Robust Representation Learning for Context Recognition on Weakly-supervised Mobile Sensed Data with Covariate-shifts

by

Abdulaziz S. Alajaji

A Dissertation
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Doctor of Philosophy
in
Data Science
April 2023

APPROVED:

Prof. Emmanuel Agu, Major Advisor, CS Department & Data Science Program, WPI

Prof. Elke Rundensteiner, Co-Advisor, CS Department & Data Science Program, WPI

Prof. Nima Kordzadeh, Dissertation Committee Member, The Business School, WPI

Dr. Akhil Mathur, Dissertation Committee Member, Meta AI

Abstract

Context-aware applications adapt their behavior based on the user's current situation, targeting diverse domains, including smart homes, assisted living, fitness tracking, military deployment, and mobile health. Human Context Recognition (HCR), the task of detecting a user's current situation that includes their activity, location, and other semantic information, is a fundamental problem in context-aware applications. Smartphone HCR datasets for supervised machine learning are gathered using one of two study designs: 1) Scripted studies in which users visit pre-planned contexts in a script yield high-quality data with strong context labels but are unrealistic. 2) In-the-wild, unscripted studies gathered as subjects live their lives and provide context labels periodically yield realistic datasets but are frequently imbalanced with missing or wrong labels. Moreover, in-the-wild study designs are more vulnerable to attacks. For instance, adversaries can send modified data samples to mislead the HCR model, causing wrong predictions.

This dissertation presents HCR research that utilizes neural networks for HCR representation learning to facilitate robust HCR on single datasets, enhance HCR robustness to distributional shifts between multiple HCR datasets and mitigate perturbations maliciously caused by adversaries. DeepContext, a novel proposed neural network for HCR, utilizes joint learning with a parameterized compatibility-based attention mechanism to focus on the most predictive parts of sensor data. Two lab-to-field methods are proposed that learn a robust HCR model from a scripted dataset with strong labels to improve performance on a weakly supervised, in-the-wild dataset with similar context labels. Covariate shifts between the scripted and in-the-wild context datasets present a challenge to such lab-to-field methods. Positive Unlabeled Context Learning (PUCL) uses transductive learning with inaccurate supervision to address erroneous labels. Triple-DARE uses a Domain Adaptation approach under incomplete supervision to utilize unlabeled, in-the-wild data. Finally, an adversarial HCR approach that learns a robust representation is proposed. This dissertation focuses on black-box evasion attacks that can generate input samples with minor changes that result in high-confidence misclassifications. We propose RobustHCR, which uses a duality-based method to improve neural network robustness, allowing it to be provably resilient to norm-bounded perturbations. Results generated using a rigorous experimental plan are presented.

I have a lot to be grateful for. First and foremost, I praise God. I am eternally thankful for his greatness and for giving me the strength and courage that helped me overcome all the obstacles while completing this dissertation. My father instilled in me the tenacity to keep trying no matter how difficult things may become. My family's support, mainly my brother Abdulmageed, who has been an excellent role model, and my mother's unconditional love have kept me going. I am grateful for my wife, Arwa, who has been immensely supportive and selfless. I am so grateful for my wonderful kids, Dana and Saoud. Their laughter and smiles bring me so much joy and happiness that is truly priceless.

Acknowledgement

Throughout the writing of this dissertation, I received a great deal of assistance and encouragement. First of all, I like to thank my supervisor, Professor Emmanuel Agu, whose knowledge and experience were crucial in developing the research questions and methods. His insightful feedback encouraged me to improve my thoughts and raise the quality of my work. I also would like to thank my co-advisor, Professor Elke Rundensteiner, whose expertise was invaluable in improving the overall quality I produced in my research. Thank you for your valuable feedback and patient support throughout this research. Thanks to Prof. Nima Kordzadeh and Dr. Akhil Mahur for your help and support as dissertation committee members. I appreciate the time and effort you put into evaluating my work and providing insightful feedback. Moreover, I would like to thank my DARPA WASH teammates (Luke Buquicchio, Kavin Chandrasekaran, Walter Gerych, and Hamid Mansoor) for their valuable comments and continuous support. The time and effort that our team put into the WASH project data collection studies have helped in answering the research questions and methods proposed in this dissertation.

I want to give special thanks to my wife, Arwa Alhajeri, and my family for their continuous support and understanding while I was conducting my research and writing my papers and this dissertation. Their prayers for me sustained me this far. Also, Special appreciation goes to Drs. Walter Goula, Mariah Liberty, and John Job, who assisted me greatly with my chronic pain. Finally, I could not have completed this dissertation without the support of my friends, who provided stimulating discussions and happy distractions to rest my mind outside of my research.

This material is based on research sponsored by DARPA under agreement number 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.

Contents

List of Figures

List of Tables

Papers Contributing to this Dissertation

- Abdulaziz Alajaji, Walter Gerych, Kavin Chandrasekaran, Luke Buquicchio, Emmanuel Agu, Elke A. Rundensteiner. DeepContext: Parameterized Compatibility-Based Attention CNN for Human Context Recognition. IEEE International Conference on Semantic Computing (ICSC) 2020: 53-60
- Abdulaziz Alajaji, Walter Gerych, Luke Buquicchio, Kavin Chandrasekaran, Hamid Mansoor, Emmanuel Agu, Elke A. Rundensteiner. Smartphone Health Biomarkers: Positive Unlabeled Learning of In-the-Wild Contexts. IEEE Pervasive Computing 20(1): 50-61 (2021)
- Abdulaziz Alajaji, Walter Gerych, Luke Buquicchio, Kavin Chandrasekaran, Hamid Mansoor, Emmanuel Agu, Elke A. Rundensteiner. Triplet-based Domain Adaptation (Triple-DARE) for Lab-to-field Human Context Recognition. Accepted at The 20th International Conference on Pervasive Computing and Communications (PerCom 2022) - Industry Track.
- Abdulaziz Alajaji, Walter Gerych, Luke Buquicchio, Kavin Chandrasekaran, Hamid Mansoor, Emmanuel Agu, and Elke Rundensteiner. Domain Adaptation Methods for Lab-to-Field Human Context Recognition. Sensors 23, no. 6 (2023): 3081.
- Abdulaziz Alajaji, Walter Gerych, Luke Buquicchio, Kavin Chandrasekaran, Emmanuel Agu, and Elke Rundensteiner. Adversarial Human Context Recognition: Evasion Attacks and Defenses. Accepted at the IEEE Computer Society Signature Conference on Computers, Software, and Applications (COMPSAC) 2023.

Part I

Dissertation Introduction

Chapter 1

Introduction

1.1 Motivation

1.1.1 Context-Aware Systems

Context is defined as any information that can be used to characterize the situation of users during their interactions with computer applications. Context information is typically utilized to provide task-relevant information and/or services to users [1]. Context-Aware (CA) systems adapt their behavior to the user's current context. Context awareness is crucial in enabling ubiquitous computing systems to optimize usability, aligning with Mark Weiser's vision of ubiquitous computing [2]. In this dissertation, we focus on recognizing human behavioral contexts from smartphone sensor data to support CA applications, a task referred to as "*Human Context Recognition*" (HCR). A fundamental assumption in such a system is that sensor data exhibit similar patterns in similar contexts [3]. CA applications target various domains, including smart homes [4], assisted living [5], fitness tracking [6], military deployment [7], and mobile health [8, 6, 9, 10].

1.1.2 Human Context Recognition (HCR) for Context-Aware (CA) Smartphone Healthcare applications

Traditionally, health monitoring and lifestyle treatments have relied on manual, subjective reporting [11], sometimes supplemented by end-of-day recalling [12]. As a result, patient evaluations are infrequent, commonly months apart, and frequently result in late diagnoses that exacerbate patients' prognoses, causing tremendous socioeconomic damage to a community. Many patients receive very little care between scheduled hospital appointments. For example, it is anticipated that under-resourcing mental health will cost the global economy $16 trillion between 2010 and 2030 due to the early onset of mental illnesses and accompanying loss of productivity[13]. Due to the enormous societal costs, governments must focus on early, preventive interventions. Given that, accurate HCR can facilitate passive context-specific patient assessments and continuous monitoring, decreasing operational costs [10]. Specifically, automatic context recognition will aid in detecting critical ailment periods and facilitate the provision of prompt treatment[14].

Smartphones have recently become popular for CA applications as they are ubiquitous and often equipped with a plethora of built-in sensors. According to recent studies, 85% of people in the U.S. own a smartphone as of Feb 2022, with a steadily increasing number each year, even worldwide [15, 16]. We focus on CA and HCR on smartphones that leverage passive smartphone sensing and facilitate passive context-specific patient assessments and continuous monitoring, decreasing high operational costs in traditional infrequent health assessments.

1.1.3 HCR for Context-Aware Warfighter Health

The DARPA-funded Warfighter Analytics using Smartphone for Healthcare (WASH) DARPA project [17] is investigating passive smartphone assessment of Traumatic Brain Injuries (TBI) and infectious diseases. This will provide an up-to-date assessment of the warfighter's battle readiness. Target populations initially include active duty service members and veterans, but the scientific discoveries made will also apply to civilians. In the envisioned use case, the WASH smartphone app will continuously gather smartphone sensor data passively throughout each day. Each subject's full day of smartphone data will be pushed to the cloud, where disease inference models will analyze this data to generate a bioscore (or probability of illness) for each warfighter[17].

1.2 Definition of Context

While there are many definitions of context in the literature, in this dissertation, we define Human Context as the ⟨Activity, Prioception⟩ tuple, which consists of the user's current activity (e.g., walking, running) and the phone prioception (placement) (e.g., in a pocket, hand, or bag). As human context frequently includes their current activity, it is essential to clarify that the Human Activity Recognition (HAR) tasks, which involve recognizing the user's current activity, are related to Human Context Recognition (HCR) that this dissertation focuses on.

1.3 Scripted vs. In-the-wild Context Recognition Data Gathering Studies

Most existing datasets for HCR were gathered in human subjects studies that were either *scripted* or *in-the-wild*. In scripted studies, participants perform tasks in a pre-planned order under the supervision of a human proctor while a smartphone app continuously records smartphone sensor readings. Afterward, the human proctors annotate users' sensor data with labels of the contexts they visited. In contrast, *in-the-wild* studies involve collecting data for several days in the real world as subjects live their everyday lives. A smartphone app continuously gathers sensor data and periodically prompts the smartphone owner to report their current context, which is then used to annotate their sensor data. Scripted datasets have accurate context labels, but user behaviors are not realistic. While realistic, in-the-wild datasets often have wrong or missing labels as users stop labeling when their lives get busy in the real world. We summarize the characteristics of each type of data collection study in Figure (1.1).

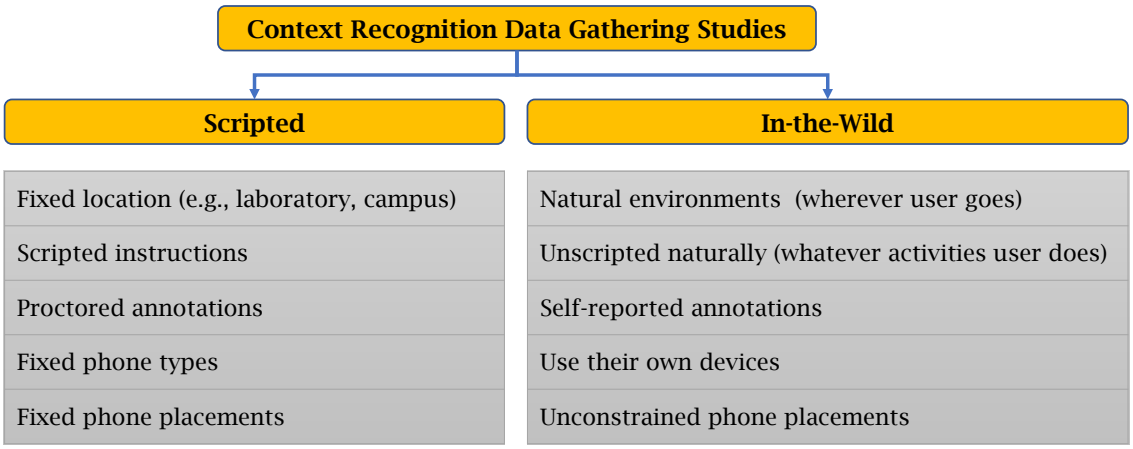


Figure 1.1: Scripted and In-the-wild approaches for Context Recognition Data Gathering.

