

Article

Domain Adaptation Methods for Lab-to-Field Human Context Recognition [†]

Abdulaziz Alajaji ¹, Walter Gerych ¹, Luke Buquicchio ¹, Kavin Chandrasekaran ¹, Hamid Mansoor ², Emmanuel Agu ^{1,*} and Elke Rundensteiner ¹

¹ Data Science, Worcester Polytechnic Institute, Worcester, MA 01609, USA

² Department of Computer Science, University of Victoria, Victoria, BC V8P 5C2, Canada

* Correspondence: emmanuel@wpi.edu

[†] This paper is an extension version of the conference paper: Alajaji, A.; Gerych, W.; Chandrasekaran, K.; Buquicchio, L.; Mansoor, H.; Agu, E.; Rundensteiner, E. Triplet-based Domain Adaptation (Triple-DARE) for Lab-to-field Human Context Recognition. In Proceedings of the 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), IEEE, Pisa, Italy, 21–25 March 2022.

Abstract: Human context recognition (HCR) using sensor data is a crucial task in Context-Aware (CA) applications in domains such as healthcare and security. Supervised machine learning HCR models are trained using smartphone HCR datasets that are scripted or gathered in-the-wild. Scripted datasets are most accurate because of their consistent visit patterns. Supervised machine learning HCR models perform well on scripted datasets but poorly on realistic data. In-the-wild datasets are more realistic, but cause HCR models to perform worse due to data imbalance, missing or incorrect labels, and a wide variety of phone placements and device types. Lab-to-field approaches learn a robust data representation from a scripted, high-fidelity dataset, which is then used for enhancing performance on a noisy, in-the-wild dataset with similar labels. This research introduces Triplet-based Domain Adaptation for Context REcognition (*Triple-DARE*), a lab-to-field neural network method that combines three unique loss functions to enhance intra-class compactness and inter-class separation within the embedding space of multi-labeled datasets: (1) domain alignment loss in order to learn domain-invariant embeddings; (2) classification loss to preserve task-discriminative features; and (3) joint fusion triplet loss. Rigorous evaluations showed that *Triple-DARE* achieved 6.3% and 4.5% higher F1-score and classification, respectively, than state-of-the-art HCR baselines and outperformed non-adaptive HCR models by 44.6% and 10.7%, respectively.

Keywords: ubiquitous computing; domain adaptation; context aware systems; machine learning



Citation: Alajaji, A.; Gerych, W.; Buquicchio, L.; Chandrasekaran, K.; Mansoor, H.; Agu, E.; Rundensteiner, E. Domain Adaptation Methods for Lab-to-Field Human Context Recognition. *Sensors* **2023**, *23*, 3081. <https://doi.org/10.3390/s23063081>

Academic Editor: Jungpil Shin

Received: 26 December 2022

Revised: 28 January 2023

Accepted: 29 January 2023

Published: 13 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

There is a great potential for context-aware (CA) systems to impact many fields, such as healthcare, smart homes, and security [1]. An important part of CA systems is Human Context Recognition (HCR), the process of determining the user's current state. Several definitions exist, but ours is as follows: Human Context is a tuple <Activity, Prioception> comprising of the user's current activity (e.g., walking, standing) and the phone's placement in the user's body (the "prioception") (e.g., in a bag, pocket, or hand). We focus on CA and HCR on smartphones, which are now almost ubiquitously owned and possess a wide variety of sensors such as accelerometers, gyroscopes, and position detectors. There are two popular study designs for collecting HCR datasets for supervised machine learning involving human participants: (1) *scripted* [2] or (2) *in-the-wild* [3]. *Scripted* studies involve participants carrying out a series of tasks in a prescribed sequence while being monitored by a human proctor and having their smartphone sensor data continuously recorded by an app. After the users' sensor data has been collected, human proctors label the data with the locations they were in. On the other hand, *in-the-wild* studies entail data collection in the

subjects' natural environments over the course of several days. Smartphone sensors collect data in real time, and users are periodically prompted to report their current context so that the app can annotate their collected data. **Motivation.** Due to the high costs associated with gathering labeled data, it is challenging to gather adequate labels of sufficient quality for fully supervised learning in many real-world, in-the-wild scenarios. This presents a challenge for fully supervised learning methods. Thus, it is highly desirable to design innovative learning methods that obviate the need for fully annotated data but instead leverage unlabeled data.

Problems with In-the-wild datasets that hinder HCR performance: Due to the high precision of sensor data and high quality context labeling, supervised HCR classification models often achieve excellent precision on scripted datasets. *DeepContext*, a cutting-edge deep learning HCR model, achieved 91.2% accuracy on a scripted dataset [4]. Nevertheless, scripted datasets are not realistic as the contexts visited and visit patterns do not reflect reality. It is important that HCR models are precise on datasets collected in the wild, which are more reflective of actual deployment circumstances. However, HCR models often exhibit reduced performance on more naturalistic, in-the-wild datasets. Vaizman, for example, achieves 71.7% accuracy using a Multi-Layer Perceptron (MLP) model trained on a HCR dataset gathered in the wild [3]. Essentially, the accuracy of state-of-the-art HCR models on scripted datasets decreases by 19.5% when compared to their accuracy on datasets gathered in the wild. This performance disparity is a result of in-the-wild dataset problems, including:

(1) *Diversity of phone placements:* Or locations where smartphone devices are typically kept (prioceptions). When the phone is carried in various prioceptions, sensor signals for the same activity have distinct characteristics [5,6]. In fact, as seen in Figure 1, prioception is one of the most significant causes of variation in smartphone context sensor data [3]. Smartphone users may opt to carry their devices in a backpack, in their hand, or in a coat pocket when engaging in a certain activity (e.g., walking).

(2) *Weak, noisy, and missing context labels:* Supervised machine learning algorithms have difficulty when users quit supplying labels [7] due to lack of time or because they offer incorrect labels [8].

(3) *Diversity of smartphone models:* Typically, while participants in scripted HCR studies use a specific study phone model provided by the proctor, participants in HCR field (in-the-wild) studies utilize their own mobile devices. Different smartphone models can record sensor values for the same context that vary by as much as 30%, which presents an additional challenge for machine learning classifiers [6].

Lab-to-field methods: Have recently been proposed as promising approaches for achieving acceptable HCR performance on noisy, low-quality labeled datasets collected in the wild [9]. The goal of lab-to-field techniques is to train highly accurate machine learning models on scripted HCR datasets, which are subsequently adapted for *in-the-wild* datasets with the expectation of maintaining great performance. (*Going from Lab-to-field*). Due to differences between the contexts visited by participants in the scripted study visited vs. the in-the-wild study, as well as the order in which they visited those contexts and the length of their visits, which were quite different from in-the-wild scenarios, there are substantial discrepancies between the distribution of features derived from scripted and unscripted datasets, often known as the *covariate shift problem* [9–11].

Domain Adaptation (DA): One of the primary approaches of adapting neural networks to deal with the covariate shift problem is called Domain Adaptation (DA), and it is a transductive transfer learning method. DA has been utilized in other related disciplines, such as object detection in computer vision and the issues posed by the unpredictability in wearable sensor location in ubiquitous computing [5,11]. Utilizing both labeled source samples (e.g., scripted dataset) and unlabeled target samples (e.g., in-the-wild), each of which has a unique distribution, Unsupervised Domain Adaptation (UDA) attempts to learn a deep learning model that can accurately predict the labels of unlabeled (e.g., in-

the-wild) data samples [5,12]. Figure 2 presents an overview of the topic, its obstacles, and our strategy.

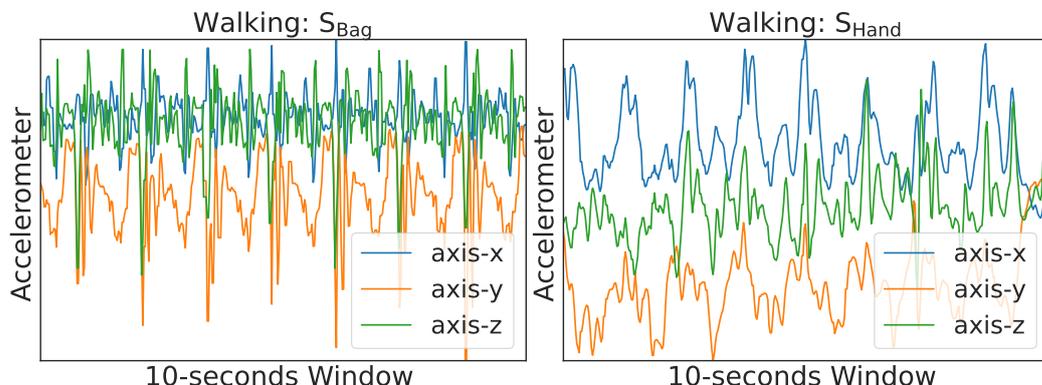


Figure 1. The effect of different phone placements on sensor data can be seen in triaxial accelerometer tracings for the same walking activity but with different phone perceptions.

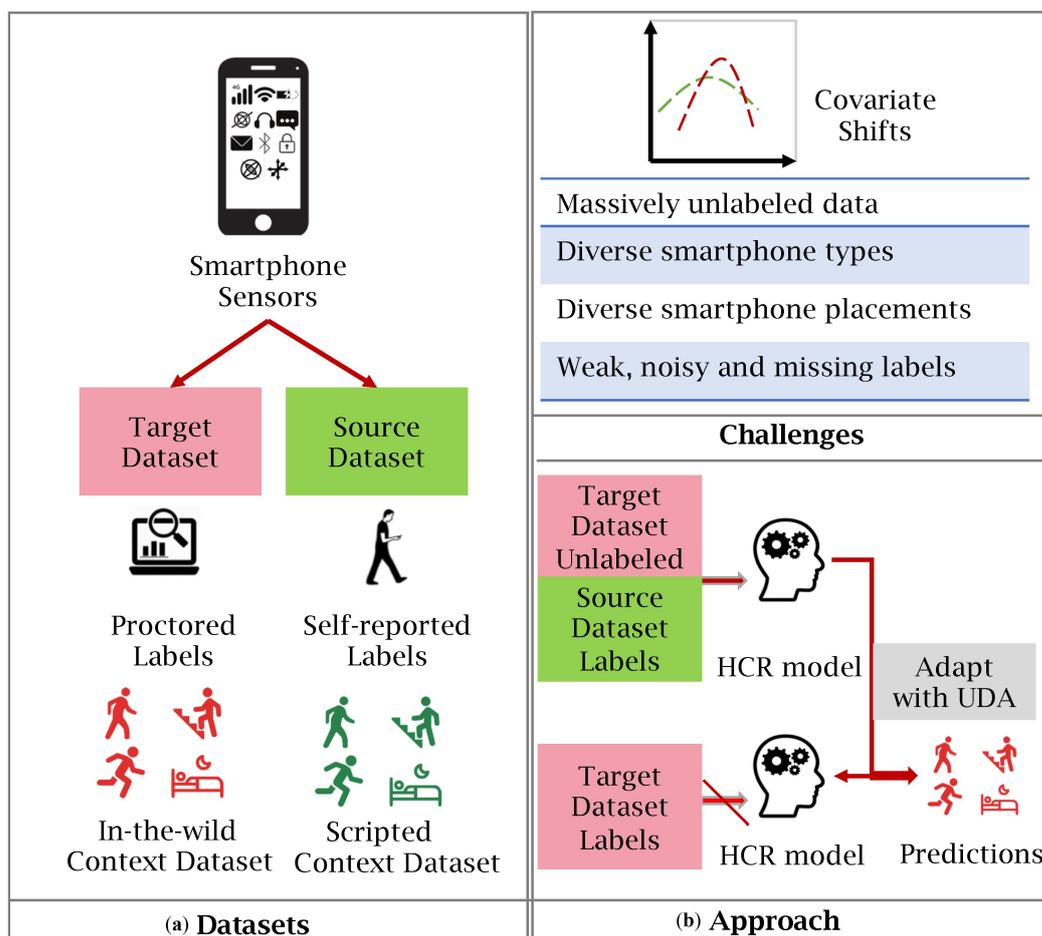


Figure 2. (a) The two kinds of smartphone context data used in this work. (b) Overview of Triple-DARE's problem and approach.

