-axial acceleration and three-axial gyroscope measurements. The sensors’ signals are sampled into a fixed width window of 2.56 seconds [19].
- The KU-HAR dataset is based on 18 activities collected from 90 individuals using a smartphone. Five android smartphone models worn on the waist in a fanny pack were used for data collection: Samsung Galaxy J7 (2017), XiaomiRedmi Note 4, Realme 3 Pro, Realme 5i, and Realme C3 [20]. The time is in ms, three axis-acceleration is in m/s^2 , and three-axis rotation via the gyroscope is in rad/s . The sampling frequency is 100Hz [20].

IV. JOINT LOSS MODEL

An LSTM model is a type of Recurrent Neural Network (RNN) [21] that is used to process sequential data. At its most basic, an LSTM model consists of chained LSTM cells, which pass states from one cell to the next. Each cell takes the cell state from the previous cell as input, modifies it, and outputs it to the next cell. The modifications within each cell are controlled by gates composed of two sigmoid layers and one hyperbolic tangent layer. The gates decide what information from the previous cell to forget, update, and add via pointwise

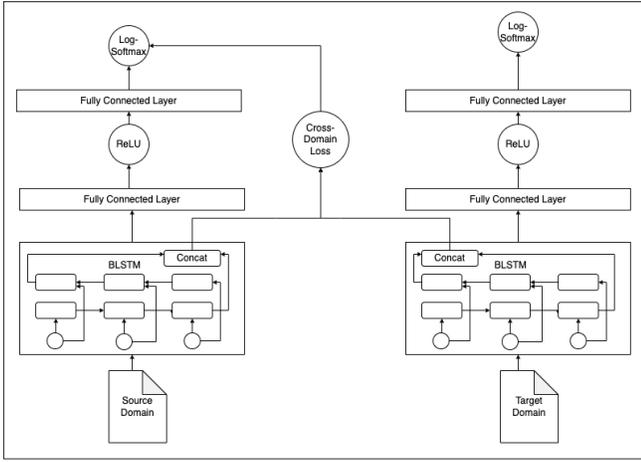


Fig. 1. BLSTM-Based Architecture for Joint Loss Training

operations. This modified cell is then passed on as input to the next cell. LSTMs, however, do not consider future information in the output [22, 23]. As a result, BLSTMs, which consider hidden states in future and past directions, have been applied to time-series data. BLSTM is a type of RNN that conducts learning in forward and backward directions by adding an additional LSTM backward layer on top of an LSTM forward layer to simultaneously learn all inputs in a sequence [22].

Based on the work of Hosseini et al. [15] and Tzeng et al. [24], we propose a BLSTM-based model (Fig. 1) that jointly trains on the source and target datasets by including a similarity-based loss function in the training. The goal of the model is to identify a function (ϕ) that learns the difference (or similarity) between the feature spaces of the source and target datasets and a prediction function (f) that classifies an instance in the target dataset as a particular activity. The data vectors \mathbf{x}_m^s and \mathbf{x}_n^t are instances in the feature space ($\mathbf{x}^s \sim \mathcal{X}^s$ and $\mathbf{x}^t \sim \mathcal{X}^t$, respectively). Taking a labeled source dataset \mathcal{D}^s with labels (y^s) and unlabeled target dataset \mathcal{D}^t , each dataset is split into windows l where $l_m^s = \{\mathbf{x}_1^s, \mathbf{x}_2^s, \dots, \mathbf{x}_{m-1}^s\}$ and $l_n^t = \{\mathbf{x}_1^t, \mathbf{x}_2^t, \dots, \mathbf{x}_{n-1}^t\}$ for source and target datasets respectively. The assumption is that the probability distributions of the source and target datasets differ and by identifying the function ϕ , we can minimize the distance between the domains so that $P(\phi(\mathbf{x}^s)) \approx P(\phi(\mathbf{x}^t))$. Using a network architecture that consists of two layers of LSTM cells (one forward and one backward), each window is fed simultaneously to a forward LSTM cell and a backward LSTM cell. The outputs of the individual LSTM cells from each dataset are concatenated. Then these concatenated outputs from the source and target LSTM cells (\mathbf{X}^s and \mathbf{X}^t , respectively) are utilized to derive the cross-domain similarity metric between the source and target datasets. This metric is key to performing parameter-based transfer learning. It represents the gap (or loss) between the source and target domains, which is minimized during gradient descent to update model parameters. It is calculated using the following functions.