1.3.1 Existing Human Activity Recognition (HAR) and Human Context Recognition (HCR) Datasets

Several smartphone-labeled HAR and context datasets have been collected for creating machine and deep learning models and publicly released [18, 19, 20, 14]. The majority of these datasets were collected using study designs that were mainly scripted and typically gathered data using one or a few smartphone models that proctors used for the study. Analyzing data from diverse smartphones is vital as prior work has found that sensor readings for the same activity or context can vary by up to 30% across smartphone models. In such cases, models trained on data from only one smartphone model do not generalize well to other smartphones [21].

Consequently, more recent datasets, such as the *ExtraSensory dataset* [14] were gathered using a more realistic *in the wild* study design. Participants installed data collection apps, which collected data passively from their smartphones and

smartwatches simultaneously as they lived their lives. Periodically, subjects were prompted to annotate the activities they performed and contexts they visited using the *ExtraSensory app*. In addition to being more realistic, these in-the-wild studies gathered data using subjects' smartphones that reflected diverse manufacturer hardware. While the *ExtraSensory* dataset came close to meeting our project's needs, several context labels were sparse (subjects rarely performed them). Out of 100 labels defined by the investigators, subjects provided labels for only 51 contexts in 2 weeks of participation in the study. Moreover, this dataset did not gather data on several context labels our project aimed to recognize for our infectious disease and TBI tests.

1.3.2 Our Novel Coincident Data Gathering Study Approach

Our innovative coincident study design approach conducted scripted and in-the-wild gathering studies to gather labeled data in the same contexts shown in Table (1.1). The coincident study facilitates the use of machine learning methods that combine the accuracy of the scripted labels with the realistic context visit patterns of the in-the-wild studies. Our in-the-wild context study was similar to the *Extrasensory* study approach, illustrated in Figure (1.2). The smartphone app continuously gathered sensor data on 103 subjects' phones as they lived their lives. Users were then prompted to self-report labels of contexts they visited. Our scripted study was conducted in a specific laboratory, buildings, or routes on our campus. The smartphone app collected data from 100 participants that visited the listed contexts in Table (1.1) in a scripted fashion. The scripted data-gathering session lasted approximately 1 hour per subject, and human proctors oversaw and

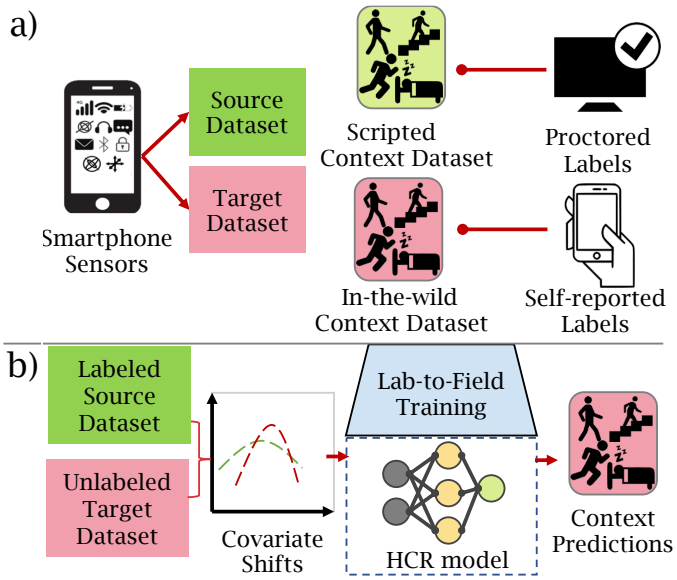


Figure 1.2: Our innovative coincident study design approach. (a) The two kinds of smartphone context data used in this work. (b) Overview of the lab-to-field's problem and approach.

manually annotated the data.

Table 1.1: Contexts for which data was gathered in our WASH Study Collected Contexts - Expanded into 25 binary labels.

Phone Placement	
Phone in Bag	Phone in Hand
Phone in Table Facing Down	Phone in Table Facing Up
Phone in Pocket	
Long activity	
Walking	Sitting
Jumping*	Jogging
Lying Down	Running

Standing	Sleeping
Stairs - Going Up	Stairs - Going Down
Talking On Phone	Trembling*
Typing	In Bathroom
Short activity	
Coughing*	Sneezing*
Standing up (transition)*	Laying Down (transition)*
Sitting Down (transition)*	Sitting Up (transition)*

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

1.4 HCR Challenges

Supervised machine learning classification HCR models typically achieve high accuracy on scripted datasets due to their high-fidelity sensor data and high-quality context labels. For instance, *DeepContext*, a state-of-the-art deep learning HCR model, achieved 91.2% accuracy on a scripted dataset[22]. However, scripted datasets are not realistic as the contexts visited and visit patterns are not representative of real life. It is crucial that HCR models are accurate on in-the-wild datasets, which are more representative of real-world deployment scenarios. However, HCR models achieve lower performance when trained directly on more realistic, in-the-wild datasets. For instance, Vaizman achieved 71.7% accuracy using a Multi-Layer Perceptron (MLP) HCR model trained directly on an in-the-wild dataset[23]. This represents a 19.5% drop in accuracy of state-of-the-art HCR

models on scripted vs. in-the-wild datasets, which underlines the difficulty of the problem of achieving robust, high HCR performance on in-the-wild datasets. Specific issues posed by in-the-wild datasets include Diversity of Causes (DoC) and labeling issues. The approaches that train a robust HCR model on a scripted dataset, which is then transferred to an in-the-wild dataset, face the additional challenge of a covariate shift between the scripted and in-the-wild datasets. These issues are now expounded upon.

1.4.1 In-the-wild HCR Dataset Challenges: Diversity of Causes (DoC)

Collecting smartphone sensor data in the wild often results in naturally occurring variations in the data. While realistic, in-the-wild HCR datasets present challenges due to the diversity of phone placements, smartphone models, human behaviors and environments encountered. These challenges are known collectively as Diversity of Causes (DoC).

1. *Diversity of phone placements:* or positions in which smartphones are placed (prioceptions). Sensor signals have different sensor signatures for the same activity when the phone is carried in different prioceptions [24]. In fact, prioception is one of the most significant sources of variability in smartphone context sensor data [14], as illustrated in Figure (6.2). Smartphone users may choose to carry their smartphones in a bag, their hand, or their coat pocket while performing a given activity (e.g., walking).

2. *Diversity of smartphone models:* Unlike scripted HCR studies where subjects use a single study phone model provided by the proctor, subjects in in-the-

wild HCR studies typically use their own phones. The sensor values recorded for a given context by different smartphone models can differ by as much as 30% [25], presenting an additional challenge for machine learning classifiers.

3. *Diversity of human behavior:* In addition to the above existing diversities in such data, humans, by nature, can perform activities differently. For example, people may walk at a different pace depending on their age, gender, or well-being. Even the same person may walk at different paces depending on their mood. This results in a high inter-person variance in the captured human behavior represented in smartphone sensor data[14, 26, 27, 28]. At the same time, people might change their daily routine or how they use their phone, e.g., people walk about at night due to insomnia, which is not part of their usual routine. These unexpected changes and heterogeneous behaviors can lead to biased representations of human behavior. The robustness of models may be improved by *data generation* and synthesizing in place[29].

4. *Diversity of environments:* The contexts visited by subjects in the scripted study and their context visit order and visit duration differ significantly from in-the-wild scenarios. This results in significant changes in the data generation process from a data-centric perspective [26, 30].

1.4.2 Data Labeling Issues

Acquiring annotated sensor data, in general, is a complicated and expensive task. There are specific issues with data labeling done in in-the-wild data gathering studies, which presents a challenge for supervised machine learning algorithms

[31].listed as the following:

1. *Weakly assigned context labels:* Sensor data preparations for HCR models often rely on data segments taken from a sliding window, and since labels are self-reported, segments that lie within provided times get assigned with provided labels (see Chapter 4 - Section 4.3.1 for details). People might forget or not be precise in their self-reported performed activity times. As a result, segment timestamps may be wrong, and while training HCR models, only particular sub-segment(s) are truly representative of the assigned label within each training sensor segment. However, their exact duration and location within the segment are unknown.

2. *Imbalanced context labels:* context labels in datasets generated by in-the-wild gathering studies are weak and biased toward the user's specific contexts. Subjects typically visit various contexts unequally. For instance, desk workers will have more "sitting at desk" labels than construction workers. Bias also occurs because specific contexts are easier to label than others. For instance, "sitting at a desk" is easier for the participant to label than "swimming," a hands-free activity. The labeling quality also depends on how conscientious the study subject is, which is variable. Such poor, biased, and varied quality of context labels poses a significant challenge for machine and deep learning algorithms.

3. *Noisy and missing context labels:* as users stop providing labels when their lives get busy, or worse yet, they may erroneously provide the wrong labels [32]. As a result, most of the collected sensor data are unlabeled, which

suggests the design of unsupervised HCR models capable of leveraging unlabeled data and reducing impacts of mislabeled (corresponding to incomplete and inaccurate supervision)[33].

Considering this fact about data labeling challenges, learning methods trained under weak supervision are desirable. [34]. Additionally, we may also consider increasing the amount of labeled data by using synthetic *data generations*.

1.4.3 Covariate Shifts Between Scripted and In-the-wild Datasets

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 significant differences between the distribution of features extracted from scripted vs. in-the-wild datasets [35, 36, 37]. More broadly, because real-world applications must face some type of dataset shits, it is critical to address the covariate shift problem for successful deployment of machine learning models in the wild [35].

1.4.4 Adversarial Attacks

Recent work has proven how vulnerable machine learning models are to *adversarial examples*, small perturbations on inputs cause models to produce erroneous classifications with high confidence [38, 39]. At the same time, machine learning models showcase inconsistent and excessively confident performance on out-of-distribution data (o.o.d), and adversarial examples are one kind of this extended

problem [40]. These flaws negatively influence real-world applications in industries where safe and dependable predictions are critical, such as security [41] and healthcare [42]. Importantly, Papernot *et al.*'s work demonstrated that adversarial threats on a specific classifier could be easily transferred to other comparable classifiers [38], adding to the severity of the problem. While researchers have studied these vulnerabilities in-depth in domains such as computer vision [43], natural language processing [44], and speech recognition [45], the focus of adversarial detection methods has only recently been shifted to time-series-based models [46, 47, 48] or sensory-based classifications [49, 50].

Threat Model

As we deal with such large-scale data collection, security concerns arise. Here we describe our threat model briefly. First, the categorization of adversarial attacks on machine learning models can be described as the following:

1. *Poisoning attacks* or training-stage attacks are essentially adversarial contamination of training data. When adversaries have a way to access the model's training data, they may try to undermine the effectiveness of machine learning during deployment by manipulating the training data or its labels. This typically happens by inducing the classifier to learn erroneous associations between input and output [51, 52]. Since HCR data collected in the wild rely on self-reported annotated data, using such data for training HCR models is vulnerable to adversarial contamination as it might not always reflect actual events when data was collected.

2. *Evasion attacks* Deployed HCR models can be targeted with inference-based attacks, often referred to as *evasion attacks* [41], which are the most common and well-studied types of attacks. Evasion attacks occur during deployment when the attacker manipulates test data to fool previously trained classifiers. Manipulating data for this purpose is often referred to as Adversarial Examples in the literature [53, 54]. One example of potential evasion attacks in our context detection model is when users try to fool HCR models by supplying modified test data such that they pretend to perform activities other than their current activities. While the severity of this type might not be as critical as the poisoning attacks, evasion attacks test model robustness against out-of-distribution (o.o.d) testing data subsets [40] or in other applications where attackers might exploit the transferability property of adversarial examples in launching attacks against pre-trained models, which has been proven to work even in HAR datasets [50].

3. *Model extraction or "model inversion" attacks* are mainly concerned with privacy aspects. By reverse-engineering the learning algorithm, these attacks attempt to access confidential information about the system, its users, or data [55]. Specific to HCR, personal information can be inferred from sensor data used to train the model. The reason behind this is that people tend to produce similar characteristics in the captured human behavior represented in smartphone sensor data corresponding to their age, gender, or well-being [14, 26, 27, 28]. The severity of this attack depends on the privacy aspect of sensor-based activity recognition [56]. This dissertation considers poisoning and evasion attacks but not model extraction attacks.