Challenges. For the application of UDA to the lab-to-field generalization of smartphone context recognition, two significant obstacles must be overcome. Initially, the previously noted data concerns with in-the-wild datasets (the diversity of phone placements, noisy labels, and the variety of smartphone models) must be resolved. Second, it is difficult to build a strategy for knowledge transfer from a scripted dataset to a more realistic, but significantly noisier, unscripted dataset with sparse labels.

Our approach. Recent empirical accomplishments of the triplet loss function in the facial recognition task [13,14], where changes of the same person's face pictures are tightly mapped in a learned embedding space, has inspired us. We believe that HCR sensor data can benefit from a similar approach even in cases where sensor signatures associated with the same context often vary. Our opinion is also consistent with the findings of Khaertdinov et al., who utilized triplet loss to reduce the impacts of subject variability and enhance model generalizability [15].

We present *Triple-DARE*, a Lab-to-field UDA approach that may harness the vast volumes of unlabeled smartphone HCR data collected in the wild, therefore reducing the requirement for human-annotated labels. *Triple-DARE* uses both handcrafted features and features autonomously extracted from raw sensor data by a CNN. *Triple-DARE* uses domain alignment and triplet losses to learn domain-invariant embeddings with discriminative capabilities for context predictions learned from unlabeled samples. *Triple-DARE* collects domain-invariant features that increase the effectiveness of predicting contexts under unobserved prioceptions.

In addition, to support our DA strategy, we used HCR datasets with coincident scripted and in-the-wild data with equivalent context labels collected in both studies [1]. These coincident datasets and identical context labels guarantee that there is a feature representation of contexts that is shared throughout scripted and unscripted datasets, which is a crucial need for the DA strategy. By only using context labels that were collected in a scripted study during model development, we are able to demonstrate that our method is applicable to HCR models that are implemented in realistic environments by employing DA in order to reduce the impact of potentially noisy labels while maintaining HCR performance on a dataset collected in the wild. *Triple-DARE* outperforms state-of-the-art baselines by 3.79% and 1.89% gains in F1-score and accuracy, respectively, and outperforms HCR models without *Triple-DARE* by 39% and 14.7% in F1-score and accuracy, respectively.

State-of-the-art limitations. There is a paucity of research on laboratory-to-field generalization approaches for HCR. Previously proposed lab-to-field methods include importance re-weighting [9,16] and Positive Unlabeled (PU) classifiers [1]. DA has been used in the past to solve the problem of variable on-body locations of wearable sensors [5,17] but not for HCR. The majority of prior DA work for wearable sensors focuses on decreasing the global distribution gap across domains while learning common feature representations [5,17]. However, we observe that even if the global distribution is effectively aligned, samples from different domains with the same label may be mapped such that they are far apart in feature space. Thus, in addition to using a domain alignment loss [18,19], *Triple-DARE* improves intra-class compactness and inter-class separability by utilizing a joint fusion triplet loss [12,13] intended for multi-labeled datasets. Moreover, unlike other existing methods for dealing with domain shifts [1,9,17,20], we do not utilize target labels in the target (in-the-wild) dataset, instead following the UDA problem setting outlined by Chang et al. [5].

Contributions. The main contributions of this paper are:

1. We provide *Triple-DARE*, a novel UDA deep-learning architecture that employs a scripted dataset to increase the HCR accuracy of predicting contexts in the wild. *Triple-DARE* employs a domain alignment loss for domain-independent feature learning, a classification loss to keep task-discriminative features, and a joint fusion triplet loss to improve intra-class compactness and inter-class separation;
2. We carefully assessed *Triple-DARE*, comparing it to numerous state-of-the-art unsupervised domain approaches, including DAN [18], CORAL [19], and HDCNN [17], and bench-marking advances in HCR performance on target domains in multiple application scenarios. Our ablation study demonstrates that all component of *Triple-DARE* contributes non-trivially;
3. We illustrate that *Triple-DARE* minimizes in-the-wild dataset problems when compared to state-of-the-art DA algorithms, delivering improved prediction accuracy

on the target (in-the-wild) domain without the requirement for large amounts of source-labeled samples.

The rest of this paper is organized as follows. Section 2 includes the background. Section 3 reviews the related work. Section 4 describes our proposed approach. Section 5 presents our evaluation and findings. Section 6 outlines the limitations of our work and plans for future work. Section 7 finally concludes the paper.

2. Background

2.1. Covariate Shifts

The term “Covariate Shifts” was first introduced by Shimodaira [21], and is described as changes in the distribution of the input x . While there are other types of existing shifts [10], the most researched type of shift is covariate shift. Covariate shift occurs when data are generated based on a model $P(y|x)P(x)$ whose distribution $P(x)$ varies between the training and test situations. While there is some ambiguity in the definitions of covariate shift in the literature, we found the definition provided by Moreno-Torres et al. [10] to be the most relevant, given by the following conditions:

$$P_{\text{training}}(y | x) = P_{\text{testing}}(y | x) \text{ and } P_{\text{training}}(x) \neq P_{\text{testing}}(x), \quad (1)$$

where $P_{\text{training}}(x)$ and $P_{\text{testing}}(x)$ represents training and testing input distributions, respectively.

Collecting smartphone sensor data in the wild often results in naturally occurring variations in the data. When trying to leverage models trained on scripted data to improve performance on an in-the-wild dataset with similar context labels, we encounter a data shift problem known as covariate shift, where the distribution of features differs across training and test scenarios. Specifically, the covariate shift problem is caused by substantial differences between the distributions of features extracted from scripted and in-the-wild datasets [9–11]. More broadly, because real-world applications must face some type of dataset shifts, it is critical to address the covariate shift problem for the successful deployment of machine learning models in the wild [10].

2.2. Sensor Data Collection Studies

Inaccurate labeling or unrealistic user behavior are two common problems with context datasets. There are two types of research designs used to gather HCR datasets: *scripted* [2] or *in-the-wild* [3]. *Scripted* studies are usually conducted in a laboratory setting. Participants follow a scripted series of steps to complete a series of tasks in a predetermined order, while an app on their smartphones continuously logs data from those devices’ sensors. Human proctors annotate the sensor data with corresponding context labels. In *unscripted* (“or in the wild”) studies, data is gathered over days while people live their lives in the actual world. A smartphone continually gathers smartphone sensor data as individuals go about their daily lives. Periodically, subjects annotate their data with labels for the contexts they have visited. While the scripted technique for HCR data collection produces accurate labels suited for supervised machine or deep learning that are exceptionally precise and consistent, the contexts visited and sensor data acquired in each context are not reflective of the actual world. HCR research conducted in the wild yields more accurate results. However, certain context labels may be missing since people forget to label when their lives get busy. Additionally, some labels may be incorrect due to human labeling errors [8].

2.3. DARPA WASH Project: Motivation Use Case

The Warfighter Analytics utilizing Smartphone for Healthcare (WASH), a DARPA-funded project, investigates passive smartphone evaluation of traumatic brain injury and infectious disease [22]. This will offer a current evaluation of the warfighter’s battle readiness. Initially, the target groups are active-duty military personnel and veterans, but the findings will also apply to civilians. In the intended use case, the WASH smartphone application will passively collect sensor data throughout each day. Each day’s data is then sent

overnight to the cloud for processing. These data will be analyzed in the cloud by disease inference algorithms to provide a *bioscore* (or probability of illness) for each warfighter.

Program phases: The WASH program is separated into two sections. Phase one is identifying particular smartphone user scenarios for conducting targeted health evaluations. Phase two entails the development of real TBI and infectious illness assessment systems for smartphone users. In phase one, we conducted research and compiled a list of smartphone biomarkers indicative of TBI and infectious disorders, as well as their accompanying settings. Our team performed user surveys to acquire labeled data for these settings and developed HCR models to infer these smartphone contexts derived from labeled sensor data. In Table 1, the intended disease-specific tests or biomarkers relating to each of these settings are detailed. The University of Massachusetts Medical School’s specialists in traumatic brain injury (TBI) and infectious diseases were consulted while compiling our list of illness tests and situations (UMMS). As an example, trembling hands are a symptom of TBI. In phase one, our team will perform user research and develop deep learning models to recognize smartphone users holding their devices. In the second step, we will analyze if the user’s hand is shaking. This study is limited to context recognition. Actual context-specific disease assessment research is not covered.

Table 1. Context-specific diagnostic tests for traumatic brain injury (TBI) and infectious diseases with their relevant human contexts.

Traumatic Brain Injury	
Diagnostic Test	Context
Inferior Reaction Time	< Interaction with Phone, in Hand, *, *>
Elevated Light Sensitivity	<*, in Hand, *, *>
Pupil Dilation	< Interaction w/ Phone, in Hand, Typing, *>
Hands Shaking	<*, in Hand, *, *>
Slurred Speech	<Speaking into Phone, *, *, *>
Infectious Diseases	
Ailment Test	Test Context
Elevated Frequency of Coughing	<Coughing, *, *, *>
Elevated Frequency of Sneezing	<Sneezing, *, *, *>
Rate of Heart at Rest	<Sitting, in Pocket, *, *>
Elevated Toilet use Frequency	<Using Toilet, *, *, *>
Variation in respiration	<Sleeping, on Table, *, *> <Exercising, *, *, *>
Both TBI and Infectious Disease	
Ailment Test	Test Context
Elevation In Activity Transition Period	<Lying down, in Pocket, *, *> <Sitting, in Pocket, *, *> <Standing, in Pocket, *, *>
Variation in Sleep Quality	<Sleeping, *, *, *>
Variation in Gait	<Walking, in Pocket/Hand, *, *>

“*” denoting a wildcard.

2.4. Our Coincident Data Gathering Study Approach

Using an innovative coincident study design, we conducted scripted and in-the-wild data collection studies to collect labeled data in the same contexts shown in Table 2. This coincidental study enables the use of machine learning techniques that combine the precision of scripted labels with the natural context visit patterns of studies conducted in the wild. Our in the wild study followed a similar methodology to the *Extrasensory* study. The smartphone app constantly collected sensor data from 103 participants’ smartphones as they went about their daily lives. The users were subsequently prompted to self-report context labels. Our scripted study was conducted in a specific laboratory, campus building, or route. The smartphone app systematically collected data from 100 participants

who visited the contexts listed in Table 2. The scripted data collection session lasted approximately one hour per subject, and human proctors monitored and annotated the data manually.