A. Maximum Mean Discrepancy

MMD is the distance in the feature means between the source and target distributions, which can be used to determine the similarity of two datasets. For example, given a source domain \mathcal{D}^s and a target domain \mathcal{D}^t , it can be stated that $\mathbf{X}^s \in \mathcal{D}^s$ and $\mathbf{X}^t \in \mathcal{D}^t$ (where \mathbf{X}^s and \mathbf{X}^t are concatenated outputs of the source and target LSTM cells, respectively) and that these domains have a shared feature space \mathcal{X} .

Then, a kernel trick (ϕ) can be employed to transform the shared features space \mathcal{X} to the Reproducing Kernel Hilbert Space (\mathcal{H}), i.e., $\phi : \mathcal{X} \rightarrow \mathcal{H}$. In classical kernel methods, embedding points are used as inputs to the Reproducing Kernel Hilbert Space (\mathcal{H}). But in MMD, a generalization of these classical kernel methods is employed, which uses the feature means as input elements in \mathcal{H} . Our implementation utilizes a Gaussian Radial Basis Function (RBF) transformation, which maps the datapoints from \mathcal{X} to \mathcal{H} by calculating the dot products of the concatenated output from the LSTM cells (\mathbf{X}^s and \mathbf{X}^t) and results in MMD being zero only if the source and target distributions are identical. Finally, the L_2 distance between the feature means of the distributions is calculated.

$$k(\mathbf{X}^s, \mathbf{X}^t) = \exp\left(-\frac{\|\mathbf{X}^s - \mathbf{X}^t\|^2}{2\sigma^2}\right)$$

$$\begin{aligned} \text{MMD}^2(\mathcal{D}^s, \mathcal{D}^t) &= \|\mathbb{E}_{\mathbf{X}^s, \mathbf{X}^{s'} \sim \mathcal{D}^s} k(\mathbf{X}^s, \mathbf{X}^{s'}) \\ &+ \mathbb{E}_{\mathbf{X}^t, \mathbf{X}^{t'} \sim \mathcal{D}^t} k(\mathbf{X}^t, \mathbf{X}^{t'}) \\ &- 2\mathbb{E}_{\mathbf{X}^s \sim \mathcal{D}^s, \mathbf{X}^{t'} \sim \mathcal{D}^t} k(\mathbf{X}^s, \mathbf{X}^{t'})\|_{\mathcal{H}}^2 \end{aligned}$$

B. Kernel Canonical Correlation Analysis

In Canonical Correlation Analysis (CCA) transformation, the input source and target features spaces are transformed to some common feature space $\phi : \mathcal{X}^s \rightarrow \mathcal{X}^c$ and $\phi : \mathcal{X}^t \rightarrow \mathcal{X}^c$ and the similarity metric measures the canonical correlations, which are the cosines of principle angles and smaller angular planes most shared between the source and target datasets in each canonical direction [25]. Therefore, two datasets that have higher canonical correlations are deemed to be more similar.