The poisoning and evasion attacks adversarial attacks described above can be perpetrated by various types of subjects, including:

1. *Sloppy or careless actors:* These users have limited knowledge and no apparent intention to degrade model performance. They carelessly provide labels that do not reflect actual events during data collection [26, 57].

2. *Adversary data scientists:* These are expert users knowledgeable in applying data science techniques with clear adversary intentions. They could inject malicious data that could effectively degrade the training or test processes, which could be detected easily by traditional anomaly detection or data sensitization techniques [55, 54]. They could also craft adversarial examples to fool HCR models into doing something else [50]. For instance, one subject could bypass labeling quality procedures that ensure subjects are doing what they are supposed to do in data-gathering studies.

In summary, it is critical to put in place security measures on both data and model architectures to defend against various types of adversarial attacks.

1.5 Dissertation Objective

This dissertation presents HCR research in three broad areas (Illustrated in Figure (1.3)):

1. *Extraction and learning of robust, highly predictive features from sensor data:* overcoming the challenges posed by weak and noisy labeled context data.

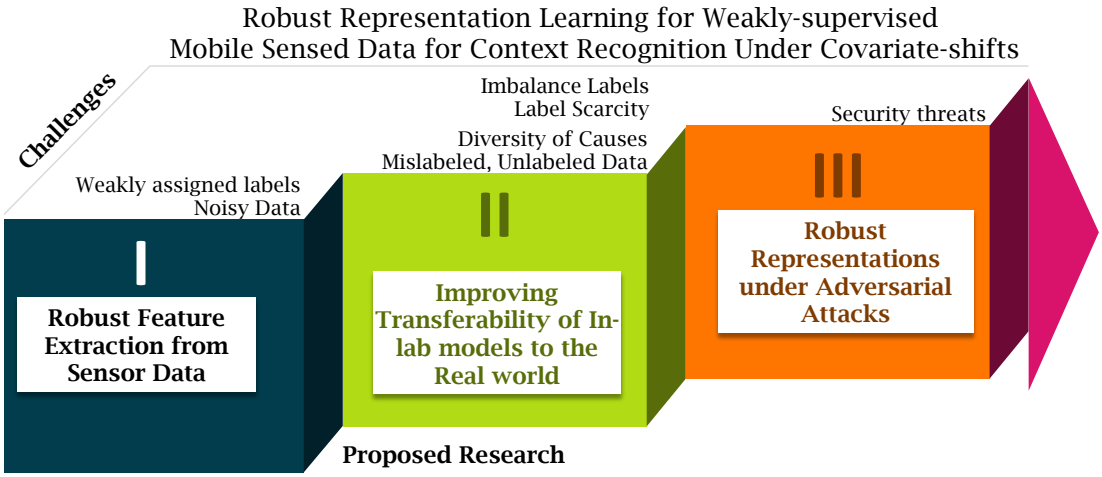


Figure 1.3: Dissertation challenges and objectives summary.

2. *Improve transferability of in-lab models to the real world:* Deploying HCR models in the wild requires learning robust representations that ensure successful transferability of models trained from high-fidelity data gathered in the lab to real-world applications. Transferability is hindered by the challenge of a covariate shift, which is a difference in the sensor and label distributions between the in-lab study data and the in-the-wild study data. Moreover, the in-the-wild dataset is highly noisy as they were self-reported by the smartphone user while living their lives.

3. *Learn representations that are robust to adversarial attacks:* HCR models deployed in the wild might be under adversarial threats such as data poisoning or evasion attacks by bad actors. It is crucial to increase model robustness and generalization to real-world data, using adversarial threats as a measure of robustness.

The overarching goal of this dissertation is to design practical solutions to the above problems using various *representation learning* techniques, a set of meth-

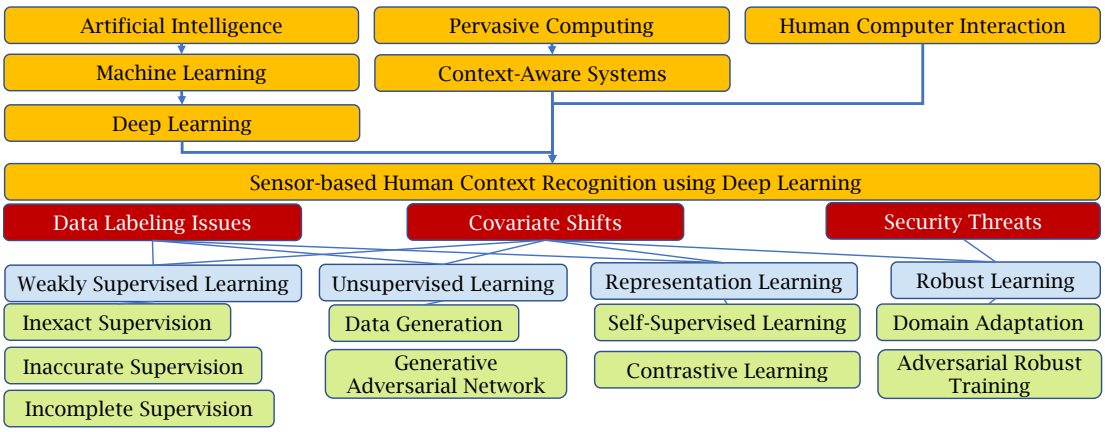


Figure 1.4: A high-level overview of the proposed dissertation in a top-down approach, including interdisciplinary research fields, scope, challenges, and proposed techniques.

ods that allow a system to automatically discover the representations needed for performing machine learning tasks from raw data in a weakly-supervised-learning setting. Figure (1.4) provides an overview of the research fields, challenges, and types of techniques proposed as solutions.

1.6 Dissertation Contributions

In this dissertation, we propose a set of methods to tackle the challenges detailed above and advance state-of-the-art mobile-sensing representation learning for HCR systems. We aim to design methods to improve model robustness and transferability from the lab to real-world applications in a weakly-supervised learning setting. The findings of this dissertation are expected to aid academics in a variety of fields, including ubiquitous computing, human-computer interaction, and machine learning. In this dissertation, we specifically make the following contributions to the following aspects of HCRs:

1. **Human Context Recognition under inexact supervision**

DeepContext has two significant innovations. First, *DeepContext* employs a joint-learning fusion strategy that utilizes both domain-specific handcrafted features with a Multi-Layer Perceptron (MLP) classifier and features that are autonomously generated by a Convolutional Neural Network (CNN). Second, *DeepContext* addresses the problem of coarse-grained labels by discovering and giving higher importance to the most salient regions of the sensor data. These regions have higher predictive value for specific contexts. This allows our model to overcome potentially noisy inputs, which is achieved by *DeepContext's* Parametrized Compatibility-based attention mechanism [22].

2. **Leveraging Coincident Context Data Gathering Study**

   Scripted datasets have accurate context labels, but user behaviors are not realistic. In-the-wild datasets have realistic user behaviors but often have wrong or missing labels. We proposed two methods that try to learn accurate HCR models from scripted datasets to improve performance on a related in-the-wild dataset, focusing on two settings:

   (a) Inaccurate supervision using *PUCL*: Positive Unlabeled Context Learning [26]. *PUCL* uses a transductive positive unlabeled learning methodology to transfer knowledge from the highly-accurate labels of the scripted dataset to the less accurate, more sparse but yet more realistic in-the-wild dataset.

   (b) Incomplete supervision using *Triple-DARE*: Triplet-based Domain Adaptation for Lab-to-field Human Context Recognition. *Triple-DARE* utilized a transductive transfer learning method with triplet loss to adapt

neural networks in various domains to mitigate covariate shifts.

3. **Adversarial-Robust Human Context Recognition**

   We identify and propose defenses for two potential adversarial threats to mobile-sensing in-the-wild gathering studies. *Poisoning attacks* are adversarial contamination of training data, undermining the effectiveness of a machine learning model during deployment. *Evasion attacks* are mainly concerned with manipulating data to deceive pre-trained classifiers, causing them to misclassify data. We aim to improve model robustness and transferability from the lab to real-world HCR applications by learning robust representations of both out-of-distribution testing data and adversarial threats. Defensive approaches will be proposed using robust optimizations and testing them against adversarial threats.

Chapter 2

Background

2.1 Context Sensor Data Collection Studies

Context datasets have either inaccurate labels or unrealistic user behaviors: HCR datasets are collected using study designs that are either *scripted* [58] or *in-the-wild* [59]. *Scripted* studies are typically conducted in a laboratory setting. Participants perform scripted tasks in a fixed, pre-determined order while a smartphone app continuously records reading from smartphone sensors. Human proctors annotate the users' data with corresponding context labels. In *unscripted* ("or in the wild") studies, data is collected for days in the real world as subjects live their lives. A smartphone continuously records smartphone sensor data continuously as subjects live their lives. Periodically, subjects annotate their data with labels of the contexts they visited. While the scripted method for HCR data collection yields exceptionally accurate and consistent labels suitable for supervised machine or deep learning, the contexts visited and sensor data collected in each context are not representative of real life. In-the-wild HCR studies yield more realistic data.

However, some of the context labels may be missing as users forget to label them when they get busy with their lives. Some labels may also be wrong due to human labeling errors [32].

2.2 Attention Mechanisms

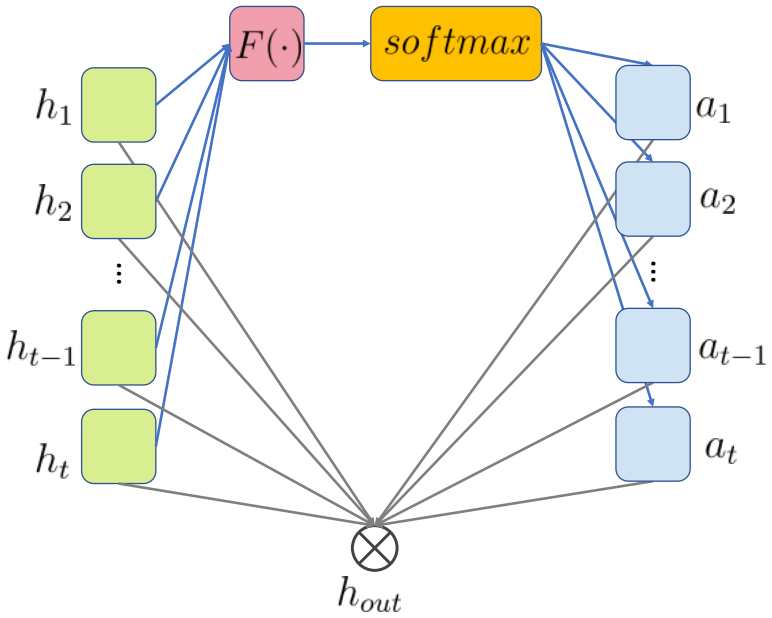


Figure 2.1: The fundamental mechanism of attention.

Attention mechanisms are motivated by how humans pay visual attention only to specific regions of a picture or correlating words in a sentence [60, 61, 62]. Although some attention mechanisms are mainly used during post-hoc analysis of neural networks, several trainable attention mechanisms have been influential not only in increasing the neural network model's performance but also in explaining the final predictions by facilitating the visualization of attention scores. Attention mechanisms try to focus on (weight) a few key features and select the most important ones from a large set of options. The basic attention mechanism is depicted

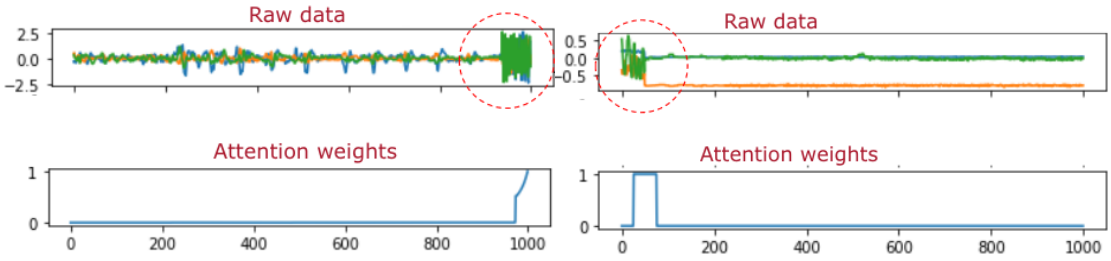
in Figure (2.1). The output feature h_{out} is the weighted sum of each input feature based on its relative importance as follows:

$$\boldsymbol{h}_{\text{out}} = \sum_{i=1}^{t} \alpha_i \boldsymbol{h}_i, \tag{2.1}$$

where h_1, h_2, ..., h_{t1} and h_t are input features, along with their corresponding weights: a_1,a_2, ..., a_{t-1} and a_t. The number of input features is denoted by t. α_i is obtained by a softmax function along with a scoring function $F(\cdot)$, as the following:

$$\alpha_i = \frac{\exp\left(F\left(\boldsymbol{h}_i\right)\right)}{\sum_{i=1}^{t} \exp\left(F\left(\boldsymbol{h}_i\right)\right)} \tag{2.2}$$

There are two main types of attention mechanisms: 1) hard attention and 2) soft attention [61]. Hard attention is a stochastic process and often cannot be trained through back-propagation. Thus, the distribution of attention scores has to be assumed and fixed a priori [61]. Soft attention uses a probabilistic distribution function to apply attention scores to the source input [61], which makes it more suitable for sensor data, where a fixed distribution of scores or the size and number of attention regions to focus on cannot be assumed a priori. As shown in Figure (2.2), we employ an attention mechanism in our robust feature generation method in order to discover and focus on the most pertinent regions of the sensor data, which exhibit patterns that predict particular contexts.