Table 2. Contexts for which data was obtained as part of our WASH Study Collected Contexts—split down into 25 binary labels.

Phone Placement	
Phone in Hand	Phone in Bag
Phone in Table—Facing Up	Phone in Table—Facing Down
Phone in Pocket	
Long Activity	
Standing	Sleeping
Walking	Sitting
Stairs—Going Up	Stairs—Going Down
Talking On Phone	Trembling *
Jumping *	Jogging
Typing	In Bathroom
Lying Down	Running
Short Activity	
Coughing *	Sneezing *
Sitting Down (transition) *	Sitting Up (transition) *
Standing up (transition) *	Laying Down (transition) *

*: Labels associated with contexts collected in the scripted study only.

2.5. Weakly Supervised Learning (WSL)

In supervised learning tasks, predictive models are trained on annotated training examples, common types of which are classification and regression models. A training example consists of an input feature vector (also known as an instance) and a label that is associated with it (or ground-truth). Due to the high costs associated with gathering labeled data, it is difficult to gather adequate labels of sufficient quality for fully supervised learning in many real-world scenarios, such as our study of HCR using data collected in the wild. This presents a challenge for fully supervised learning. Various types of weak (or inaccurate) labels can occur in such practical scenarios, including several encountered in our mobile HCR scenarios, requiring innovative learning methods. According to a recent survey by Zhou et al. [7], weakly supervised learning can be categorized into three types:

1. *Inexact supervision* in which only coarse-grained labels are provided. Due to the nature of the annotation process of sensor data, only a few selected sub-segments of each training sensor segment can be considered accurate representatives of their respective labels. However, their precise length, as well as their position within the segment, is unknown;
2. *Inaccurate supervision* in which data labels are not always correct. For example, in-the-wild datasets often depend on self-reported labels. However, users may erroneously provide wrong labels as they might not recall which contexts they previously visited accurately;
3. *Incomplete supervision* that utilizes unlabeled training data. When study participants get busy with their lives, they might forget to label the data in the dataset, which means that some of the context labels might be missing from the dataset.

For these various forms of weak labeling, innovative learning methods that are trained under weak supervision are desired [7].

3. Related Work

Lab-to-field generalization. Our Lab-to-field method tries to leverage a scripted dataset that contains high-quality, relatively cheaper to obtain, ground truth labels to improve HCR model performance on an in-the-wild dataset [9]. The ability of deep neural networks to generalize to real-world scenarios, where domain shift is expected, is a critical challenge in smartphone HCR developed for in-the-wild data [1,23]. Importance re-weighting [9,16] and Positive Unlabeled (PU) classifiers [1] are two methods that have been presented in the past to deal with covariate shifts. The transferability of HCR findings from the laboratory to the real world is an area that has received little attention. However, a related study employed importance re-weighting to modify a linear logistic regression model for application with data from wearable electrocardiograms (ECGs). When applied to deep neural networks, however, these techniques have a diminished impact on performance [24]. Unlike other existing methods for dealing with domain shifts [1,9,17,20], our approach does not require target domain labels.

Domain Adaptation (DA). Prior research has demonstrated substantial progress in adapting deep neural networks to various related domains [11]. Recent deep DA methods are either discrepancy-based approaches that minimize a discrepancy metric over feature distributions [18,19], or adversarial-based approaches [25] that aim to maximize domain confusion. The Deep Adaptation Network (DAN) [18] minimized the mean distance between two feature distributions in a Reproducing Kernel Hilbert Space (RKHS), effectively matching higher-order statistics of the two distributions. On the other hand, the deep Correlation Alignment (CORAL) [19] technique proposed matching the mean and covariance of two distributions. Other strategies have used an adversarial loss to maximize domain confusion [25]. The domain alignment loss, one component we utilized in *Triple-DARE*, is based on DAN.

DA for wearable sensor data. In ubiquitous computing, several DA techniques have been developed to transfer a trained model to a new dataset with similar characteristics [5,17,26,27]. Previous work has shown that DA can be used to unsupervisedly learn domain-invariant accelerometer [5,17] and gyroscope [5] features from sensor data by minimizing a discrepancy distance in the Convolutional Neural Network (CNN) embedding, thereby mitigating the effects of variability in wearable sensor placement. HDCNN [17] looked at whether or not a model pre-trained on smartphone data could be used with unlabeled smartwatch data. The researchers used Kullback–Leibler (KL) divergence and a discrepancy-based technique to transfer the trained model from smartphones to the unlabeled wristwatch data. Stratified Transfer Learning (STL) [26] is a DA method for adapting on-body sensor-based activity recognition tasks to various sensor placements (wrist, chest, leg, etc.). It also maps source and target domain data into the same subspace where distances can be computed, exploiting intra-affinity of classes to transform intra-class knowledge. UDA methods based on Variational Auto Encoders have been used for adapting models to work on another dataset, and have been applied on binary sensors for smart-homes applications [27]. DA was also used to adapt models to subject variability [20], using multi-domain adaptation to address target label shift by incorporating the target domain label distribution in the training process.

The majority of existing work solely focuses on domain-general feature representation learning with the goal of decreasing the global distribution disparity [5,17]. While STL proposed a way to perform intra-class transfer by minimizing the discrepancy between feature distributions of instances of the same class, this approach does not scale well to large-scale datasets, especially datasets with a large number of class labels. By employing a joint fusion triplet loss, our study expands upon previous efforts to enhance intra-class compactness and inter-class separability [12,13]. A summary of the related work is included in Table 3.

Table 3. Related work summary.

Research Work	Method	Type of Data	Task	Lab-to-Field	Distribution Discrepancy Minimization
Natarajan et al. [9]	Importance-reweighting	Wearable electrocardiogram sensor data	Cocaine use detection	×	No
Alajaji et al. [1]	Positive Unlabeled Classifier	Smartphone sensor data	Context recognition	×	No
Chang et al. [5]	Feature matching and confusion maximization	Wearable sensor data	UDA for activity recognition under sensor position variability		Global only
Long et al. [18]	MK-MMD	Images	UDA for cross-dataset image classification		Global only
Sun et al. [19]	Correlation Alignment	Images	UDA for cross-dataset image classification		Global only
Khan et al. [17]	KL Divergence	Smartphone and smartwatch sensor data	DA for cross-device activity recognition		Global only
Chen et al. [26]	Stratified Transfer Learning	Wearable sensor data	DA for cross sensor placement		Non-scalable intra-class separation
Sanabria et al. [27]	Variational Autoencoder	Binary event sensor data	DA for cross-user activity recognition		Global only
Wilson et al. [20]	Weak-supervision using target label distribution	Wearable sensor data	DA for cross-user activity recognition		Global only and utilized target labels

4. Proposed Triple-DARE Methodology

4.1. Problem Formulation

Our study makes use of two datasets annotated with the same context labels as those in Table 2: (1) a scripted dataset (source) with high-fidelity labels, and (2) an in-the-wild dataset (target) with identically labeled data. These two datasets are acquired using a study design with coincident data collection, where data was gathered for the same context labels in separate scripted and in-the-wild settings. With respect to UDA, there are source domain labeled samples and target domain unlabeled samples, both of which have distinct data distributions. Our objective is to use both labeled and unlabeled target data to train a classifier that performs effectively on the target domain. In a more formal sense, we have labeled samples $\mathcal{D}_s = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^{n_s}$ and unlabeled samples $\mathcal{D}_t = \{(\mathbf{x}_i^t)\}_{i=1}^{n_t}$, with n_s and n_t standing for the number of samples in the source and target domains, respectively. The feature space and label space are identical across the source and target domains $\mathcal{X}_s = \mathcal{X}_t$ and $\mathcal{Y}_s = \mathcal{Y}_t$, but the marginal distribution is not the same ($P_s(x_s) \neq P_t(x_t)$). The conditional distributions are assumed to be equal $P_s(y_t|x_s) = P_t(y_t|x_t)$ in the two domains.

We refer to x as the feature vector, and y as a multi-label output vector representing the human context where each label used is a binary output (e.g., walking vs. not walking). Presumably, the source and target tasks are identical. At first, we use the labeled source dataset to train the HCR model. Once the HCR model has been trained, it may be used to detect unlabeled contexts in the target dataset by integrating unlabeled data from the target dataset into the training set.

4.2. Overview

In the model in Figure 3, the framework of *Triple-DARE* is depicted. *Triple-DARE* extracts two distinct kinds of feature sources from the source and target datasets: (1) Hand-crafted features based on temporal and spectral information processed by a feed-forward neural network and (2) three-axis sensors' raw data is put into a convolutional neural network (CNN) that uses a soft attention method to identify prominent characteristics from the data. *Triple-DARE* consists of three main parts: (1) A domain alignment loss \mathcal{L}_d to generate domain-invariant embeddings; (2) a classification loss \mathcal{L}_{cls} in order to preserve task-discriminating characteristics; and (3) a joint fusion triplet loss \mathcal{L}_{tri} that improves intra-class compactness and inter-class separation in the learned embedding space by learning comparable contexts represented by differences in sensor inputs.