KCCA is a variation of CCA and utilizes a kernel trick to derive nonlinear correlations and avoids overfitting on the training data [26]. In KCCA, given two matrices (\mathbf{X}^s and \mathbf{X}^t), we can learn the projection vectors \mathbf{u}^s and \mathbf{u}^t using weighting vectors ($\alpha^s \in \mathbb{R}^n$ and $\alpha^t \in \mathbb{R}^n$, respectively) that maximize the correlation coefficient ρ .

$$\begin{aligned} \mathbf{u}^s &= \mathbf{X}^s \alpha^s \\ \mathbf{u}^t &= \mathbf{X}^t \alpha^t \end{aligned}$$

Employing the aforementioned Gaussian RBF kernel trick, we can derive kernel matrices $\mathbf{K}^s = k(\mathbf{X}^s, \mathbf{X}^s)$ and $\mathbf{K}^t = k(\mathbf{X}^t, \mathbf{X}^t)$, which maximizes the correlation coefficient.

$$\max_{\alpha^s, \alpha^t} \rho = \frac{\alpha^{s\top} \mathbf{K}^s \mathbf{K}^t \alpha^t}{\sqrt{\alpha^{s\top} \mathbf{K}^s \alpha^s} \sqrt{\alpha^{t\top} \mathbf{K}^t \alpha^t}}$$

However, if \mathbf{K}^s and \mathbf{K}^t are invertible, the weighting vectors can be arbitrary but will result in a trivial solution. Therefore a regularization term (λ) is added and requires solving:

$$\mathbf{K}^t(\mathbf{K}^t + \kappa\mathbf{I})^{-1}\mathbf{K}^s\alpha^s = \lambda^2(\mathbf{K}^s + \kappa\mathbf{I})\alpha^s$$

where α^t can be derived from:

$$\alpha^t = \frac{(\mathbf{K}^t + \kappa\mathbf{I})^{-1}\mathbf{K}^s\alpha^s}{\lambda}$$

KCCA has demonstrated higher accuracy in classification when compared to standard measures such as Kullback Leibler-Divergence because the latter is highly sensitive to small changes in the data that are irrelevant to classification [25]. In our implementation, we utilize Tensor KCCA with RBF transformation developed by Kim et al. specifically for tensors [25, 27]. They utilized TKCAA for extracting similarity features from HAR video datasets, but they note that their framework can be used for other tasks requiring measurements between tensor data.

C. Cosine Similarity

For Cosine Similarity, the metric returned is the cosine of the angle between instances in the source and target datasets and is valued between $[-1, 1]$, where $\cos(\theta) = 0$ indicates two orthogonal vectors, $\cos(\theta) = 1$ indicates two proportional vectors, and $\cos(\theta) = -1$ indicates two opposite vectors. In our implementation, Cosine Similarity is measured between HAR instances.

$$\cos(\theta) = \frac{\mathbf{X}^s \cdot \mathbf{X}^t}{\|\mathbf{X}^s\| \|\mathbf{X}^t\|}$$

The use of the aforementioned functions to calculate similarity metrics in our BLSTM architecture aims to reduce the discrepancy between the source and target domains by jointly training the source and target data and therefore reducing the domain differences. The concatenated last hidden layer output from the BLSTM layers is utilized to calculate the cross-domain loss. In addition, during training, the last layer output resulting from the source data is fed to a fully connected layer followed by a Rectified Linear Unit (ReLU) activation function. The output of this activation function is fed to another fully connected layer, which is followed by a log Softmax activation function. The output of the log Softmax activation (L_{class}) function is combined with the cross-domain loss ($L_{\text{cross-domain}}$) derived jointly from the source and target data to calculate joint loss (L_{joint}).

We utilize an ω value to proportionally calculate cross-domain and softmax loss in our final loss measure, which enables us to emphasize one loss value over another as necessary. We implement the log of a Softmax function as our activation function in the last layer of our BLSTM-based architecture. This combined loss output is then minimized in training by a Stochastic Gradient Descent (SGD) algorithm with momentum.