(a) Phone In Hand, Sneezing (b) Phone In Hand, Typing

Figure 2.2: This figure illustrates how we use attention to identify representative subsegments in one instance of coarsely labeled sensor data. On top, only raw accelerometer data is plotted (different colors represent the various trial axes). On the bottom, normalized learned attention weights are plotted. More information is available in Chapter 4.

2.3 Covariate Shifts

The term "Covariate Shifts" was first introduced by Shimodaira [63], and is described as changes in the distribution of the input x. While there are other types of existing dataset shifts [35], the most researched type is covariate shift. Covariate shift occurs when data is generated according to a model $P(y|x)P(x)$ and the distribution $P(x)$ differs across training and test scenarios. While there is some ambiguity in the definitions of covariate shift in the literature, we found the definition provided by Moreno-Torres *et al.* [35] the most relevant, given by the following conditions:

$$P_{tr}(y \mid x) = P_{tst}(y \mid x) \text{ and } P_{tr}(x) \neq P_{tst}(x), \tag{2.3}$$

where $P_{tr}(x)$ and $P_{tst}(x)$ 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 perform well on a test set sampled from an in-the-wild dataset,

we are faced with covariate shifts, where the distribution of features differs across training (scripted) and test (in the wild) scenarios.

2.4 Robustness

In a broad sense, robustness is an overused phrase that may be interpreted in a number of different ways for machine learning models. This includes but is not limited to preserving task performance on manipulated or changed inputs [64], generalization across domains [65], and resilience to adversarial attacks, also known as *adversarial robustness* [66, 67, 68]. In our work, we aim to improve model robustness in general and decrease prediction inconsistency under low changes in the input using adversarial threats as a measure of model robustness because we analyze mobile-sensed data that contain many naturally occurring variations. Moreover, it has been demonstrated that adversarial robustness is closely connected with covariate shift resiliences [69, 70, 65]; thus, researchers are urged to evaluate adversarial robustness defensive measures against data with covariate shifts. Adversarial attacks are a helpful way of analysis for generalizing under the worst possible conditions, making them a viable instrument for evaluating the robustness of models in general [71, 40]. Notably, the best practices for deploying AI models in healthcare emphasized the need for robustness, safety, and security while developing trustworthy AI systems [72, 73, 74].

2.5 Motivating HCR Use Case: DARPA WASH Project

The DARPA-funded Warfighter Analytics using Smartphone for Healthcare (WASH) DARPA project [17] is investigating passive smartphone assessment of TBI and infectious diseases. This will provide an up-to-date assessment of the warfighter's battle readiness. Target populations include active duty service members and veterans initially, but discoveries made will also apply to civilians.

In the envisioned use case, the WASH smartphone app will passively gather smartphone sensor data throughout each day. Each day of data is then pushed to the cloud overnight for analysis. Disease inference models will analyze this data in the cloud to generate a *bioscore* (or probability of illness) for each warfighter.

Program phases: The WASH program is divided into two distinct phases. Phase one involves recognizing specific smartphone user contexts in which targeted health assessments will be conducted. Phase two involves creating the methods for the actual TBI and infectious disease assessments of smartphone users. In phase one, we researched and created a list of smartphone biomarkers that were predictive of TBI and infectious diseases and corresponding contexts. Our team conducted user studies to collect labeled data for those contexts and created HCR models to infer those contexts from labeled smartphone sensor data. Table 2.1(a) shows the list of contexts, such as "walking, phone in hand" that our team gathered labeled data on and created HCR models. The planned ailment-specific tests or biomarkers corresponding to each of these contexts are listed in Table 2.1(b). We created our list of ailment tests and contexts in consultation with TBI and infectious disease experts from the University of Massachusetts Medical School

(UMMS). As a concrete example, shaking hands is a sign of TBI. In phase one, our team conducted user studies and created deep-learning models to detect the smartphone user holding their phones. In phase two, we focus on assessing whether the user's hand is shaking. This dissertation focuses only on context recognition. Research into the actual context-specific ailment assessments is not covered.

2.6 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 is comprised of an input feature vector (or instance) and an associated label (or ground truth). In many practical scenarios, such as in our in-the-wild HCR study, it is challenging to gather adequate high-quality labels for fully supervised learning due to the high costs of gathering labeled data. 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.* [34], 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, within each training sensor segment, only certain sub-segment(s) are truly representative of the assigned label. However, their exact duration and location within the segment are unknown.

2. *Inaccurate supervision* in which data labels are not always correct. For ex-

Table 2.1: Context-specific ailment tests to detect TBI and infectious diseases and relevant human contexts.

a)

Target Contexts	
Laying Down, Phone on Table	Exercising, Phone in Pocket
Toilet, Phone in Pocket	Walking, Phone in Bag
Walking, Phone in Hand	Walking, Phone in Pocket
Typing	Sleeping
Sitting	Running
Laying Down (state)	Standing
Talking On Phone	Bathroom
Phone in Pocket	Phone in Hand
Phone in Bag	Phone on Table, Facing Up
Phone on Table, Facing Down	Stairs - Going Up
Stairs - Going Down	Walking

b)

Traumatic Brain Injury	
Ailment Test	**Test Context**
Worse Reaction Time	<Interacting with Phone, in Hand, *, *>
Increased Light Sensitivity	<*, in Hand, *, *>
Unilateral Pupil Dilation	<Interacting w/ Phone, in Hand, Texting, *>
Hands Shaking	<*, in Hand, *, *>
Slurred Speech	<Talking into Phone, *, *, *>
Infectious Diseases	
Ailment Test	**Test Context**
Increased Cough Frequency	<Coughing, *, *, *>
Increased Sneezing	<Sneezing,*, *, *>
Resting Heart Rate	<Sitting, in Pocket, *, *>
Increased Toilet use Frequency	<Using Toilet, *, *, *>
Change in respiration	<Sleeping, on Table, *, *>
	<Exercising, *, *, *>
Both TBI and Infectious Disease	
Ailment Test	**Test Context**
Increase In Activity Transition Time	<Lying down, Phone In Pocket, *, *>
	<Sitting, Phone In Pocket, *, *>
	<Standing, Phone In Pocket, *, *>
Change in Sleep Quality	<Sleeping, *, *, *>
Change in Gait	<Walking, Phone in Pocket/Hand, *, *>

ample, 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. For instance, some of the context labels in the dataset may be missing as users forget to label them when they get busy with their lives.

2.7 Proposed Solutions to Address Weak Labeling

For these various forms of weak labeling, innovative learning methods that are trained under weak supervision are desirable [34]. In this dissertation, we have proposed solutions for Smartphone HCRs in each WSL category including:

1. **Mitigating weak labeling in DeepContext**: We proposed *DeepContext*, an HCR system that uses neural networks to recognize smartphone user contexts under inexact supervision. *DeepContext* can extract salient discriminative features under weakly labeled scenarios. Utilizing an attention mechanism, *DeepContext* can autonomously learn context-specific salient features while suppressing potentially irrelevant parts of the input, tackling the issue of coarse-grained labeling that usually exists in smartphone sensor data.

2. **Mitigating mislabelled data in PUCL**: Smartphone HealthBiomarkers: focuses on Positive Unlabeled Learning of In-the-Wild Contexts. PUCL addresses the issue of mislabeled data in in-the-wild datasets by training a deep neural network with a correcting factor learned from the high-fidelity scripted dataset that puts less attention on instances that are most likely mislabeled.

3. **Domain adaptation across coincident HCR datasets in Triple-Dare**: We proposed *Triple-Dare*, a deep learning method that improves the performance of HCR models by first training them on similar scripted datasets, then adapting them for use in predicting context labels in in-the-wild datasets. We utilized coincident scripted and in-the-wild HCR datasets in which similar context labels were gathered in both studies. Triple-DARE can leverage the tremendous amounts of unlabeled in-the-wild data, decreasing the need for human-annotated labels.

   In subsequent chapters, we describe these solutions in detail.

Chapter 3

Literature Review

In this chapter, we include the relevant research work.

3.1 Related Applications for Mobile-sensed Data

3.1.1 Human Context Recognition Using Smartphones

As it is typical for individuals to have their smartphone close by the majority of the time, smartphones are an excellent tool for HCR [14, 26] and passive sensing [75]. In order to capture realistic human behavior, smartphone HCR data collection studies collect data while users live their lives, performing whatever activities they choose [76, 77, 78, 14, 22]. Recently, Vaizman *et al.* collected and analyzed a dataset containing a large number of participant-reported labels, combining numerous smartphone and smartwatch sensor modalities, and identifying human behavioral context using shallow neural network models with handcrafted features [14]. Our definition of context ⟨Activity, Prioception⟩ incorporates the user's activity and thus relates to Human Activity Recognition (HAR), a well-

studied research topic [79, 80]. As more data became available, numerous deep-learning architectures utilizing smartphone sensor data have been proposed for HAR [79, 80]. Nevertheless, these designs are only able to classify sensor data into one of k possible labeled activities. In addition, these conventional HAR methods are not applicable to real-world problems because most of them presume that the sensor is located at a specific location on the body (waist, hip, or wrist) [81].

3.1.2 Smartphone-based Mission-Critical Applications

Smartphone-based recognition of user context and ambulatory activities [82] has several practical, mission-critical applications. Compromising such systems could have serious ramifications. For instance, smartphone-based HCRs may be utilized to continuously track and monitor the health of soldiers or veterans to detect Traumatic Brain Injuries (TBI) or infectious diseases (e.g., Covid-19). By monitoring smartphone health biomarkers, abnormal user behavior, physiological indicators, activities, and context visit patterns can be identified [26]. Sun *et al.* showed that smartphone-based activity recognition could detect aggravated assaults. They created iProtect to identify abuse and kidnapping. iProtect uses smartphone accelerometers to record and identify physical assaults. The importance of real-time assault detection for personal safety cannot be overstated [83]. These applications require accurate smartphone HCR predictions, which we aim to improve.

3.2 Sensory Representation Learning

3.2.1 Handcrafted-features based Methods

Most prior work for sensory representation learning utilizes handcrafted features to learn discriminative sensor features, which incorporates prior knowledge but limits the learned representation's capacity by reliance on human creativity [27, 84, 85, 86]. Such methods lack the power to capture underlying non-linear patterns in low-level sensory inputs [28].

3.2.2 Deep-learning based Methods

While handcrafted features may suffice to recognize simple cases, deep learning methods have been shown to be more effective in complex HAR tasks [87, 88, 89, 87, 90]. Deep learning has shown potential in HAR and HCR, extracting valuable features for the target task automatically [28, 23, 22, 26]. However, the bias and constraints introduced by traditional laboratory-based scripted mobile sensing datasets may negatively affect the performance of both classic HAR approaches and deep learning models in deploying HCRs for real-world use cases. Considering limitations resulting from their scale, diversity, and ability to capture the richness and complexity seen in in-the-wild, unconstrained data is crucial. Moreover, these architectures classify sensor data into only one of k possible labeled activities [91, 92]. Moreover, these conventional human activity recognition methods are not suitable for real-world problems since most of them assume that the sensor is placed at a fixed location on the body (hip, wrist, or waist) [28]. The difficulty in gathering labeled data outside of laboratory conditions only adds to the constraints

of mobile sensing datasets [26, 93].