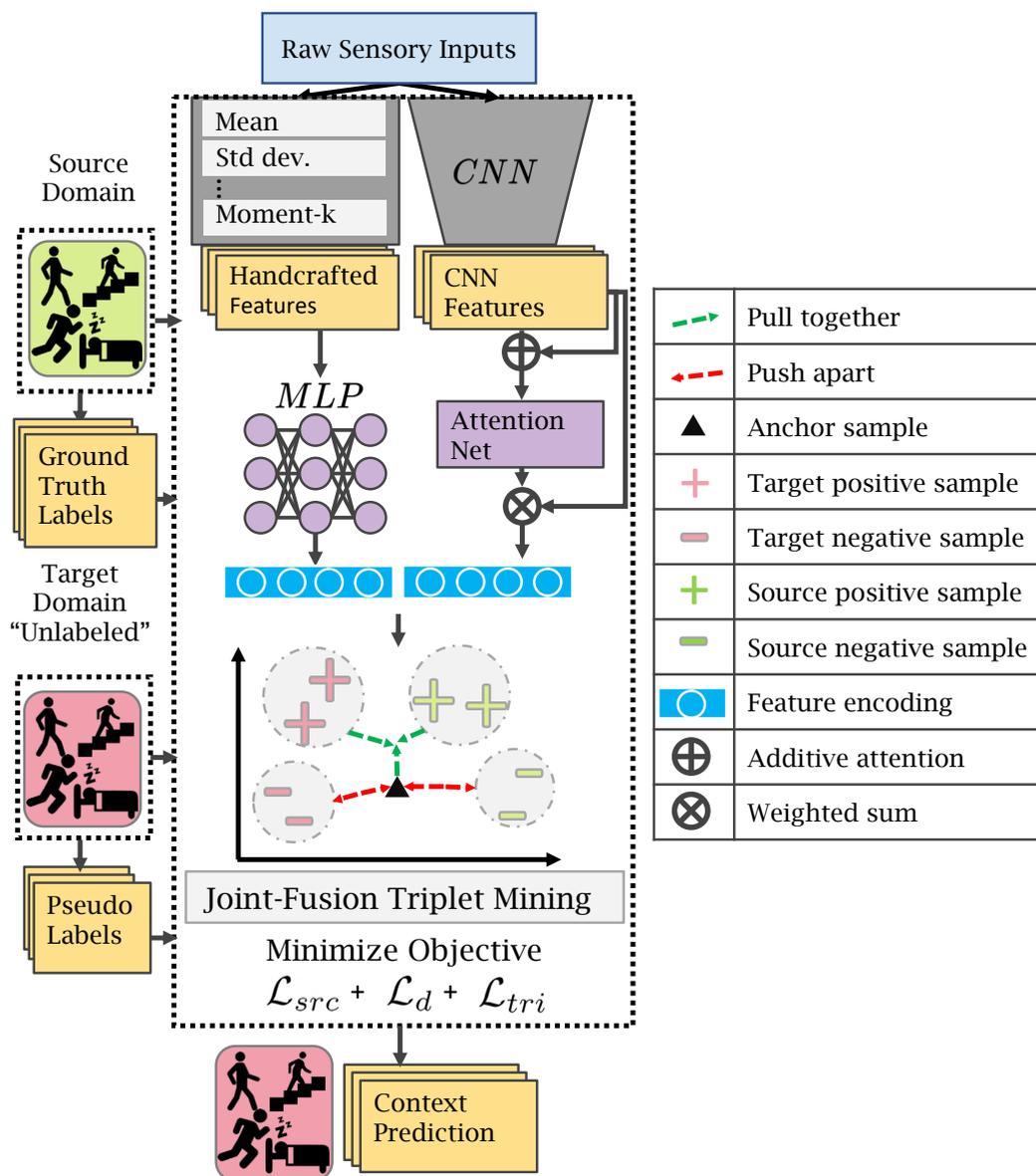


Figure 3. *Triple-DARE* framework.

The final result is used for context predictions with multiple labels. For instance, according to our definition of context as an $\langle \text{Activity, Phone placement} \rangle$, a context could be $\langle \text{“Sitting”, “In Bathroom” with “Phone In Hand”} \rangle$. Our ultimate aim is to minimize the

cost function $C(\cdot)$ in order to execute context predictions by learning discriminative and domain-invariant embeddings:

$$C(\theta) = \lambda_1 \mathcal{L}_{\text{cls}}^\theta + \lambda_2 \mathcal{L}_{\text{d}}^\theta + \lambda_3 \mathcal{L}_{\text{tri}}^\theta, \quad (2)$$

where θ are model parameters, λ_1 , λ_2 , and λ_3 are balancing coefficients. Subsequent sections elaborate on this procedure and each kind of loss objective. Each loss function is applied to the three feature encodings produced by our deep network, namely the MLP, CNN, and joint fusion encodings.

4.3. Feature Generation

From the raw sensory inputs for a specific smartphone context dataset, we generate two views. The first is a vector derived by manually applying handcrafted features to all accessible sensors. The second view is comprised only of raw three-axial sensors. We use distinct feature encoders for each input view type. Specifically, (1) Handcrafted feature encoding using a Multi-Layer Perceptron (MLP), which is adopted from Ref. [28] and (2) attention-based CNN encoder [4] for raw sensor data. The two resulting feature encodings are then concatenated to yield a joint fusion encoding.

We use data from five sensors: accelerometer, gyroscope, GPS, magnetometer, and phone status (discrete attributes such as whether the phone screen is locked or unlocked). At the sliding window level, we compute statistical, time-based, and frequency-based features for each of the sensor modalities (10-s in this application). Then, we implemented Z-score normalization $z_i = \frac{x_i - \bar{x}}{s}$ through subtraction of the mean and division by the standard deviation. Handcrafted features, including 188 features borrowed from Ref. [3], are utilized to build a vector that is then put into a feed-forward network. Table 4 lists some of the handcrafted features incorporated in our work.

Table 4. A small selection of the handcrafted features applied to accelerometer, gyroscope, and magnetometer data that we use, taken from Refs. [3,29].

Feature	Formulation
Tri-axial sensors Features	
Arithmetic mean	$\bar{s} = \frac{1}{N} \sum_{i=1}^N s_i$
Standard deviation	$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (s_i - \bar{s})^2}$
Frequency signal Skewness	$\mathbb{E} \left[\frac{(s - \bar{s})^3}{\sigma} \right]$
Frequency signal Kurtosis	$\mathbb{E}[(s - \bar{s})^4] / \mathbb{E}[(s - \bar{s})^2]^2$
Signal magnitude area	$\frac{1}{3} \sum_{i=1}^3 \sum_{j=1}^N s_{i,j} $
Pearson Correlation coefficient	$C_{1,2} / \sqrt{C_{1,1} C_{2,2}}, C = \text{cov}(s_1, s_2)$
Spectral energy of a frequency band [a, b]	$\frac{1}{a-b+1} \sum_{i=a}^b s_i^2$
s: signal vector Q: quartile, N: signal vector length, cov: covariance	
GPS Features	
Significant changes from the prior location state	
Estimated speed	
Variations in latitude and longitude	
Phone State Features	
Is smartphone screen unlocked?	Is smartphone charging?
Is ringer setting set to silent?	Is smartphone connected to WIFI?

CNN's auto-learning capabilities employ raw sensor data from three axial sensors (accelerometer, gyroscope, and magnetometer). The CNN we utilized, which was adapted

from the DeepContext [4], has a soft attention mechanism that aids in the learning of prominent features by assigning greater priority to those parts of the raw sensor data that are more indicative of the user's context. The intuition behind the design of this attention mechanism is similar to that proposed by Refs. [4,30]. The effectiveness of this architecture comes from using attention layers on features generated by single-sensor CNNs and features generated by CNNs that assessed the combined sensor outputs. This enables the model to emphasize CNN characteristics that are context-specific. For more details about the DeepContext CNN architecture, we refer the reader to Ref. [4].

4.4. Domain Alignment Loss

The objective of domain alignment loss is to transform the source and target feature encoding into a common feature distribution space in order to discover feature representations that are shared across domains. Gretton et al. [31] presented Multi Kernel Maximum Discrepancy Mean (MK-MMD) as an improvement for Maximum Discrepancy Mean (MMD), which we employ in our method. MMD is a non-parametric distance metric that can be employed to evaluate the disparity between marginal distributions [18]. MMD maps the feature representations of the source and target domains (\mathcal{X}_s and \mathcal{X}_t) to the Reproducing kernel Hilbert space (RKHS) and then computes the mean distance between the two distributions in RKHS. MK-MMD has been proposed as an optimal kernel selection approach for MMD because it can find an ideal kernel created by a weighted combination of various kernels based on the source and target datasets [18]. Let $\phi(\cdot)$ be a feature map defined as a combination of G positive kernels k_u with their associated bandwidth $\beta_u \geq 0$, given as the following:

$$k = \sum_{u=1}^G \beta_u k_u, \quad (3)$$

$$\phi(x^s, x^t) = k(x^s, x^t), \quad (4)$$

where x^s and x^t represent feature embeddings for the source and target domain, respectively. The formulation of MK-MMD is thus defined as follows:

$$q(\mathcal{X}^s, \mathcal{X}^t) = \|E_{\mathcal{X}^s}[\phi(x^s)] - E_{\mathcal{X}^t}[\phi(x^t)]\|_{H^k}, \quad (5)$$

where $\|\cdot\|_{H^k}$ is the RKHS norm. The domain alignment loss can be obtained by:

$$\mathcal{L}_d^\theta = \sum_{l \in N^l} q^2(\mathcal{X}_l^s, \mathcal{X}_l^t). \quad (6)$$

MK-MMD is computed per network layer to measure the distance between the source and target domain data representations. N^l indicates the number of layers, and we denote $(\mathcal{X}_s^l, \mathcal{X}_t^l)$ for the distributions of the source and target domains, retrieved from the l th layer in the network. $d(\mathcal{X}_s^l, \mathcal{X}_t^l)$ is the MK-MMD calculated by Equation (5) between the source and target domains distributions evaluated on the l th layer embeddings. Intuitively, the domain alignment loss is a regularizer that minimizes the distance between the distributions generating source domain data and target domain data.

4.5. Classification Loss

The objective of classification loss is to use source domain labels to discover discriminative features for context predictions. Both domains utilize the same context labels for classification. The overall learning process is guided by the optimization of our model for context classification on the source domain. Given the availability of D_s 's labels (labeled source domain data), the classification loss is defined as:

$$\mathcal{L}_{cls}^\theta = \frac{1}{N_s} \sum_{i=1}^{N_s} \ell_\Psi(f_\phi(x_i^s), y_i^s), \quad (7)$$

where the classifier is denoted as $f_\phi(\cdot)$, N_s represents the number of labeled training samples, ℓ_ψ is a binary cross-entropy function with inverse class frequency weighting that corrects for class imbalance, and $(x^s), y^s$ represents labeled context data sampled from source domain data.

4.6. Triplet Loss

In an embedding space, the triplet loss is primarily utilized to group samples from the same or related classes together and push samples associated to different classes away. An empirical success was seen in the field of face recognition. This is because different images of the same person map very closely to the learned embedding space [13,14]. Since numerous variations in the sensor inputs can represent the same context, we think the same approach can be applied to sensor data.

Given three samples, an anchor sample x_a (also called a query sample), a positive sample x_p (one that belongs to the same class as the anchor), and a negative sample x_n (i.e., a sample with a different class from the anchor), and with a distance function d , we can define triplet loss as follows:

$$\mathcal{L}_{\text{tri}}^\theta = \sum_i^N [d(x_a^i, x_p^i) - d(x_a^i, x_n^i) + \alpha]_+ \quad (8)$$

where α represents the margin between positive and negative samples and x represents an embedding of x for ease of notation. We reduce the triplet loss by pushing $d(x_a^i, x_p^i)$ towards zero and making $d(x_a^i, x_n^i)$ to be greater than $d(x_a^i, x_p^i) + \alpha$. In other words, pairs of positive samples are jointly grouped together, while positive and negative sample pairs are separated by some margin α . To put this in perspective, we want the network to learn a feature space in which the squared distance between all feature embeddings of the same context is minimal, while the squared distance across sensor contexts associated with different labels is large.

4.7. Joint-Fusion Triplet Mining

The process of constructing triplets (anchor, positive, and negative) for triplet loss calculations is known as triplet mining. The two main strategies for selecting triplets are offline and online. Finding triplets offline is not recommended as it requires a complete pass over the training set [14]. In accordance with the method described in Ref. [13], we employ an online triplet mining strategy that does not require a prior pass on the training set. Because discovering triplets across two domains necessitates the presence of target domain labels, one of the most prevalent solutions for UDA problems is to use the classifier trained on the source domain to generate pseudo labels for samples of the target domain during training [12]. During this procedure, it is vital to remember that the pseudo labels generated may not be accurate. Nonetheless, we reassign pseudo labels every few iterations since the accuracy of the classifier on the target dataset improves continuously throughout training. In addition, domain alignment loss can help improve the accuracy of the classifier on the target dataset by reducing distribution disparity. Consequently, the quality of the pseudo label can improve automatically.