During training, the parameters of the model are updated via SGD to optimize the joint loss function. Therefore, these

parameters account for both the learned classification accuracy from being trained on the source domain and also the reduction in domain discrepancy via the cross-domain similarity metric that attempts to minimize the difference between the source and target domains. The model with these jointly trained parameters is then used for classification of the target dataset during testing where the output of the target BLSTM layer is passed to the fully connected layer, followed by the ReLU activation function, then the second fully connected layer, and finally through to the log softmax function.

$$L_{\text{joint}} = \omega L_{\text{class}}(\mathbf{X}_s, \mathbf{y}_s) + (1 - \omega)L_{\text{cross-domain}}$$

V. EXPERIMENT DESIGN

A. Pre-Experiment Processing

Prior to experimentation, unit conversion was performed to normalize units for acceleration and rotation across the datasets. In addition, KU-HAR and PAMAP2 were downsampled to 50Hz. The raw datasets were then split into individual files for each activity and individual. This enabled randomized data shuffling while keeping the temporal component of the data (i.e., the order of the rows for each activity for each individual) intact. Only the four activities being experimented on (outlined in Section III) were retained for experiments. The files were split into validation (10% for KU-HAR and PAMAP2 and 20% for UCI-HAR) and training datasets. UCI-HAR raw dataset was already split into test and train by its creators with min-max scaling applied, so this split was kept intact in our experiments. In addition, UCI-HAR is the smallest dataset, so a larger proportion of the data were used for validation as compared to the other two datasets.

B. Hyperparameter Tuning

Hyperparameter tuning was performed using grid search. The following values were used in the final experiments. For UCI-HAR as target dataset, 250 epochs were run per fold while when KU-HAR and PAMAP2 were target datasets, 40 epochs were run per fold. In addition, for all experiments, the sequence length was set to five, the number of hidden units was 128, the batch size was 64, and the exponentially decaying learning rate was set to an initial value of 0.9. Principal Component Analysis (PCA) was implemented to reduce the dimensionality of the higher dimension dataset to match the dimensionality of the smaller dimension dataset. The MMD sigma value was set to 0.1 and the KCCA regularization was set to 0.001. Finally, the joint loss omega (ω) value for the custom loss function was set to 0.5.

C. Training and Testing

For every fold for a given number of epochs, the source and target data are loaded in batches. Each batch is composed of 64 windows. Each window is created based on the sequence length provided. For example, given a sequence length of five, the window includes the current instance and the four previous instances from the dataset. To optimize learning, the learning rate is decayed exponentially at the end of every epoch and the

SGD algorithm with momentum is implemented to overcome noisy gradients.

There are three possibilities for training and testing: NT, GT, and joint training. NT is when the model is trained only on the source dataset, but tested on the target dataset with no explicit knowledge transfer or domain adaptation. GT is when the model is trained only on the target dataset and tested on the target dataset. Joint training is where source and target data are processed simultaneously to learn the joint feature space using MMD, KCCA, or Cosine Similarity, which is optimized at the end of every epoch via gradient descent. During training, the model learns classification using the source test dataset to mimic the real-world transductive transfer learning scenario where the target dataset is unlabeled, but the source dataset is labeled. However, during testing to determine the effectiveness of domain adaptation, the model is tested on the target dataset. For evaluating model performance, our experiments utilize the macro-averaged F1 score.

For datasets where the source dataset is larger than the target dataset, the target dataset is concatenated to itself until it is of the same size as the source dataset, thereby enabling continuous training and learning of the joint feature space. For instances where the source dataset is smaller than the target dataset, target instances are randomly selected so that the target dataset is of the same size as the source dataset. Random selection increases the likelihood that source data will be trained against a representative sample of activities from the target dataset.

D. Data Loading and Cross-Validation

Data from source and target files is split into 10 folds. Each fold in the source dataset is further split into train (80%) and test (20%) sets for training. Next, on-the-fly processing tasks such as Nearest Neighbors imputation for missing data and min-max normalization are applied as necessary. In addition, PCA dimension reduction is implemented for all datasets by training on the train set and applying the transformation to the train and test sets. PCA reduces feature differences between datasets by equalizing the number of dimensions. This is especially necessary in the case of MMD and Cosine Similarity where the number of dimensions must be equal between the source and target datasets.