3.3 Weakly Supervised based Methods

In many practical scenarios, such as in our in-the-wild HCR study, gathering adequate high-quality labels for fully supervised learning is challenging due to the high costs of gathering labeled data. In such practical scenarios, various types of weak (or inaccurate) labels can occur, requiring innovative learning methods that can work under weak supervision, such as unsupervised learning methods (more details are given in Chapter 2). While there exist a few HCR methods using mobile-sensing data for supervised deep learning [14, 94], self-supervised learning [95, 93, 96] or active-learning [97], they provide limited, or no, solutions to the aforementioned data labeling or DoC naturally occurring variations challenges. Additionally, there are several other existing prior works focused on leveraging techniques that minimize data annotations [93, 98, 96] applied to HAR tasks. However, exploring such methods to improve the performance of in-the-wild mobile sensed HCR data is yet to be studied.

3.3.1 Positive Unlabeled (PU) Learning

In PU learning, only a subset of the dataset is labeled. The training dataset has some labeled positive examples $\mathcal{P}$ and a set of unlabeled examples $\mathcal{U}$ that is a mixture of positive and negative examples. The goal is to create a binary classifier that can classify previously unlabeled instances or correct wrong labels in a test set into the positive or negative class [99]. PU Bagging [99] is an ensemble method

that creates a set of trees. Each tree is trained on a subset of the entire training set and is used to classify the positive instances from the unlabeled instances. Next, each tree scores all the remaining instances in the in-the-wild dataset except the instances it was trained on. The average prediction probability from all trees for each in-the-wild instance is then given as the probability that the instance is of the positive class.

3.3.2 Domain Adaptation (DA)

Research has made substantial progress in adapting deep neural networks to various related domains [36]. Recent deep DA methods are either discrepancy-based approaches that minimize a discrepancy metric over feature distributions [100, 101], or adversarial-based approaches [102] that aim to maximize domain confusion. The Deep Adaptation Network (DAN) [100] minimized the mean distance between two feature distributions in a reproducing kernel Hilbert space, effectively matching the higher-order statistics of the two distributions. On the other hand, the deep Correlation Alignment (CORAL) [101] technique proposed matching the mean and covariance of two distributions. Other strategies have used an adversarial loss to maximize domain confusion [102].

3.3.3 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 [24, 103, 104, 105]. Previous work has shown that DA can be used to unsupervisedly learn domain-invariant accelerometer [24, 103] and gyroscope [24] 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 [103] 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) [104] 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 the 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 [105]. DA was also used to adapt models to subject variability [106], 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 [103, 24]. 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 [107, 108]. A summary of this subsection related work is included in Table (3.1).

Table 3.1: DA for sensor-data related work summary.

Research Work	**Method**	**Type of Data**	**Lab-to-Field**	Discrepancy Minimization
Natarajan *et al.* [109]	Importance-reweighting	Wearable electrocardiogram sensor data	Yes	No
Chang *et al.* [24]	Feature matching	Wearable sensor data	No	Global only
Long *et al.* [100]	MK-MMD	Images	No	Global only
Sun *et al.* [101]	Correlation Alignment	Images	No	Global only
Khan *et al.* [103]	KL Divergence	Smartphone and smartwatch sensor data	No	Global only
Chen *et al.* [104]	Stratified Transfer Learning	Wearable sensor data	No	Non-scalable intra-class separation
Sanabria *et al.* [105]	Variational Autoencoder	Binary event sensor data	No	Global only
Wilson *et al.* [106]	Using target label distribution	Wearable sensor data	No	Global only and utilized target labels

3.3.4 Mitigating Poor Labeling Quality

In order to mitigate missing labels, some prior mobile-sensed data collection methods include interfaces and mechanisms that subjects can utilize to label batches of past contexts retrospectively whenever they have free time [59]. However, as subjects often do not accurately remember some contexts or their start/end times, recall bias diminishes the quality of labels. Careless subjects may also provide many wrong labels [110]. Zeni *et al.* [111] devised an interactive machine-learning framework for testing user trustworthiness by checking the consistency of the user-provided annotations using available ground truth. Their work required continuous feedback from the user, which is undesirable and focused on user location only.

3.4 Adversarial Threats

Researchers have studied *adversarial examples* vulnerabilities in computer vision [43], natural language processing [44], and speech recognition [45], the focus of exploring adversarial vulnerabilities has only recently shifted to time-series-based models [46, 47, 48] or sensory-based classifications [49, 112, 50]. Sah *et al.* recently studied utilizing wearables for activity recognition, investigating the transferability and generation of adversarial examples. However, they did not propose any countermeasures [50]. Moreover, in comparison to wearables, the nature of HCR data collected by smartphones is much more complicated. For instance, sensor signals for the same activity have different characteristics when the phone is held in various prioceptions [24, 25]. In fact, prioception has the most signifi-

cant impact regarding differences in smartphone context sensor data [59]. When performing a particular activity, smartphone owners may choose to either hold the smartphone in their hand or place it in their pants or coat pocket. Therefore, methods to evaluate the security of HCR data collected by smartphones in the wild are needed, especially for large amounts of data with many diversities. To understand the impact of this problem, imagine a detection algorithm failing to detect an older person's fall or a warfighter with TBI not being detected on time due to an adversarial attack. There is a dire need to define and evaluate the effects of adversarial examples for smartphone HCRs, which could have fatal outcomes if perpetuated in CA systems for mobile health and behavioral medicine.

3.4.1 Potential Adversarial Attacks Specific to HCRs that we Focus on

Poisoning and Evasion are two major adversarial attacks [41]. *Poisoning* or training-stage attacks contaminate training data. When adversaries can access a model's training data, they may try to undermine the machine learning model by manipulating the data or labels. *Evasion attacks*, or inference-based attacks, are common and well-studied. This paper focuses on evasion attacks, which occur when an attacker manipulates test data to fool classifiers. Adversarial attacks can be divided into white-box and black-box attacks based on system knowledge and access [41]. White-box attacks require model parameters [113, 53]. Black-box attacks (listed in Figure (3.1)) require the ability to query the model with arbitrary inputs [114, 115]. We focus on score-based and label-based evasion attack generation methods in accordance with plausible scenarios of possible adversarial

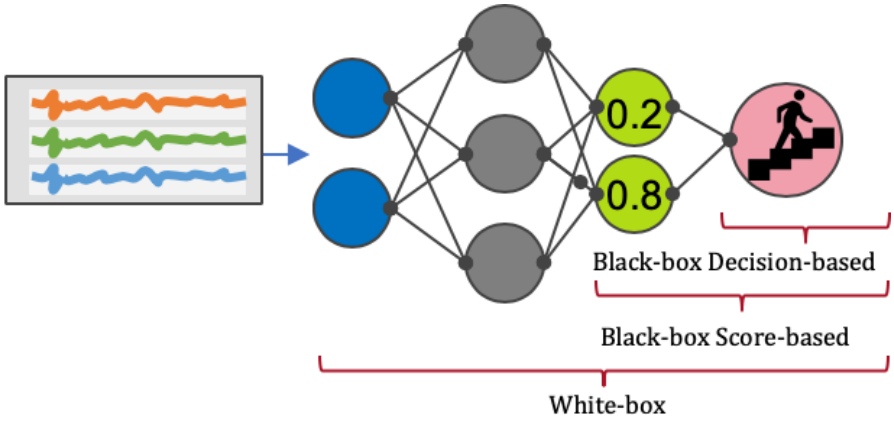


Figure 3.1: Types of Evasion attacks according to the knowledge needed to carry out the attack.

attacks. These methods can generate adversarial perturbations using only class confidence scores (Zoo attack) or class decisions (HSJ attack).

Part II

Robust Feature Extraction from Sensor Data

Chapter 4

Human Context Recognition under inexact supervision

4.1 Introduction

We proposed *DeepContext*, an HCR system that uses neural networks to recognize the smartphone user contexts in which the TBI and Infectious diseases tests can be performed (See the Background chapter). *DeepContext* has two major innovations. First, *DeepContext* employs a joint-learning fusion strategy that utilizes both domain-specific handcrafted features and features that are autonomously generated by a Convolutional Neural Network (CNN). Second, *DeepContext* addresses the problem of coarse-grained labels by discovering and giving higher importance to the most salient regions of the sensor data. These regions are expected to correspond to a higher predictive value for specific contexts. This allows our model to overcome potentially noisy inputs, which is achieved by *DeepContext's* parametrized compatibility-based attention mechanism.

4.2 Prior Work

Prior studies have focused on the related problem of recognizing ambulatory human activities (e.g., sitting, walking, running, etc.), also called Human Activity Recognition (HAR), though they typically classify the sensor data into only one out of k possible labeled activities [86, 27]. However, while human context includes the person's current activity, it is also critical to include other semantic information such as their location and social situation. While there exist a few HCR methods that aim to classify human behavioral context [14, 94], they still do not address coarse-grained labeling.

4.3 *DeepContext* Approach

In this work, we use the scripted HCR dataset, manually annotated by proctors who oversaw the study. The labels they assigned are however coarse-grained, not fine-grained labels. Formally, in our training data set $D = (X_1, y_1), \ldots, (X_m, y_m)$ where $X_i = \{x_{i1}, \ldots, x_{i,m}\} \subseteq \mathcal{X}$ is a bag, $x_{ij} \in \mathcal{X}$ $(j \in \{1, \ldots, m_i\})$ is an instance, m_i is the number of instances in X_i, $y_i \subseteq \mathcal{Y} = \{0, 1\}$, X_i is a positive bag, i.e. $y_i = 1$, if there exist(s) one or more positive x_{ip}, while $p \in \{1, \ldots, m_i\}$ is unknown. In other words, within each training sensor segment, only certain subsegment(s) are truly representative of the assigned label. However, their exact duration and location within the segment are unknown (corresponding to inexact supervision) [34]. As many of our target activities can be performed concurrently (e.g., walking and talking on the phone), *DeepContext* formulates the HCR problem as a multi-label classification problem in a manner similar to Vaizman *et al.*

[116].

4.3.1 Overview

Our deep learning architecture for Human Context Recognition (*DeepContext*) is comprised of two CNNs with the attention that jointly learn from raw smartphone sensor data and handcrafted features in parallel, fusing their outputs. Our attention mechanism is inspired by a promising model proposed by Jetley *et al.* , an end-to-end trainable attention mechanism for CNN for the task of object detection and localization [117]. Fig. 4.2 shows the overall architecture of *DeepContext*. This joint learning fusion approach enables our model to learn not only discriminative features from handcrafted features and raw sensor data, but also from a shared representation, discovering complex cross-modality correlations. Moreover, the attention mechanism utilized enables *DeepContext* to learn salient features, giving higher weights (importance) to regions of the raw sensor data that contain predictive features for context recognition. Figure (4.1) shows *DeepContext's* classification pipeline.

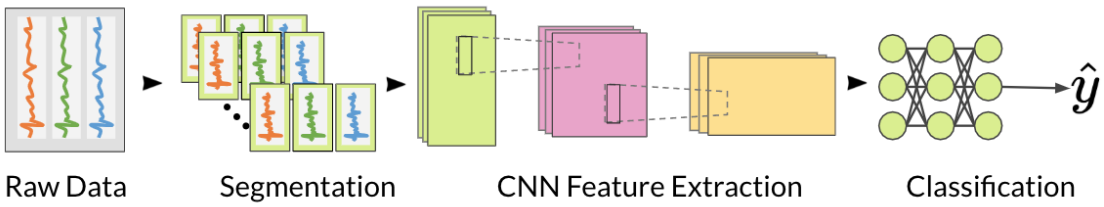


Figure 4.1: Classification Pipeline for Raw Sensor data - showing a CNN feature generation approach.