Our joint-fusion triplet mining technique operates as follows: After concatenating two mini-batches of samples from the source and target domains into one mini-batch, triplets are generated. We need a concept of similarity between multi-labeled vectors in order to construct triplets that are compatible with our multi-labeling settings. First, we define a compatibility score between two binary labeled contexts y_1, y_2 as the dot product between them:

$$c(y_1, y_2) = y_1 \cdot y_2 \quad (9)$$

Due to the imbalanced nature of our dataset, we consider all the positive examples when constructing triplets. We use a strategy similar to Ref. [13] that focuses on triplets that

contribute the most to the learning process, but we modify it by using our compatibility score to select triplets that meet the following condition:

$$d(x_a, x_p) + \alpha > d(x_a, x_n) \ \& \ c(y_a, y_p) > c(y_a, y_n) \quad (10)$$

Our triplet mining strategy is detailed in Algorithm 1.

Algorithm 1: JOINT-FUSION ONLINE TRIPLET MINING finds triplets with multi-labeled vectors

Input: Number of samples in a batch m , classifier $f(\cdot)$, D_s , D_t , distance d , compatibility score (Equation (9)) c , α , and sample size k

Output: List of triplets $[(a, p, n)]$

```

{ $x_i^s, y_i^s$ } $_{i=1}^m$   $\leftarrow$  Read next mini-batch ( $D_s$ );
{ $x_i^t$ } $_{i=1}^m$   $\leftarrow$  Random sample mini-batch ( $D_t$ );
{ $y_i^t$ } $_{i=1}^m$   $\leftarrow$  Assign pseudo labels using  $f$  on { $x_i^t$ } $_{i=1}^m$ ;
{ $x_i^z, y_i^z$ } $_{i=1}^m$   $\leftarrow$  concatenate( { $x_i^s, y_i^s$ } $_{i=1}^m$ , { $x_i^t, y_i^t$ } $_{i=1}^m$ );
triplets  $\leftarrow$  {}
for each  $a, p$  in { $x_i^z, y_i^z$ } $_{i=1}^m$  do
  if  $a$  has positive labels And  $a \neq p$  then
    Q  $\leftarrow$   $k$  Random samples from { $x_i^z, y_i^z$ } $_{i=1}^m$ ;
    N  $\leftarrow$  {};
    for  $q$  in N do
      //Negative example candidate selection:
      if  $c(a, q) == 0 \ \& \ d(a, q) < d(a, p)$  then
        | N.add( $q$ );
      end
    end
    for  $n$  in N do
      Equation (10):
      if  $d(a, p) + \alpha > d(a, n) \ \& \ c(a, p) > c(a, n)$  then
        | triplets.add ( $(a, p, n)$ );
      end
    end
  end
end
end
return triplets

```

5. Experiments

We compared Triple-DARE and baseline models on both scripted and in-the-wild smartphone HCR datasets, where we performed multiple UDA use cases. Overall, *Triple-DARE* was used to obtain a robust representation from the scripted dataset (source), which was then applied to enhance HCR on the in-the-wild dataset (target).

5.1. Datasets

In-the-wild dataset: A total of 103 participants downloaded a smartphone app that passively collected data for 2 weeks as they went about their daily lives. Periodically, participants were asked to self-report the context labels that they visited. In addition to being more realistic, our in-the-wild dataset reflected a variety of manufacturer hardware because it was collected using individuals' smartphones.

Scripted dataset: The smartphone application collected information from 100 participants who visited predetermined locations. During the data collection session, which lasted about an hour per subject, human proctors oversaw and manually annotated the data. Both scripted and unstructured datasets were similarly preprocessed and characterized. The contexts were treated as vectors with multiple labels, with 10-s window size used to

create segments. The number of samples for scripted and unscripted datasets is 21,846 and 631,026, respectively. In Table 5, we list the context labels used throughout this study. To increase the applicability of our model to unseen subjects, subject-wise cross-validation was used, wherein a given subject’s data was included in either the training set or the test set, but never both. Each UDA experiment utilized 90% of source domain data for training, 10% of source domain data for validation, and 100% of target domain data for testing. Figure 4 displays data extracted from the two datasets, displaying only the accelerometer sensor readings for three context examples.

Table 5. The proportions of contexts that have been given a positive label.

Contexts	Scripted % P	In-the-Wild % P
Bathroom	3.15%	2.17%
Jogging	2.04%	0.27%
Lying Down	1.10%	16.24%
Running	1.95%	0.37%
Sitting	11.99%	38.71%
Sleeping	2.19%	37.69%
Stairs—Going Down	2.52%	2.00%
Stairs—Going Up	0.89%	1.92%
Standing	1.71%	8.46%
Talking On Phone	1.41%	1.27%
Typing	3.65%	6.45%
Walking	64.00%	13.51%
Phone Prioceptions		
Phone In Hand	Phone In Pocket	Phone In Bag
Datasets Notations		
$S_{Prioception}$	Scripted context dataset	
$W_{Prioception}$	In-the-wild context dataset	
e.g., S_{Pocket} refers to scripted contexts, annotated with “Phone In Pocket”		

5.2. Baselines

We compared *Triple-DARE* with cutting-edge deep-learning-based DA models: (1) *CORAL* [19]: A cutting-edge UDA model that uses deep-coral discrepancy loss. (2) *DAN* [18]: A model with just our MK-MMD alignment loss. (3) *HDCNN* [17]: An advanced baseline DA technique was previously applied to smartphone sensor data. *HD-CNN* cross-domain transfer learning model that uses KL divergence loss on the acquired feature vectors. (4) *SOURCE*: A model that has only been trained in the source domain, with no adjustments made for the target domain. Our proposed model, *Triple-DARE*, which employs our joint-fusion triplet loss.

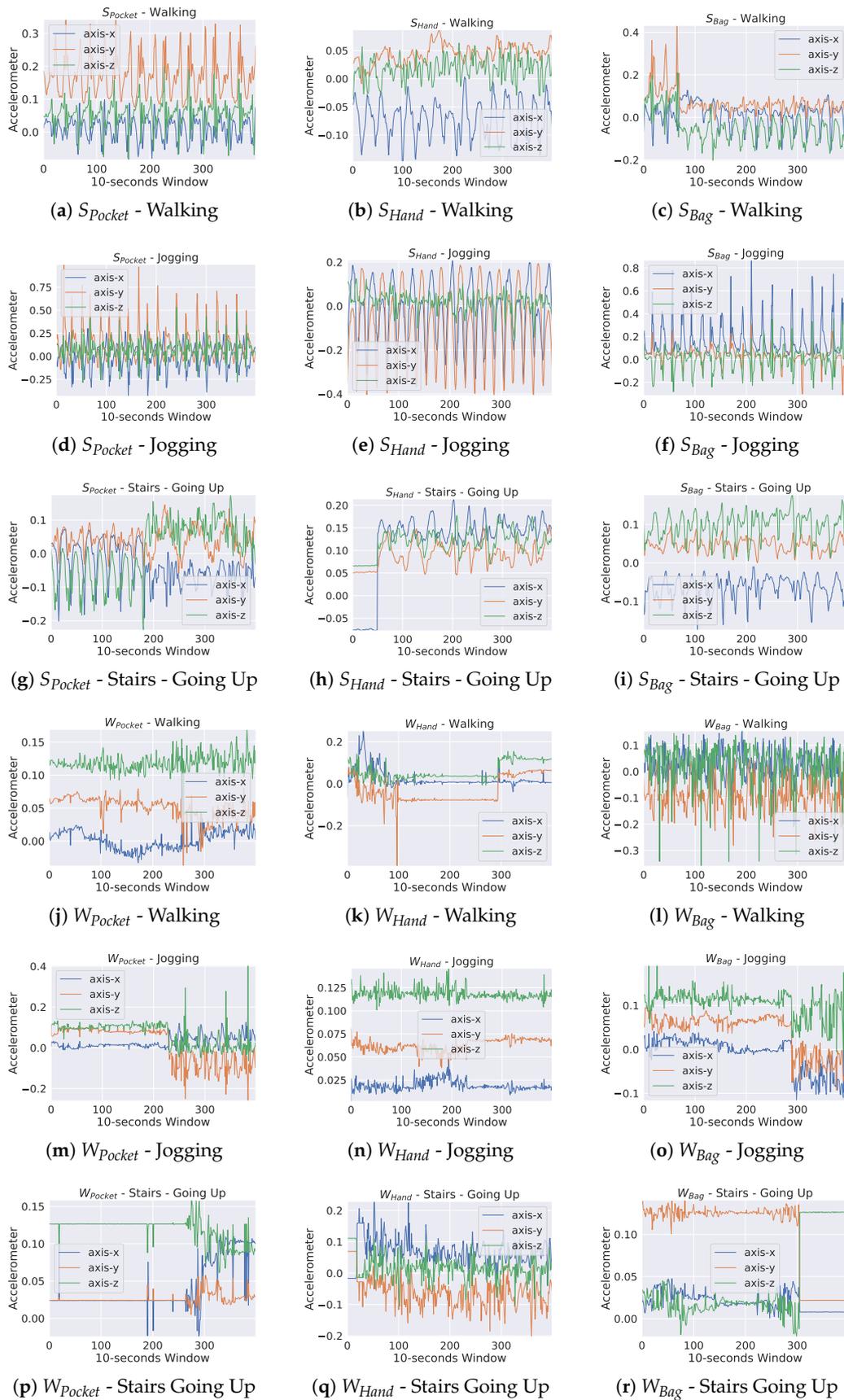


Figure 4. Raw accelerometer tracings sampled from Walking, Jogging, and Stairs Going up contexts within each dataset.

5.3. Implementation and Experimental Settings

(1) *Hyper-parameters*: Grid search was used to optimize the hyper-parameters of MLP and CNN. The learning rate is initialized at 1×10^{-1} , balancing coefficients are initialized as $\lambda_1 = 1$, $\lambda_2 = 0$, and $\lambda_3 = 0$. Following the schedule outlined in Ref. [25], the balance coefficients and the learning rate are raised or lowered, making our model more confident in source labels and less sensitive to low-quality pseudo labels during the early phases of training.

The batch size is set to 256. Adam optimizer was used. All trials use the same backbone layers utilized by our DA technique: Two-layer handcrafted-features MLP with 16 hidden dimensions, single-layer MLP domain classifier with 32 hidden dimensions, and a convolutional neural network with attention blocks for individual and combined sensor layers, then an average pooling layer, adopted from Ref. [4]. All sensor data was fed into a three-layer CNN. The outputs are then concatenated and sent to another 3-layer CNN. Attention blocks are utilized to concentrate on prominent regions of inputs [4,30]. In triplet mining, pairwise distances are computed using Euclidean distance and α is set to 0.1. Following the LeakyReLU activation in the final context prediction layer is the Sigmoid activation.

(2) *Evaluation Protocol*: In addition to reporting classification accuracy, we used the F1 metric to evaluate HCR performance in the UDA setting due to the class imbalance in our context datasets. As the sizes of the source and target domain datasets may not be identical, random sampling is used to iterate through the target domain dataset. However, we evaluate our model on every sample in the dataset for the target domain.

5.4. Results and Findings

(1) *Notations*: First, we define the notations used in our experimental results. $S_{Priocation}$ is denoted for the scripted context dataset and $W_{Priocation}$ for in-the-wild dataset, e.g., S_{Bag} refers to scripted contexts, annotated with “Phone In Bag”.