E. Experiment Setup

To evaluate the effectiveness of the various cross-domain similarity metrics, the following three experiments were conducted.

- 1) **UCI-HAR \Rightarrow PAMAP2:** UCI-HAR as source dataset and PAMAP2 as target dataset.
- 2) **PAMAP2 \Rightarrow UCI-HAR:** PAMAP2 as source dataset and UCI-HAR as target dataset.
- 3) **PAMAP2 \Rightarrow KU-HAR:** PAMAP2 as source dataset and KU-HAR as target dataset.

For each of the above experiments, NT, GT, joint training with MMD, KCCA, and Cosine Similarity sub-experiments were performed.

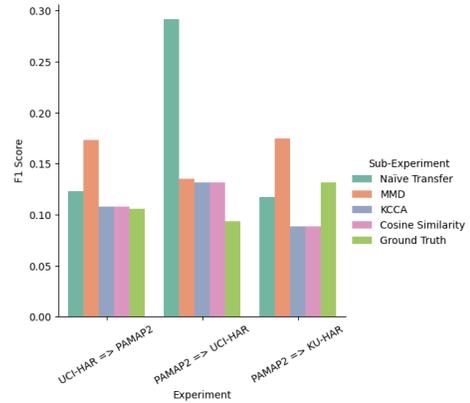


Fig. 2. Mean F1 scores across folds by experiment and sub-experiment.

Finally, a fourth experiment was conducted where the KU-HAR dataset as source domain was processed in increasing increments of 100,000 rows starting from 50,000 rows up to 650,000 rows against PAMAP2 as the target domain. The purpose of this experiment was to understand how increasing the size of the source dataset impacts domain adaptation utilizing MMD as the cross-domain similarity metric.

VI. RESULTS AND ANALYSIS

Our experiments resulted in four findings. First, MMD employed as a loss function outperforms KCCA and Cosine Similarity (Fig. 2) by a 45% improvement in mean F1 scores over the latter two loss functions. Second, unexpectedly, jointly training source and target datasets with MMD outperforms GT (where the model is trained and tested on the same dataset) in every experiment and results in a 45% improvement in mean F1 score over GT results. The low F1 scores for GT sub-experiments illustrate that, as discussed in Section I, HAR classification is a difficult deep-learning task. Third, in one of the three experiments (PAMAP2 \Rightarrow UCI-HAR) and in contravention of our hypothesis MMD underperforms NT. In NT sub-experiments, no explicit knowledge transfer or domain adaptation was performed, however utilizing larger source datasets (by number of instances) improves classification accuracy on target datasets because of the implicit knowledge transfer resulting from information and patterns contained in the larger source dataset. In addition, KCCA and Cosine Similarity underperform against NT in every experiment. Fourth, as hypothesized, incrementally increasing the size of the source dataset (by number of rows) while simultaneously reducing source and target domain discrepancy improves performance of the jointly trained model on the target dataset as the size of the source dataset increases.

A. UCI-HAR \Rightarrow PAMAP2

UCI-HAR dataset is the smallest dataset by number of rows (approximately 2% of the size of PAMAP2). For this experiment, use of MMD for joint training achieved the best mean F1 score (Fig. 2) across the folds (0.17) followed by NT (0.12), KCCA (0.11), Cosine Similarity (0.11), and GT (0.11).