Figure (4.1) shows the classification pipeline for HCRs that utilizes sensory data. Sensory data is initially segmented using sliding windows to generate training instances, which are then input to CNN layers that extract feature vectors that

are utilized for context prediction later in the pipeline [118]. Formally, we use a sliding window N to generate segmented input vectors (e.g. accelerometer and gyroscope). For segmented sensor data input vectors $x_i^0 = [x_{r_1,\ldots,x_N}]$. Thus, in the first convolutional layer the output will be:

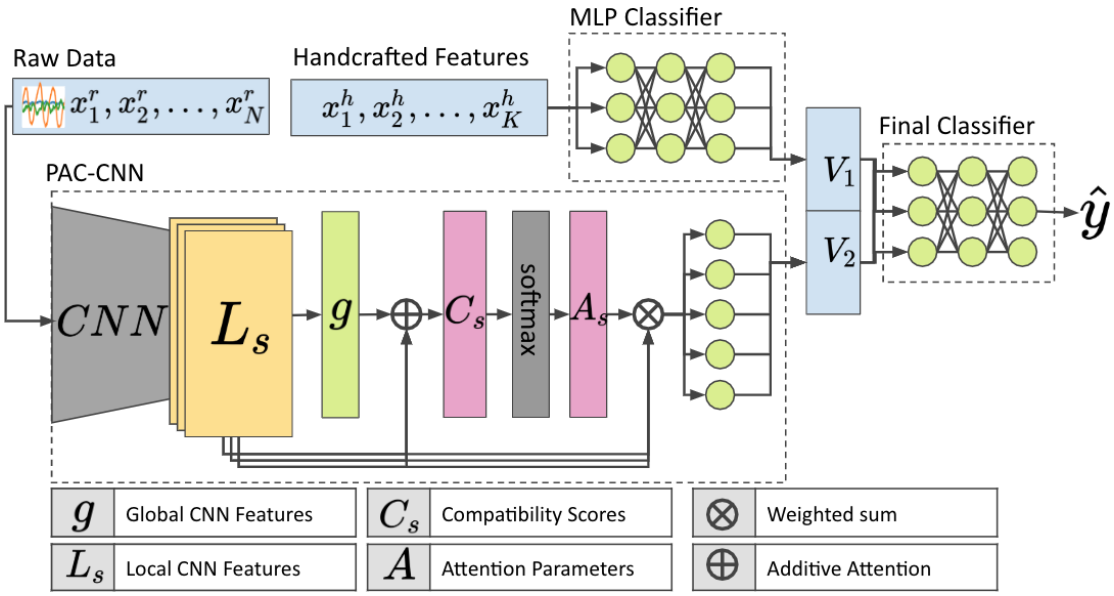
$$c_i^{1,j} = \sigma\left(b_j^1 + \sum_{m=1}^{M} w_m^{1,j} x_{i+m-1}^{0,j}\right) \tag{4.1}$$

σ being the activation function, b_j is the bias introduced by the j the feature map, M is the size of the kernel used, and w_m^j is the weight associated with the feature map j and the filter index m.

The design of our CNN feature extractor follows a separate-and-merge strategy, which produces state-of-art performance on mobile-sensing data, as proposed in[119], where data generated by each sensor is first passed into a single-sensor CNN model that learns local interactions within each sensor. The outputs of individual single-sensor CNNs are then concatenated together to form a cross-modality representation that is then passed to additional CNN layers to learn global cross-sensor interactions.

4.3.2 Parameterized Compatibility-Based Attention Convolution Neural Network (PAC-CNN)

The context labels that subjects assign to smartphone sensor data during data gathering studies are often coarse-grained, making it challenging to create reliable context classifiers. Specifically, only relatively small regions of data that a user has assigned a given context label (e.g. walking) may actually be truly represen-



N: Raw Data segment size. K: Handcrafted Features dimension

Figure 4.2: *DeepContext* architecture.

tative of that context. *DeepContext's* attention mechanism tries to learn the most relevant regions of the sensor data, which exhibit patterns that predict specific contexts. The intuition behind the design of its attention mechanism is similar to that proposed by Jetley *et al.* [117].

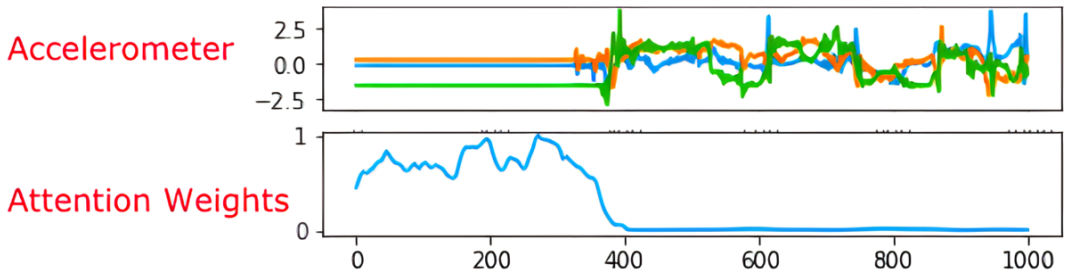


Figure 4.3: The attention mechanism assigns more importance to regions of data that contain salient context-specific features extracted from raw sensor data. For instance, the attention mechanism learns that the left side of the accelerometer signal better represents *Phone on Table* context and assigns it higher weights.

In Fig (4.3), the attention model ignores parts of the sensor data when trying to classify the *"Phone on the table"* context. The model learns predictive patterns

and increases their influence, while simultaneously suppressing irrelevant and potentially noisy parts of the data. As more data is utilized in training the model, it learns representations that are more generalizable and work better in real-world settings. The important regions detected within the data form saliency maps that could be analyzed to interprete classifier outputs, improve its performance and potentially facilitate the data-labeling process [120].

$\mathcal{L}^s = \{\ell_1^s, \ell_2^s, \cdots, \ell_n^s\}$ are intermediate (local) features extracted by convolutional layer $s \in \{1, 2, \ldots, S\}$, where l_s^i is extracted from the ith node out of a total of n nodes, each corresponding to one spatial location in the local feature vector $\mathcal{L}^s$.

In order to adapt the attention mechanism of Jetley *et al.* [117] that was designed for images, to fit the multiple-modality nature of smartphone sensor data, we considered s to be various intermediate layers in the seperate-and-merge [119] CNN pipeline.

The flattened (global) feature vector G generated by the fully connected layer is combined with the final set of CNN-extracted (local) features. The attention mechanism tries to learn a compatibility score $\mathcal{C}\left(\hat{\mathcal{L}}^s, \boldsymbol{g}\right) = \{c_1^s, c_2^s, \ldots c_n^s\}$ between the local features $\mathcal{L}^s$ and the global feature vector G, and replaces the final feature vector with an attention-weighted local features [117]. In order to constrain the parameters of the attention unit, concatenation can be reduced to an addition operation due to the existence of free parameters between the local and global feature descriptors [117].

To calculate the compatibility score, G and l_s^i are concatenated using an addition operation (additive attention [60]), followed by a dot product with a trainable

weight vector u that can be expressed as [117]:

$$c_i^s = \langle u, l_i^s + G \rangle, i \in \{1, n\} \tag{4.2}$$

These learned compatibility scores c_i^s encourage the model to learn discriminative features tailored to different contexts. In order to utilize these learned compatibility scores $C(L^s, G) = \{c_1^s, c2^s, \ldots, c_n^s\}$ to produce a 1-dimensional vector $A^s = \{a_1^s, a2^s, \ldots, a_s^n\}$, a down-sampling convolutional layer is first applied, then the compatibility scores are normalized using a softmax function:

$$a_i^s = \frac{\exp\left(c_i^s\right)}{\sum_j^n \exp\left(c_j^s\right)} \tag{4.3}$$

The last step involves producing the final attention estimation g^s, replacing G, by taking the element-wise weighted average of the corresponding normalized compatibility scores in A^s with each node in L^s.

In Fig (4.4) we show the CNN architecture used.

$$g^s = \sum_{i=1}^{n} a_i^s \cdot l_i^s \tag{4.4}$$

4.3.3 Joint-learning Fusion

Taking advantage of the joint-learning fusion strategy, we can accommodate various modalities that cannot be fed to a CNN directly. By learning a shared representation between handcrafted features and CNN-generated features, our model increases its ability to learn cross-sensor representations that are more discrimi-

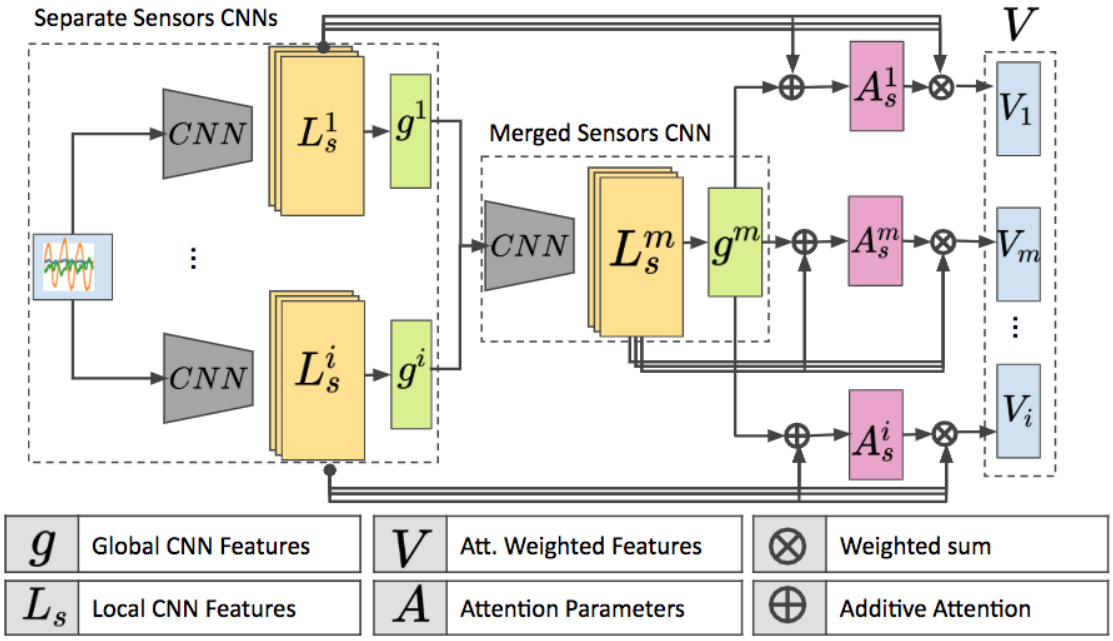


Figure 4.4: Applying the attention mechanism on the separate-n-merge CNN architecture, where we use a separate CNN for each sensor modality, concatenating the resulting CNN outputs that are finally passed to the merged-sensors CNN. Only attention-weighted features are used for subsequent classification layers.

nating for prediction tasks [121]. This shared representation can act as a regularization technique and discover additional task-specific correlations between the handcrafted and CNN-generated features. To generate this shared representation, we first forward handcrafted features to a multi-layer-perceptron neural network, which consists of two layers, 16 hidden nodes in each layer, and uses Rectified Linear Units (ReLU) as its activation function. Then, we concatenate the resulting vector along with CNN-generated features after they are mapped to the same dimension. A sample list of handcrafted features[84] extracted from our smartphone sensor data is provided in Table (7.2).

Table 4.1: A sample list of handcrafted features used for our sensor data, applied on accelerometer, gyroscope and magnetometer sensors, adopted from [14, 84].

Feature	**Formulation**
Arithmetic mean	$\bar{s} = \frac{1}{N}\sum_{i=1}^{N} s_i$
Standard deviation	$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(s_i - \bar{s})^2}$
Median absolute deviation	$\text{median}_i\left(\lvert s_i - \text{median}_j\left(s_j\right)\rvert\right)$
Largest values in array	$\max_i\left(s_i\right)$
Smallest value in array	$\min_i\left(s_i\right)$
Frequency signal Skewness	$\text{E}\left[\frac{(s-\bar{s})^3}{\sigma}\right]$
Frequency signal Kurtosis	$E\left[(s-\bar{s})^4\right]/\text{E}\left[(s-\bar{s})^2\right]^2$
Largest frequency component	$\arg\max_i\left(s_i\right)$
Average sum of the squares	$\frac{1}{N}\sum_{i=1}^{N} s_i^2$
Signal magnitude area	$\frac{1}{3}\sum_{i=1}^{3}\sum_{j=1}^{N}\lvert s_{i,j}\rvert$
Interquartile range	$Q3(s) - Q1(s)$
Signal Entropy	$\sum_{i=1}^{N}\left(c_i \log\left(c_i\right)\right), c_i = s_i / \sum_{i=1}^{N} s_j$
Pearson Correlation coefficient	$C_{1,2}/\sqrt{C_{1,1}C_{2,2}}, C = \text{cov}\left(s_1, s_2\right)$
Frequency signal weighted average	$\sum_{i=1}^{N}\left(is_i\right) / \sum_{j=1}^{N} s_j$
Spectral energy of a frequency band [a, b]	$\frac{1}{a-b+1}\sum_{i=a}^{b} s_i^2$

s: signal vector, N: signal vector length Q: quartile

4.4 Evaluation

We conducted experiments to evaluate *DeepContext's* performance on WASH scripted dataset for various segmentation window sizes (in seconds). First, we describe the evaluation protocol and metrics used to assess the model's performance, given the imbalanced nature of the dataset. Then, we assess the effectiveness of different components of *DeepContext* and discuss our empirical findings.