(2) *Overall Results*: In Table 6, we compare the overall performance scores of our *Triple-DARE* algorithm to those of the baseline models. In the overall UDA tasks and Lab-to-field UDA tasks, *Triple-DARE* outperforms the baseline methods with a 4.5% increase in F1-score and 6.3% increase in classification accuracy. Figure 5 displays the performance per context label across all UDA tasks, demonstrating that our approach outperforms state-of-the-art methods across multiple context labels. In general, UDA methods have an advantage over classifiers that are trained solely on the source domain without leveraging unlabeled data. Particularly, UDA methods were of great assistance with the Jogging, Running, Going Up and Down Stairs labels, for which the user is unlikely to provide labels while performing these activities in the wild. Nevertheless, our method makes use of the high-fidelity labels acquired during the scripted study and enhances adaptation. As shown in Table 5, predictions for the labels Sitting and Walking are the most difficult, which may be due to a significant difference in target label distributions.

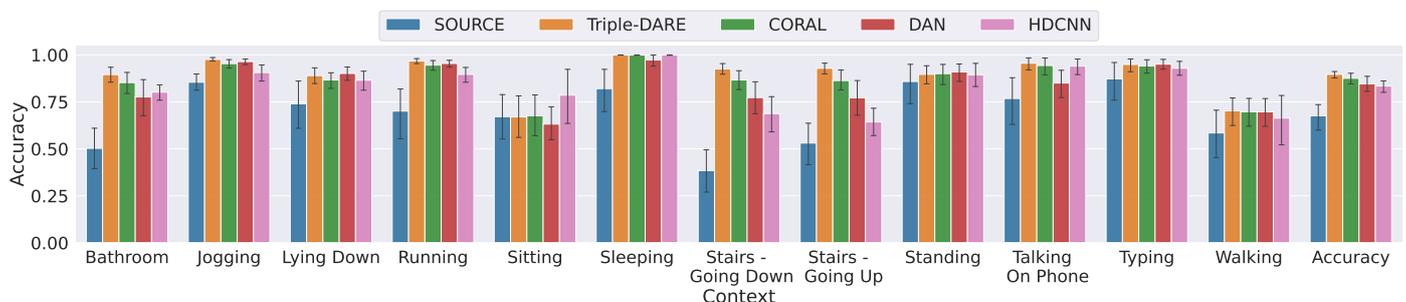


Figure 5. Target prediction scores for each label, averaged across various UDA task domains.

Table 6. Overall context prediction scores on the target domain.

Overall UDA Tasks	Accuracy	F1-micro
<i>Triple-DARE</i>	0.879	0.366
CORAL	0.806	0.302
DAN	0.673	0.294
HDCNN	0.816	0.3215
Source (no adaptation)	0.433	0.259
Lab-to-field UDA Tasks	Accuracy	F1-micro
<i>Triple-DARE</i>	0.845	0.188
CORAL	0.839	0.127
DAN	0.698	0.122
HDCNN	0.768	0.146
Source (no adaptation)	0.552	0.133

The highest scores are highlighted in bold font. Row colors match the colors used for different baseline methods throughout this article.

(3) *Scripted contexts with cross-prioception UDA tasks:* Figure 6 demonstrates that *Triple-DARE* consistently outperforms the baseline methods on all cross-prioception UDA tasks. The adaptation procedure has significantly benefited the UDA tasks with “Phone In Hand” as their target domain. This advantage is a result of the signal noise introduced when the phone is in motion. In the majority of instances, *CORAL* performs better than *DAN*.

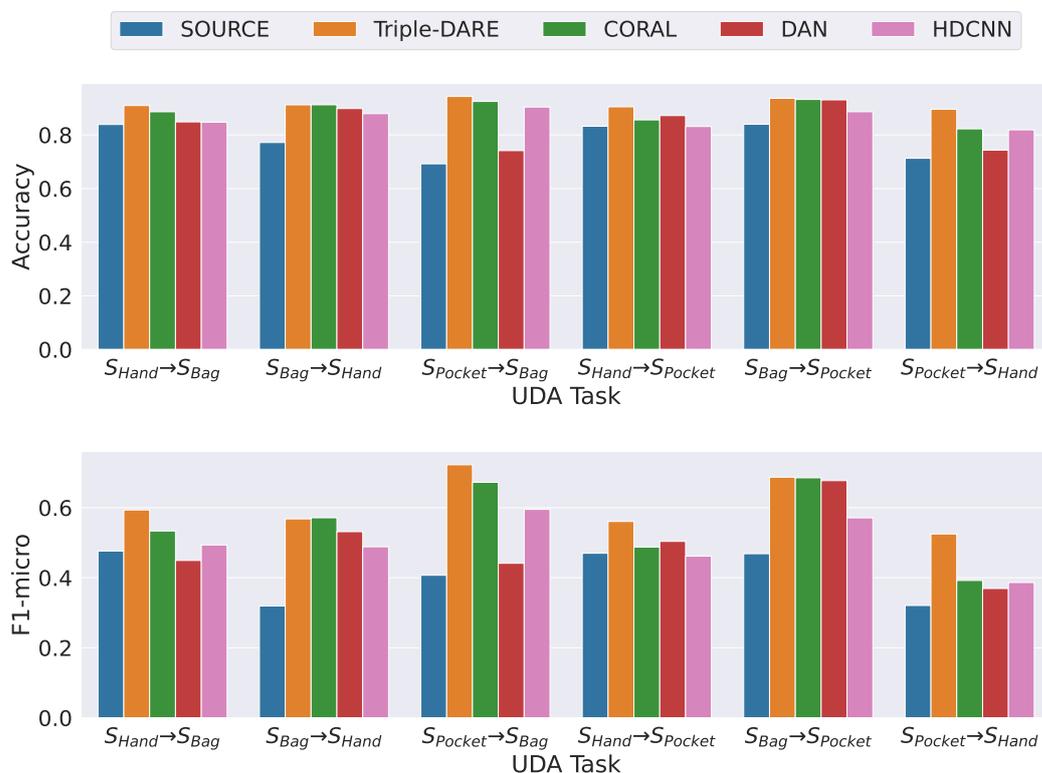


Figure 6. Scripted context data with cross-prioception UDA tasks.

(4) *Lab-to-field generalization UDA tasks:*

Figure 7 displays the results of our lab-to-field UDA generalization tasks. Massive differences in the scores obtained for “Phone in Pocket” versus “Phone in Bag” and “Hand” provide additional information about diversity placements. We hypothesize that when

the phone is placed in a bag or held in the hand, the model is unable to map data from scripted and in-the-wild datasets to a common feature space. In adapting models learned on scripted data to make context predictions on in-the-wild data with a “Phone in Pocket” prioception, however, we observe a notable improvement over baseline methods of the current state of the art.

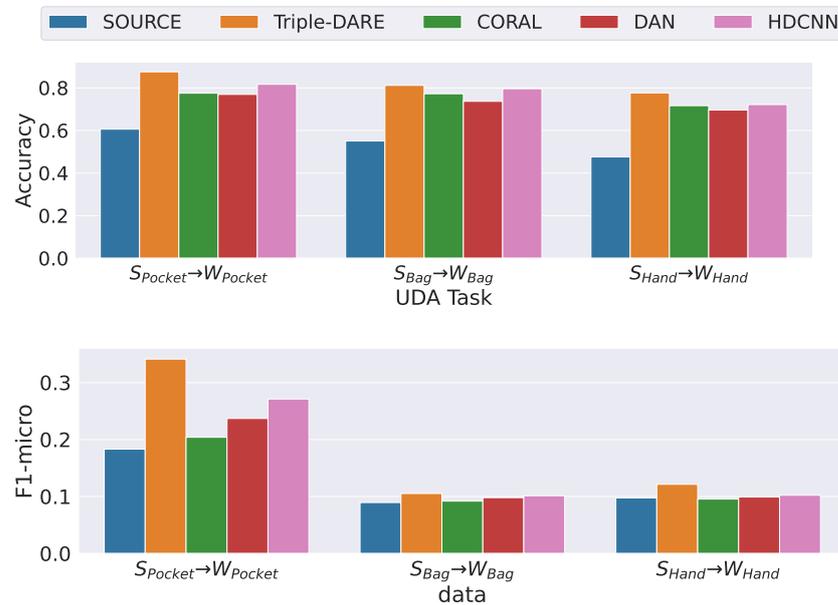


Figure 7. Scripted context to In-The Wild UDA tasks scores.

(5) *Training under insufficient labels:*

As shown in Table 7, we analyzed the performance of our model as a function of the number of labels in the source domain. We investigated how the number of labels in the source domain affected the performance of our model. In Figure 8, we plot the prediction scores obtained across multiple scripted cross-prioception domains, averaged across various source domains. The shaded region in this figure represents the amount of variance obtained when utilizing various source domains. Small regions of shading indicate that the scores are highly consistent across experiments. We observed a substantial difference when the target is “Phone in Bag” versus “Phone in Hand” or “Phone in Pocket”. Table 7 provides a more detailed version of this experiment. *Triple-DARE* attains superior prediction scores on the target domain using a small number of source labels, outdoing baseline methods in nearly all UDA tasks.

(6) *Intra-class compactness and inter-class separation:* To quantify the compactness and separation of learned feature embeddings, we employed the Silhouettescore Score $= \frac{b_i - a_i}{\max(b_i, a_i)}$, where b_i is the shortest average distance between a point and every other point in any cluster, whereas a_i represents the average distance between i and all data points belonging to the same cluster. This score accounts for both compactness and separation. To compute the Silhouette scores for the learned feature embeddings, we assign each instance one of the binary context labels as a cluster label. Then, we calculate the mean score across labels. In most UDA tasks, our *Triple-DARE* method achieves higher compactness and separation scores, as shown in Figure 9. In addition, in the majority of instances, CORAL achieves higher scores than DAN in most cases. Additionally, the quality of the learned feature embeddings can be viewed visually in Figure 10, which depicts the same context instances represented by feature embeddings learned using DAN and *Triple-DARE*. The visualization is obtained by projecting feature embeddings into a two-dimensional space using the T-distributed Stochastic Neighbor Embedding (TSNE) [32].

(7) *Ablation Study:* We conducted an experimental ablation (shown in Figure 11) to rank the utility of *Triple-DARE* for a variety of UDA tasks. The best results were seen when using

all its parts together. To further understand the relative impact of each component in this ablation investigation, we employed a non-pretrained HCR model. While the triplet loss and the domain loss are both useful, they do not provide as much insight as joint training.

Table 7. F-1 scores—Using variable amounts of source labels to compare diverse UDA methods for a variety of tasks.