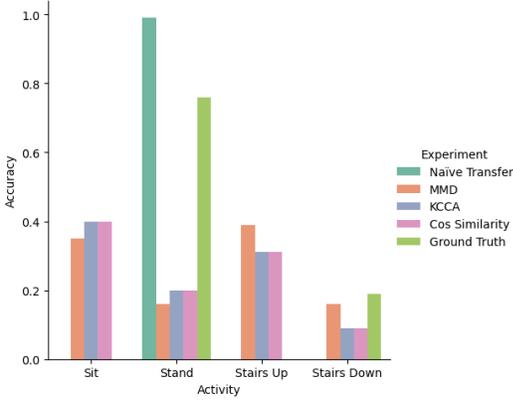


Fig. 3. UCI-HAR ⇒ PAMAP2

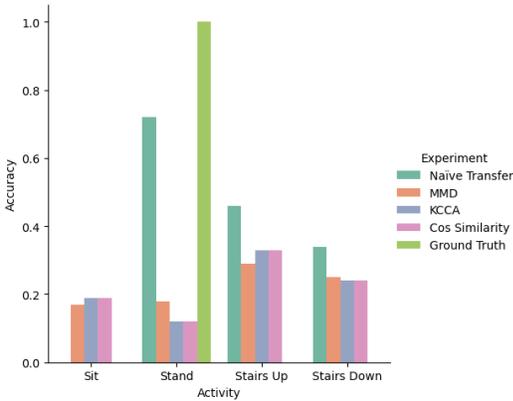


Fig. 4. PAMAP2 ⇒ UCI-HAR

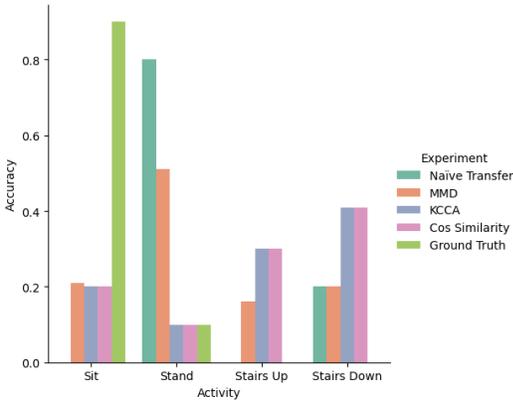


Fig. 5. PAMAP2 ⇒ KU-HAR

The comparatively similar results of the latter sub-experiments demonstrate that model performance benefited from domain adaptation made possible by joint training with MMD. Results at the activity level (Fig. 3) further illustrate that for three of the four activities cross-domain similarity metrics improved classification accuracy over NT.

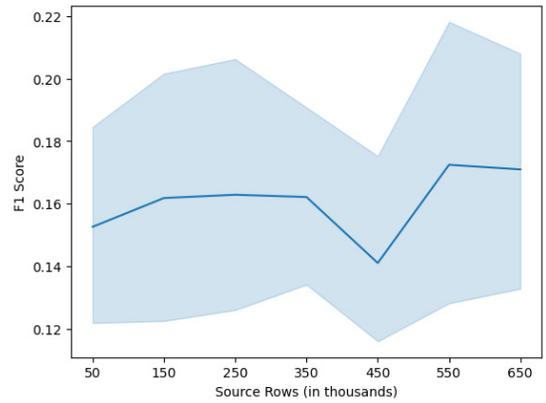


Fig. 6. Incremental processing of KU-HAR ⇒ PAMAP2 with mean results across the folds and a 95% confidence interval.

B. PAMAP2 ⇒ UCI-HAR

NT outperformed all other sub-experiments (Fig. 2) with a mean F1 score of 0.27 across the folds, including GT, which had the worse F1 score (0.09). MMD had a mean F1 score of 0.14 while KCCA, and Cosine Similarity had mean F1 scores of 0.13 each. As in the previous experiment, the GT score was the lowest indicating that the model is not able to effectively learn from the target dataset alone. However, the much larger PAMAP2 dataset (by number of instances) outperformed MMD and other cross-domain similarity metrics, indicating that it was less important to reduce the domain discrepancy between the two datasets than to use PAMAP2 as the training dataset for prediction on UCI-HAR. This was likely due to implicit knowledge transfer resulting from the use of a much larger dataset as source. However, as demonstrated at the activity level (Fig. 4) cross-domain similarity metrics improved classification accuracy over GT results.