4.4.1 Implementation

Our separate-n-merge CNN is has three layers per single-sensor CNN, and three additional layers for the merged-sensors' CNN. The number of feature maps generated in each CNN layer is 64. We also found that using larger filter sizes at the beginning of the pipeline produced better results, so we selected 8, 6, and 4 respectively as our filter sizes. We utilized Rectified Linear Units (ReLU) as our non-linear activation function. Our input batch size was 128 and we utilized dropout regularization with a probability of 20%, batch normalization, as well as L1/L2 normalization with a coefficient of $1e-5$. The model was trained for 100 epochs with early stopping if the validation loss stopped improving, to decrease the chance of over-fitting. For visualizing compatibility scores, we followed the same procedure used in [120].

4.4.2 Evaluation Protocol

To ensure that our model generalized well when utilized on data from new subjects, previously unseen subjects during the training process, we adopted a user-level cross-validation approach (5 folds). Similar to the user-level splitting approach utilized by Vaizman *et al.* [116], all of a subject's data may appear in either the training or test set, but not in both. Our final output is a multi-label output vector, where each label produced is a binary output (E.g walking vs not walking). To address the class-imbalanced nature of our WASH study dataset, we utilized *Balanced Accuracy* (BA), as our metric for evaluating our model's performance [122].

$$BA(\mathcal{D}) = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right)$$

which is also:

$$BA(\mathcal{D}) = \frac{1}{2}\left(Sensitivity + Specificity\right)$$

Also, in order to compute the BA of the context tuple after recomposition from the binary labels, we adopted macro-averaging that treats all binary labels with equal importance. That is, we calculate the BA score for each binary label separately and report the average across all binary labels (macro BA).

$$BA^{macro}(\mathcal{D}) = \sum_{c_i \in C} \frac{BA(\mathcal{D}, c_i)}{|C|}$$

.

When there are no annotated examples for c_i, then $BA(\mathcal{D}, c_i)$ is excluded from BA^{macro} calculation.

We compared our model performance against state of the art deep learning HCR (ExtraSensory MLP [23]) and HAR (DeepSense CNN-GRU [123]) models. To ensure that our comparison was fair, we only utilized handcrafted features extracted from data from three sensors accelerometer, gyroscope and magnetometer. *DeepContext* and the other models compared against are all implemented in PyTorch [124], based on the authors' published source codes. Each model was then fine-tuned on our dataset and the same highly tuned number of layers and feature maps hyper-parameters for CNN were used in the DeepSense architecture to illustrate the efficiency of our attention mechanism. We generated results for a

variety of window segmentation sizes to check that our models' performance was consistent.

4.4.3 Results

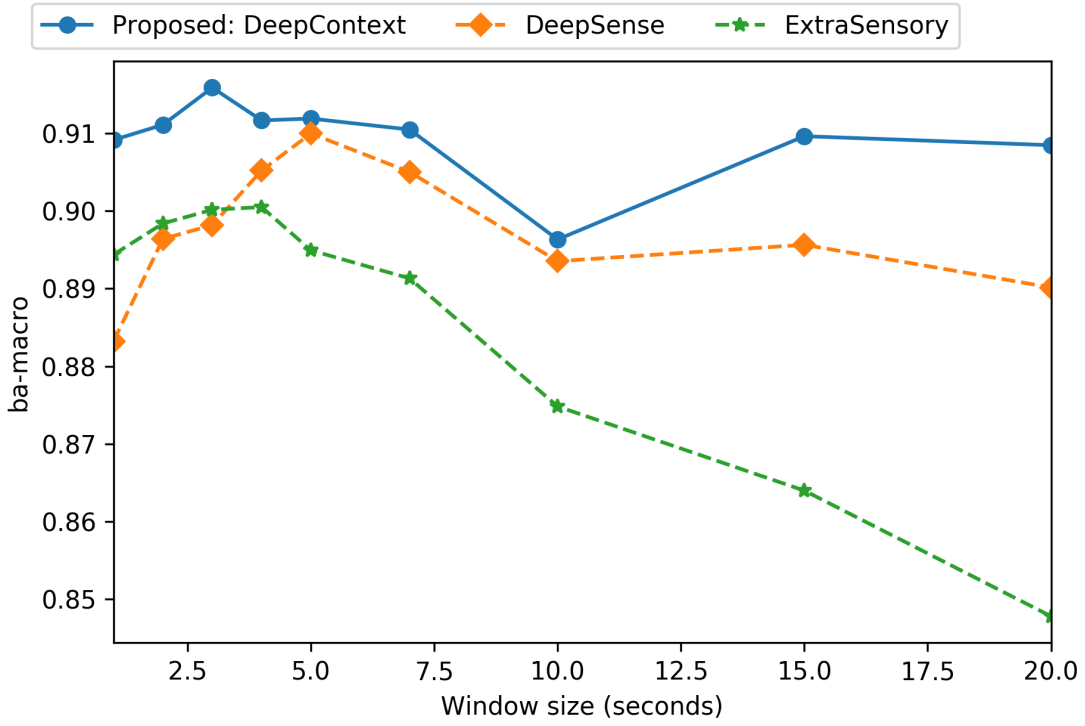


Figure 4.5: *DeepContext* performance compared to other state-of-the-art deep learning methods.

The overall performance of all evaluated models on our WPI-WASH scripted dataset can be observed in Fig (4.5), where we compare *DeepContext* to state-of-the-art methods. Additionally, results for each label are reported in Table (4.2).

In order to demonstrate the effectiveness of *DeepContext*, in Figs (4.6) and (4.7), we evaluate the improvement that can be attributed to each component separately. The two components are 1) Parameterized compatibility-based attention and 2) Joint-learning fusion to incorporate handcrafted features. Those components were

Table 4.2: Comparison of our Results with state-of-the-art methods for window size = 20 seconds.

Label	DeepSense [119]	ExtraSensory [116]	***DeepContext***
Phone in Bag	**0.8940 ± 0.020**	0.7635 ± 0.045	0.8730 ± 0.036
Phone in Hand	0.8751 ± 0.028	0.7292 ± 0.037	**0.8862 ± 0.002**
Phone in Table, Facing Down	**0.9406 ± 0.019**	0.8720 ± 0.043	0.9370 ± 0.042
Phone in Table, Facing Up	**0.9529 ± 0.012**	0.8909 ± 0.042	0.9502 ± 0.024
Phone in Pocket	0.8201 ± 0.057	0.6838 ± 0.011	**0.8409 ± 0.036**
Walking	0.9074 ± 0.027	0.8936 ± 0.022	**0.9191 ± 0.026**
Sitting	0.9101 ± 0.037	0.8718 ± 0.032	**0.9143 ± 0.025**
Jumping	0.9250 ± 0.025	0.9004 ± 0.039	**0.9396 ± 0.004**
Jogging	0.9686 ± 0.004	0.9549 ± 0.006	**0.9739 ± 0.004**
Lying Down	**0.9276 ± 0.017**	0.8879 ± 0.011	0.9040 ± 0.022
Running	0.9267 ± 0.024	0.9193 ± 0.022	**0.9586 ± 0.013**
Standing	0.8224 ± 0.011	0.8266 ± 0.022	**0.8520 ± 0.034**
Sleeping	**0.9370 ± 0.022**	0.8732 ± 0.035	0.9175 ± 0.027
Stairs - Going Up	0.7997 ± 0.048	0.8160 ± 0.010	**0.8944 ± 0.041**
Stairs - Going Down	0.7408 ± 0.072	0.7860 ± 0.035	**0.8455 ± 0.041**
Talking On Phone	**0.9499 ± 0.003**	0.8581 ± 0.022	0.9152 ± 0.001
Trembling	0.8851 ± 0.114	0.8657 ± 0.065	**0.9414 ± 0.004**
Typing	**0.9727 ± 0.020**	0.9008 ± 0.045	0.9719 ± 0.017
Bathroom	**0.9072 ± 0.035**	0.8488 ± 0.038	0.8929 ± 0.004
Average	0.89804	0.84961	**0.91197**

placed on top of our core separate-n-merge CNN architecture. A magnified view of the two proposed components can be seen in Fig (4.6) to clearly show the usefulness of each one. The two proposed components are compared against our core Separate-n-merge CNN, using the same number of layers and fine hyper-parameters. We also experimented with various ways to increases the model's performance including 1) adding an LSTM layer after the final extracted features, and 2) increasing the complexity of the model by placing residual skip links on the merged-sensors CNN (Fig. 4.8)

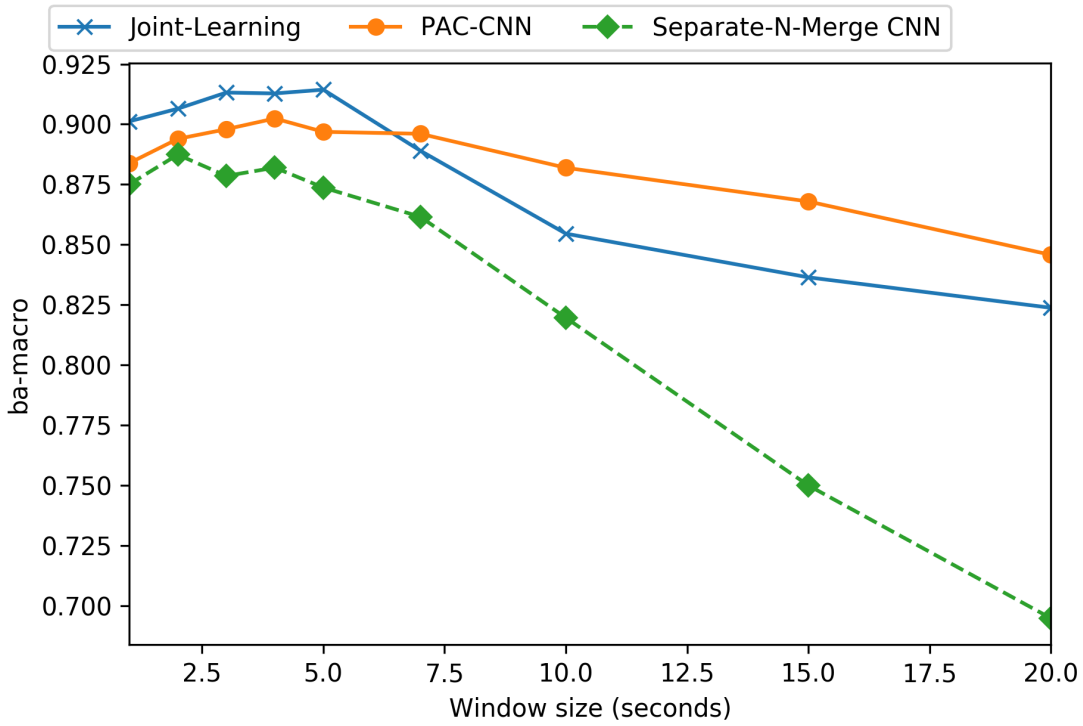


Figure 4.6: Evaluating the effectiveness of *DeepContext* components separately.