Training %	Method	Scripted Contexts with Cross-Priception UDA Tasks							Lab-to-Field UDA Tasks			
		$S_{Bag} \rightarrow S_{Hand}$	$S_{Bag} \rightarrow S_{Pocket}$	$S_{Hand} \rightarrow S_{Bag}$	$S_{Hand} \rightarrow S_{Pocket}$	$S_{Pocket} \rightarrow S_{Bag}$	$S_{Pocket} \rightarrow S_{Hand}$	Avg	$S_{Bag} \rightarrow W_{Bag}$	$S_{Hand} \rightarrow W_{Hand}$	$S_{Pocket} \rightarrow W_{Pocket}$	Avg
		0.2	Triple-DARE	0.500	0.651	0.213	0.318	0.652	0.467	0.467	0.101	0.080
	CORAL	0.357	0.328	0.357	0.428	0.352	0.378	0.367	0.089	0.087	0.150	0.109
	DAN	0.341	0.436	0.285	0.265	0.418	0.403	0.358	0.079	0.077	0.165	0.107
	HDCNN	0.341	0.492	0.472	0.470	0.468	0.380	0.437	0.087	0.084	0.181	0.117
0.4	Triple-DARE	0.557	0.617	0.444	0.492	0.767	0.511	0.565	0.118	0.143	0.359	0.207
	CORAL	0.380	0.584	0.455	0.457	0.633	0.484	0.499	0.092	0.075	0.165	0.111
	DAN	0.452	0.509	0.418	0.451	0.721	0.459	0.502	0.101	0.093	0.244	0.146
	HDCNN	0.424	0.580	0.504	0.558	0.704	0.441	0.535	0.106	0.108	0.266	0.160
0.6	Triple-DARE	0.497	0.588	0.570	0.653	0.744	0.542	0.599	0.111	0.112	0.341	0.188
	CORAL	0.577	0.688	0.505	0.505	0.754	0.448	0.580	0.110	0.123	0.210	0.148
	DAN	0.540	0.634	0.428	0.459	0.653	0.429	0.524	0.100	0.084	0.209	0.127
	HDCNN	0.345	0.561	0.575	0.540	0.645	0.445	0.518	0.094	0.102	0.285	0.160
Average	Triple-DARE	0.518	0.619	0.409	0.488	0.721	0.507	0.544	0.111	0.112	0.341	0.188
	CORAL	0.438	0.533	0.439	0.463	0.580	0.437	0.482	0.097	0.093	0.173	0.122
	DAN	0.440	0.526	0.377	0.392	0.597	0.430	0.461	0.096	0.087	0.198	0.127
	HDCNN	0.370	0.544	0.517	0.523	0.606	0.422	0.497	0.096	0.098	0.244	0.146
-	No Adaptation	0.319	0.469	0.476	0.470	0.260	0.315	0.385	0.108	0.110	0.180	0.133

The highest scores are highlighted in bold font. Row colors match the colors used for different baseline methods throughout this article.

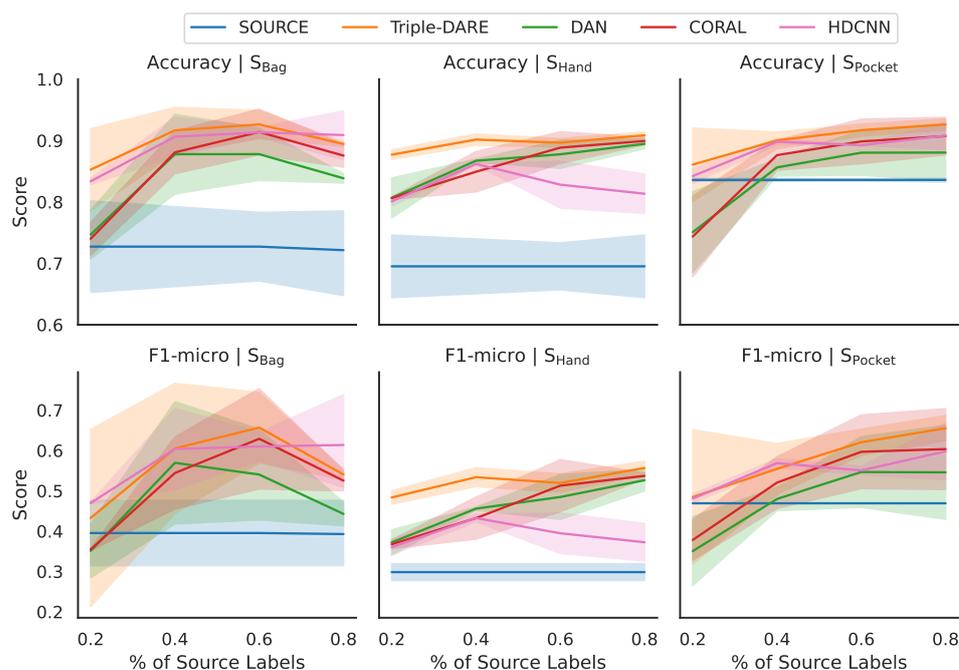


Figure 8. Scores for each source domain in scripted contexts with cross-priception UDA tasks, averaged over each target, varying the number of labels from the source domain.

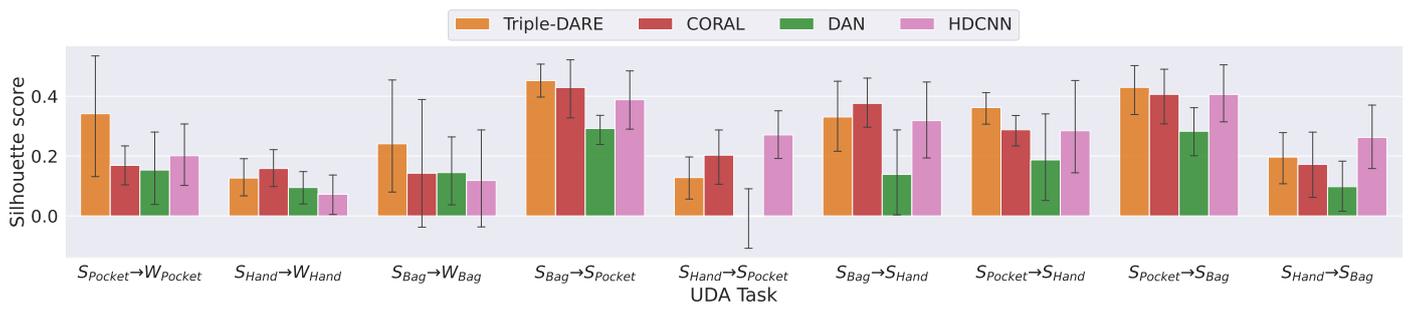


Figure 9. Compactness measure on feature embeddings.

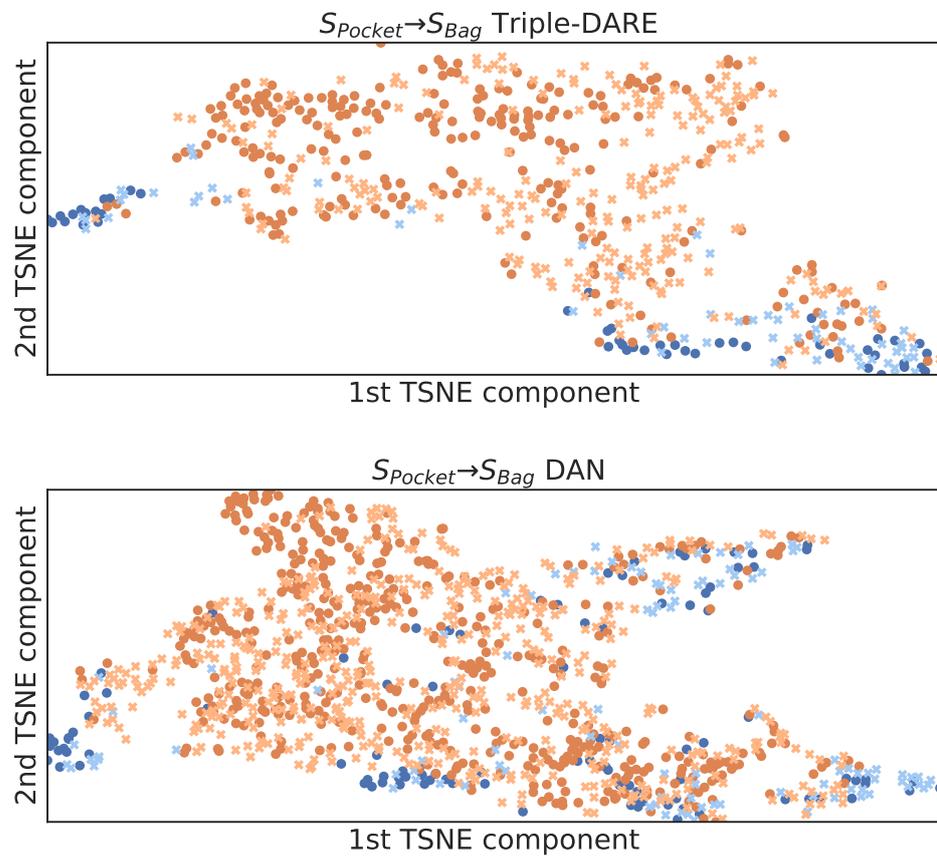


Figure 10. Visualization of the learned feature embeddings for TripleDARE (top) and DAN (bottom), using TSNE dimensional reduction.

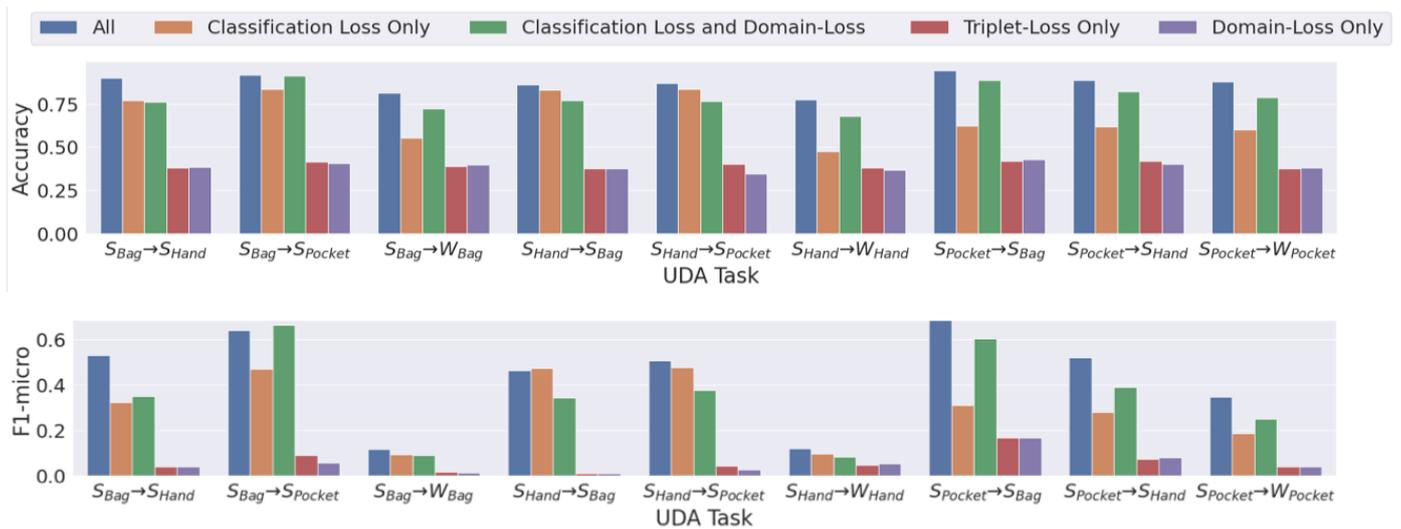


Figure 11. Ablation study, evaluating the contribution of *Triple-DARE*'s each component.