C. PAMAP2 ⇒ KU-HAR

MMD had the highest mean F1 score (Fig. 2) of 0.17 followed by GT with a mean F1 score of 0.13, then NT at 0.12 and last KCCA and Cosine Similarity with mean F1 score of 0.09 each. These results demonstrate that MMD is the superior cross-domain similarity metric for domain adaptation. It also outperforms GT at the activity level (Fig. 5) by achieving higher classification accuracy for three out of four activities when compared to GT.

D. Incremental KU-HAR ⇒ PAMAP2 with MMD

Source data were processed in increments of 100,000 starting with 50,000 rows resulting in a mean F1 score of 0.15 to 650,000 rows resulting in a mean F1 score of 0.17. The overall results are as expected. As more source data are added, the average F1 scores improve (Fig. 6). Although there is a dip at the 450,000 rows (mean F1 score of 0.14), the results pick up at 550,000 rows and level out at 650,000 rows. The latter indicates that after 550,000 rows, additional source data do not improve training. Of particular interest is that while additional 100,000 rows improve results, the ratio of rows to F1 score

improvement is very small, indicating that the underlying HAR data are difficult to learn for the model. This conclusion is also supported by the low F1 scores for GT sub-experiments in the previous experiments.

VII. CONCLUSIONS AND FUTURE WORK

Unlike other research in HAR, which evaluated cross-domain adaptation at the user level [13,14], our research addressed the challenge of transductive, heterogeneous, parameter-based, closed-set adaptation at the domain level. Specifically, we focused on evaluating the effectiveness of cross-domain similarity metrics, MMD, KCCA, and Cosine Similarity, as loss functions in improving model classification accuracy. We demonstrated that jointly training a BLSTM-based model on labeled source data and unlabeled target data with MMD cross-domain similarity metric as a loss function improves model accuracy when evaluating human activity on the target domain. Moreover, a model trained in this way outperforms models trained with KCCA and Cosine Similarity as loss functions. Our results further show that naïve knowledge transfer between large source datasets and smaller target datasets results in better performance over models that are trained only on the smaller datasets. The results also demonstrate that smaller datasets as target domain benefit from jointly training a model on source and target domains with MMD and that the classification accuracy of this model is also better than a model trained solely on the target dataset.

Future research efforts will focus on addressing within-domain heterogeneity to evaluate its effectiveness in improving domain adaptation. Having concluded that MMD significantly outperforms other cross-domain similarity metrics, which affirms previous research [13], we can build on these findings to test how combining multiple datasets impacts domain adaptation.

ACKNOWLEDGMENT

We wish to thank the administration of the Johns Hopkins University Engineering for Professionals program for supporting our research and for helping to defray the costs of attending this conference.

REFERENCES

- [1] A. Arnold, R. Nallapati, and W. W. Cohen, "A Comparative Study of Methods for Transductive Transfer Learning," in *Proceedings of the Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)*, 2007, pp. 77–82.
- [2] K. You, M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Universal Domain Adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [3] P. Panareda Busto and J. Gall, "Open Set Domain Adaptation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [4] A. Farahani, S. Voghoei, K. Rasheed, and H. R. Arabnia, "A Brief Review of Domain Adaptation," arXiv:2010.03978, 2020.
- [5] J.-Y. He, X. Wu, Z.-Q. Cheng, Z. Yuan, and Y.-G. Jiang, "DB-LSTM: Densely-connected Bi-directional LSTM for human action recognition," *Neurocomputing*, vol. 444, pp. 319–331, 2021.
- [6] L. Alawneh, B. Mohsen, M. Al-Zinati, A. Shatnawi, and M. Al-Ayyoub, "A Comparison of Unidirectional and Bidirectional LSTM Networks for Human Activity Recognition," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2020, pp. 1–6.

- [7] W. Li, W. Nie, and Y. Su, "Human Action Recognition Based on Selected Spatio-Temporal Features via Bidirectional LSTM," *IEEE Access*, vol. 6, pp. 44 211–44 220, 2018.
- [8] S. Yu and L. Qin, "Human Activity Recognition with Smartphone Inertial Sensors using Bidir-LSTM Networks," in *Proceedings of the 3rd International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, 2018, pp. 219–224.
- [9] Y. Chang, A. Mathur, A. Isopoussu, J. Song, and F. Kawsar, "A Systematic Study of Unsupervised Domain Adaptation for Robust Human-Activity Recognition," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, no. 1, 2020.
- [10] N. D. Lane, Y. Xu, H. Lu, S. Hu, T. Choudhury, A. T. Campbell, and F. Zhao, "Enabling Large-Scale Human Activity Inference on Smartphones Using Community Similarity Networks (CSN)," in *Proceedings of the 13th International Conference on Ubiquitous Computing*, 2011, p. 355–364.
- [11] M. A. A. H. Khan, N. Roy, and A. Misra, "Scaling Human Activity Recognition via Deep Learning-based Domain Adaptation," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2018, pp. 1–9.
- [12] A. Stisen, H. Blunck, S. Bhattacharya, T. S. Prentow, M. B. Kjærgaard, A. Dey, T. Sonne, and M. M. Jensen, "Smart Devices are Different: Assessing and Mitigating Mobile Sensing Heterogeneities for Activity Recognition," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, 2015, p. 127–140.
- [13] R. Ding, X. Li, L. Nie, J. Li, X. Si, D. Chu, G. Liu, and D. Zhan, "Empirical Study and Improvement on Deep Transfer Learning for Human Activity Recognition," *Sensors*, vol. 19, no. 1, 2019.
- [14] M. Hamidi, A. Osmani, L. Rасаq, G. Dogan, and N. Alotaibi, "Multiview Representation Learning for Human Activity Recognition," in *Proceedings of the IEEE 9th International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, 2022, pp. 1–5.
- [15] A. Hosseini, D. Zamanzadeh, L. Valencia, R. Habre, A. A. T. Bui, and M. Sarrafzadeh, "Domain Adaptation in Children Activity Recognition," in *Proceedings of the 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2019.
- [16] G. Wilson and D. J. Cook, "Adversarial Transfer Learning," arXiv:1812.02849, 2018.
- [17] S. Feng and M. F. Duarte, "Few-shot learning-based human activity recognition," *Expert Systems with Applications*, vol. 138, p. 112782, 2019.
- [18] A. Reiss and D. Stricker, "Introducing a New Benchmarked Dataset for Activity Monitoring," in *Proceedings of the 16th IEEE International Symposium on Wearable Computers (ISWC)*, 2012, 2012.
- [19] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A Public Domain Dataset for Human Activity Recognition Using Smartphones," in *Proceedings of the 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013*, 2013.
- [20] N. Sikder and A.-A. Nahid, "KU-HAR: An open dataset for heterogeneous human activity recognition," *Pattern Recognition Letters*, vol. 146, pp. 46–54, 2021.
- [21] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.
- [22] M. Yang, J. Moon, S. Yang, H. Oh, S. Lee, Y. Kim, and J. Jeong, "Design and Implementation of an Explainable Bidirectional LSTM Model Based on Transition System Approach for Cooperative AI-Workers," *Applied Sciences*, vol. 12, no. 13, 2022.
- [23] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artificial Intelligence Review*, vol. 53, no. 8, 2020.
- [24] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep Domain Confusion: Maximizing for Domain Invariance," arXiv:1412.3474, 2014.
- [25] T.-K. Kim, S.-F. Wong, and R. Cipolla, "Tensor Canonical Correlation Analysis for Action Classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.
- [26] Y.-R. Yeh, C.-H. Huang, and Y.-C. F. Wang, "Heterogeneous Domain Adaptation and Classification by Exploiting the Correlation Subspace," *IEEE Transactions on Image Processing*, vol. 23, no. 5, pp. 2009–2018, 2014.
- [27] "CCA-Zoo," <https://cca-zoo.readthedocs.io/>, accessed: 2022-12-11.