6. Limitations and Future Work

The assumption that the same number of sensors are available in scripted datasets and in in-wild datasets is one limitation of our methodology. We plan to investigate the possibility of using algorithms for lab-to-field recognition that utilize just a small fraction of sensors that are comparable in both domains. Increasing the model's resilience against missing sensors during inference is one way that our methodology might be improved. We hope that future studies in visual analytics will make use of our proposed strategy for representation learning for smartphone sensor data and the use of UDA for visualization.

7. Conclusions

The performance of machine learning HCR models on real-world datasets is hindered by diverse phone placements and smartphone models, as well as weak, noisy, or missing labels. The goal of lab-to-field methods is to improve the performance of HCR models by first training them on scripted HCR datasets and then modifying them so that they can be used for predicting context labels in comparable datasets that were collected in the wild. This is the first work we are aware of that uses lab-to-field techniques on HCR datasets collected from smartphones. This paper presents *Triple-DARE*, a UDA deep-learning model for HCR on smartphones, which is comprised of three components: (1) a domain alignment loss that utilizes MK-MMD (2) a classification loss and (3) a joint-fusion triplet loss particularly designed for multi-labeled datasets. *Triple-DARE* learns domain-invariant features common to both datasets, decreasing the influence of noisy in-the-wild data by concentrating on salient areas in sensor inputs, and achieving a high F1-score for multiple UDA tasks on both scripted and in-the-wild context datasets. With its domain alignment loss, *Triple-DARE* outperforms state-of-the-art baseline approaches when it comes to mapping the source and target feature embedding into a standard feature distribution. In addition, the triplet loss improves discrimination by increasing intra-class compactness and inter-class separation while utilizing enormous amounts of unlabeled data. *Triple-DARE* outperforms other state-of-the-art DA baselines, increasing the F1-score and classification accuracy by 4.6% and 1.89%, respectively, and outperforming models with no adaptations by 10.7% and 14.7%.

Author Contributions: Conceptualization, E.A. and A.A.; Data curation, A.A., W.G., L.B., K.C., H.M.; Formal analysis, W.G.; Funding acquisition, E.A. and E.R.; Methodology, A.A.; Project administration, E.A.; Software, A.A.; Supervision, E.A. and E.R.; Validation, A.A.; Visualization, A.A.; Writing—original draft, A.A.; Writing—review and editing, A.A. and E.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Computer Science Department at Worcester Polytechnic Institute and the DARPA WASH under Grant HR00111780032-WASH-FP-031 and in part by DARPA under Agreement FA8750-18-2-0077. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Helsinki, and approved by the Institutional Review Board of WPI (WPI IRB File 18-0232 “Warfighter Analytics using Smartphones for Health (WASH)” on 28 February 2018).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy agreements.

Acknowledgments: In this section you can acknowledge any support given which is not covered by the author contribution or funding sections. This may include administrative and technical support, or donations in kind (e.g., materials used for experiments).

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

MDPI	Multidisciplinary Digital Publishing Institute
HCR	Human Context Recognition
CA	Context Aware
DA	Domain Adaptation
UDA	Unsupervised Domain Adaptation
CNN	Convolutional neural network
MLP	Multilayer perceptron
WASH	Warfighter Analytics using Smartphone for Healthcare
UMMS	University of Massachusetts Medical School
WSL	Weakly Supervised Learning
ECCG	Wearable electrocardiogram
RKHS	Reproducing Kernel Hilbert Space
KL	Kullback-Leibler
STL	Stratified Transfer Learning
MMD	Maximum Discrepancy Mean
MK-MMD	Multi Kernel Maximum Discrepancy Mean

References

- Alajaji, A.; Gerych, W.; Buquicchio, L.; Chandrasekaran, K.; Mansoor, H.; Agu, E.; Rundensteiner, E.A. Smartphone Health Biomarkers: Positive Unlabeled Learning of In-the-Wild Contexts. *IEEE Pervasive Comput.* **2021**, *20*, 50–61. [[CrossRef](#)]
- Roggen, D.; Calatroni, A.; Rossi, M.; Holleczeck, T.; Förster, K.; Tröster, G.; Lukowicz, P.; Bannach, D.; Pirkl, G.; Ferscha, A.; et al. Collecting complex activity datasets in highly rich networked sensor environments. In Proceedings of the 2010 Seventh International Conference on Networked Sensing Systems (INSS) Kassel, Germany, 15–18 June 2010; pp. 233–240.
- Vaizman, Y.; Ellis, K.; Lanckriet, G. Recognizing detailed human context in the wild from smartphones and smartwatches. *IEEE Pervasive Comput.* **2017**, *16*, 62–74. [[CrossRef](#)]
- Alajaji, A.; Gerych, W.; Chandrasekaran, K.; Buquicchio, L.; Agu, E.; Rundensteiner, E. DeepContext: Parameterized Compatibility-Based Attention CNN for Human Context Recognition. In Proceedings of the 2020 IEEE 14th International Conference on Semantic Computing (ICSC), San Diego, CA, USA, 3–5 February 2020; pp. 53–60.
- Chang, Y.; Mathur, A.; Isopoussu, A.; Song, J.; Kawsar, F. A Systematic Study of Unsupervised Domain Adaptation for Robust Human-Activity Recognition. *ACM IMWUT* **2020**, *4*, 1–30. [[CrossRef](#)]
- Stisen, A.; Blunck, H.; Bhattacharya, S.; Prentow, T.; Kjærgaard, M.; Dey, A.; Sonne, T.; Jensen, M. 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, Seoul, Republic of Korea, 1–4 November 2015; pp. 127–140. [[CrossRef](#)]

7. Zhou, Z.H. A brief introduction to weakly supervised learning. *Natl. Sci. Rev.* **2018**, *5*, 44–53. [[CrossRef](#)]
8. Mansoor, H.; Gerych, W.; Buquicchio, L.; Chandrasekaran, K.; Agu, E.; Rundensteiner, E. DELFI: Mislabeled Human Context Detection Using Multi-Feature Similarity Linking. In Proceedings of the 2019 IEEE VDS, Vancouver, BC, Canada, 20 October 2019; pp. 11–19.
9. Natarajan, A.; Angarita, G.; Gaiser, E.; Malison, R.; Ganesan, D.; Marlin, B.M. Domain adaptation methods for improving lab-to-field generalization of cocaine detection using wearable ECG. In Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Heidelberg, Germany, 12–16 September 2016; ACM: Heidelberg, Germany, 2016; pp. 875–885. [[CrossRef](#)]
10. A unifying view on dataset shift in classification. *Pattern Recognit.* **2012**, *45*, 521–530.
11. Kouw, W.M. An introduction to domain adaptation and transfer learning. *arXiv* **2018**, arXiv:1812.11806.
12. Deng, W.; Zheng, L.; Jiao, J. Domain Alignment with Triplets. *arXiv* **2018**, arXiv:1812.00893.
13. Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A unified embedding for face recognition and clustering. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 815–823.
14. Hermans, A.; Beyer, L.; Leibe, B. In Defense of the Triplet Loss for Person Re-Identification. *arXiv* **2017**, arXiv:1703.07737.
15. Khaertdinov, B.; Ghaleb, E.; Asteriadis, S. Deep Triplet Networks with Attention for Sensor-based Human Activity Recognition. In Proceedings of the 2021 IEEE International Conference on Pervasive Computing and Communications (PerCom), Kassel, Germany, 22–26 March 2021; pp. 1–10.
16. Hachiya, H.; Sugiyama, M.; Ueda, N. Importance-weighted least-squares probabilistic classifier for covariate shift adaptation with application to human activity recognition. *Neurocomputing* **2012**, *80*, 93–101. [[CrossRef](#)]
17. Khan, M.A.A.H.; Roy, N.; Misra, A. Scaling Human Activity Recognition via Deep Learning-based Domain Adaptation. In Proceedings of the 2018 IEEE International Conference on Pervasive Computing and Communications (PerCom), Athens, Greece, 19–23 March 2018; pp. 1–9. [[CrossRef](#)]
18. Long, M.; Cao, Y.; Wang, J.; Jordan, M.I. Learning Transferable Features with Deep Adaptation Networks. *arXiv* **2015**, arXiv:1502.02791.
19. Sun, B.; Saenko, K. Deep CORAL: Correlation Alignment for Deep Domain Adaptation. In Proceedings of the ECCV Workshops, Amsterdam, The Netherlands, 8–10 and 15–16 October 2016.
20. Wilson, G.; Doppa, J.R.; Cook, D.J. Multi-Source Deep Domain Adaptation with Weak Supervision for Time-Series Sensor Data. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, CA, USA, 23–27 August 2020; ACM: Virtual Event, CA, USA, 2020; pp. 1768–1778. [[CrossRef](#)]
21. Shimodaira, H. Improving predictive inference under covariate shift by weighting the log-likelihood function. *J. Stat. Plan. Inference* **2000**, *90*, 227–244. [[CrossRef](#)]
22. DARPA. DARPA WASH BAA. Available online: <https://beta.sam.gov/opp/cfb9742c60d055931003e6386d98c044/view> (accessed on 7 June 2020).
23. Alajaji, A.; Gerych, W.; Chandrasekaran, K.; Buquicchio, L.; Mansoor, H.; Agu, E.; Rundensteiner, E. Triplet-based Domain Adaptation (Triple-DARE) for Lab-to-field Human Context Recognition. In Proceedings of the 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), Pisa, Italy, 21–25 March 2022; pp. 155–161.
24. Byrd, J.; Lipton, Z.C. What is the Effect of Importance Weighting in Deep Learning? In Proceedings of the ICML, Long Beach, CA, USA, 9–15 June 2019.
25. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V. Domain-Adversarial Training of Neural Networks. *J. Mach. Learn. Res.* **2016**, *17*, 1–35.
26. Chen, Y.; Wang, J.; Huang, M.; Yu, H. Cross-position activity recognition with stratified transfer learning. *Pervasive Mob. Comput.* **2019**, *57*, 1–13. [[CrossRef](#)]
27. Sanabria, A.R.; Ye, J. Unsupervised domain adaptation for activity recognition across heterogeneous datasets. *Pervasive Mob. Comput.* **2020**, *64*, 101147. [[CrossRef](#)]
28. Vaizman, Y.; Ellis, K.; Lanckriet, G.; Weibel, N. Extrasensory app: Data collection in-the-wild with rich user interface to self-report behavior. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, Montreal, QC, Canada, 21–27 April 2018; pp. 1–12.
29. Reyes-Ortiz, J.L.; Oneto, L.; Samà, A.; Parra, X.; Anguita, D. Transition-Aware Human Activity Recognition Using Smartphones. *Neurocomputing* **2016**, *171*, 754–767. [[CrossRef](#)]
30. Jetley, S.; Lord, N.A.; Lee, N.; Torr, P.H.S. Learn To Pay Attention. *arXiv* **2018**, arXiv:1804.02391.
31. Gretton, A.; Sriperumbudur, B.K.; Sejdinovic, D.; Strathmann, H.; Balakrishnan, S.; Pontil, M.; Fukumizu, K. Optimal kernel choice for large-scale two-sample tests. In Proceedings of the Conference on Neural Information Processing Systems (NIPS), Lake Tahoe, NV, USA, 3–6 December 2012.
32. Maaten, L.V.D.; Hinton, G.E. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.