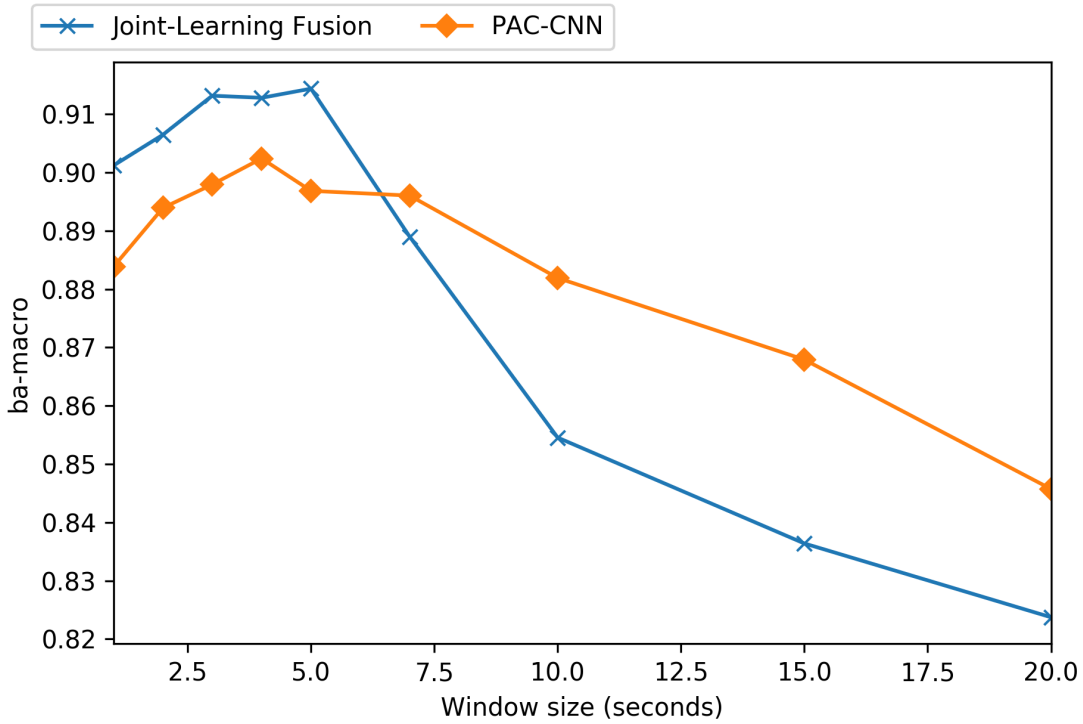


Figure 4.7: Evaluating the effectiveness of *DeepContext* components separately - magnified view.

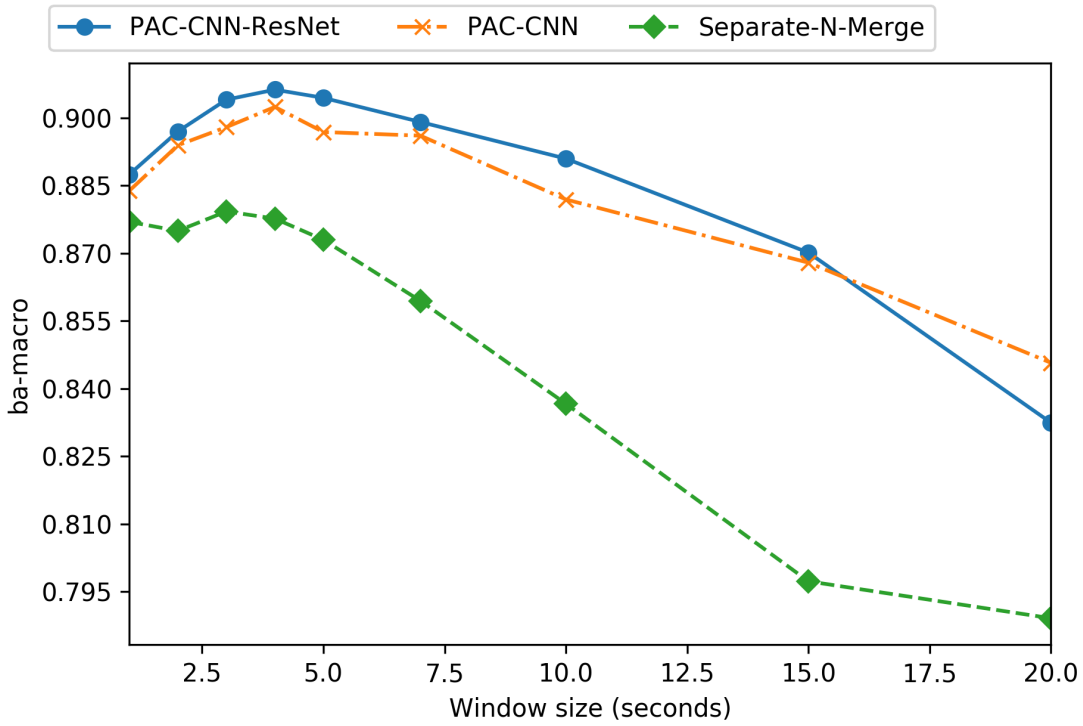


Figure 4.8: Various ways to increase *DeepContext's* performance.

4.5 Discussion

We can observe that *DeepContext* consistently outperforms the state-of-the-art approaches for both HCR (ExtraSensory) and HAR (DeepSense) especially for larger window sizes when the data captures more background noise, and the user-provided ground-truth labeling becomes more coarse-grained and less accurately associated with the entire training example. Additionally, from an application perspective, accurate predictions for larger segments are more useful, which indicates that more discriminative features have been learned regardless of the window size utilized. Intuitively, as we increase the window size, there is a greater chance for the attention mechanism to learn context-specific salient features, and more effectively suppress background noise occurring in the sensor data.

We speculate that the performance drop when using only handcrafted features

with the core Separate-N-Merge CNN classifier might be because of the difficulty of capturing useful context-specific features when the window size gets larger. In Fig (4.8), there was a slight improvement when we tweaked the *DeepContext* architecture, by adding residual skip links [125], which demonstrates the potential for achieving even better performance by using such mechanisms on sensor data. We will explore residual skip links in future work. Figure (4.6) shows the significant improvements that our proposed sub-modules achieves on top of the Separate-n-merge architecture. By looking at the detailed reported results per label, where we evaluated *DeepContext's* performance in comparison to state-of-the-art methods. *DeepContext* outperforms the other models for more than half of the labels. Even for labels where another model outperforms *DeepContext*, it's performance is very close score to that of the winning model. We speculate that this is due to the utilization of both deep learning based generated features and the domain-specific handcrafted features. One of the most challenging activities to detect, *Stairs - Going Up* and *Stairs - Going Down*, *DeepContext* is able significantly outperform the other state-of-the-arts methods. Detecting whether the subject is avoiding using stairs might provide useful insights about their mobility levels, which could facilitate the identification of potential ailments [92].

4.6 Prior Work

Numerous attention mechanisms techniques have been proposed to improve classification accuracy and providing explainability in document classification, machine translation and recently for object detection and localization in images [117].

Wang *et al.* [120] used an attention mechanism for human activity recognition from accelerometer data, addressing the same weak supervision problem as *DeepContext*, but applied it to recognizing a relatively smaller set of mutually exclusive labels. *DeepContext's* attention model is similar to that of Wang *et al.* [120] but uses a parameterized compatibility-based attention model on multi-sensor CNNs. We also propose a new way of incorporating an attention mechanism on multiple sensors by first using a separate-and-merge [119] CNN and applying attention layers on features generated by single-sensor CNNs as well as on features generated by CNNs that analyzed the merged sensor outputs. The *DeepContext* multi-sensor fusion framework is also motivated by the ability to learn cross-sensor correlations using deep-learning on multiple modalities for ubiquitous computing [121]. The two leading deep learning methods are: 1) ExtraSensory: multi-layer perceptron context recognition architecture using handcrafted features and 2) DeepSense: generic deep learning-based activity recognition model using raw sensor data. These two methods do not address the challenge of coarse-grained labeling or weakly supervised learning.

4.7 Chapter Summary

We demonstrated the applicability of *DeepContext*, a deep learning based architecture for detecting a smartphone user's current context. Using a Convolutional Neural Network (CNN) with parameterized compatibility-based attention, *DeepContext* is able extract salient discriminative features under weakly labeled scenarios. Utilizing an attention mechanism, *DeepContext* can autonomously learn

context-specific salient features, while suppressing potentially irrelevant parts of the input, tackling the issue of coarse-grained labeling that usually exists in smartphone sensor data. We have experimentally demonstrated the effectiveness of jointly learning from a combination of handcrafted features and CNN-generated features extracted from raw smartphone inertial sensor data. *DeepContext* consistently outperforms state-of-the-art methods on smartphone context sensor data gathered from 100 study participants.

Part III

Improving Transferability of In-lab models to the Real world

Chapter 5

Leveraging coincident data gathering study for Human Context Recognition under inaccurate supervision

5.1 Introduction

Our envisioned context-specific assessments require accurate recognition of specific smartphone user contexts. Existing context datasets were either *scripted* or *in-the-wild*. Scripted datasets have accurate context labels but user behaviors are not realistic. In-the-wild datasets have realistic user behaviors but often have wrong or missing labels. We leveraged our novel coincident data gathering study design in which data was gathered for the same contexts using both a scripted and in-the-wild study. The coincident study tries to combine the accuracy of the scripted labels with the realistic context visit patterns of the in-the-wild studies. For more details about the coincident study data collection, we refer the reader to the Background chapter.

Challenges. In this chapter, we address two major challenges in our in-the-wild dataset. First, the labels are extremely noisy, inaccurate and sparse. Secondly, it is challenging to devise a robust methodology for transferring knowledge from the scripted dataset to the much noisier, yet more true-to-life, in-the-wild dataset. Specifically, discovering the most likely true labels of mislabeled or unlabeled scripted data is a very challenging problem.

Prior work. To mitigate missing labels, some prior smartphone data collection methods include interfaces and mechanisms that subjects can utilize to label batches of past contexts retrospectively whenever they have free time [59]. However, as subjects often do not remember some contexts or their start/end times accurately, recall bias diminishes the quality of labels. Finally, careless subjects may also provide many wrong labels [110]. Zeni *et al.* [111] devised an interactive machine learning framework for testing user trustworthiness by checking the consistency of the user provided annotations using available ground truth. Their work required continuous feedback from the user, which is undesirable and focused on user location only.

We propose methods to mitigate the above challenges, overcome poor label quality and improve deep learning HCR models. We propose Positive Unlabeled Context Learning (PUCL) a deep learning framework that leverages the coincident context datasets. The coincident study tries to combine the accuracy of the scripted labels with the realistic context visit patterns of the in-the-wild studies. PUCL improves HCR model performance on the realistic but noisy in-the-wild data using intelligence learned from the high fidelity labeling in the scripted dataset. Specifically, *PUCL* uses a transductive positive unlabeled learning methodology to transfer

knowledge from the highly-accurate labels of the scripted dataset to the less accurate, more sparsely but yet more realistic in-the-wild dataset. Our methodology that combines coincident data collection with PUCL outperforms state-of-the-art deep learning HCR methods and is inspired by meta-learning approaches [126]. In Figure (5.1), we show a high-level overview of PUCL, in addition to the envisioned healthcare application's use case. We refer the reader to [26] for a more detailed description of our envisioned health application's use case.

5.2 Prior Work: Knowledge Transfer for Labeling Sensor Data

Several semi-supervised [127, 128] and transfer learning approaches [129, 130] have previously been proposed to tackle the issue of limited annotated sensor data. Chen *et al.* [128] utilized ensemble learning and majority voting for semi-supervised learning, using a similar feature generation mechanism to ours that uses attention to focus on salient regions in sensory inputs. Recently, an opportunistic sensor data knowledge transfer labeling mechanism was proposed, which leverages a computer-vision mechanism to label sensor-based instances. However, it requires the availability of activity recorded using a camera [130]. Additionally, the generative auto-encoder based method has been utilized for stochastic feature generation, utilized for cross-sensor classification of wearable data [129]. However, these prior works were applied on HAR datasets, containing only singly-labeled scripted activity data [129]. Our work focuses instead on phone context, which includes activity but also includes other variables such as the phone's placement.

To the best of our knowledge, ours is the first work to apply PU learning on coincident scripted and in-the-wild smartphone datasets to improve HCR performance. Our PUCL method demonstrates that data collected in laboratory settings can be used to improve the performance of classifiers designed to infer context from data gathered in the wild. PUCL does not require the use of external devices such as cameras for annotation purposes, or interactive correction of wrong labels by humans.

5.3 Positive Unlabeled (PU) Context Learning (PUCL): A Novel Learning Methodology

In Figure (5.1), we show a high-level overview of PUCL, in addition to the envisioned healthcare application's use case. We refer the reader to [26] for a more detailed description of our envisioned health application's use case.

PUCL is a novel learning methodology that has two stages and is depicted in Figure (5.2). In the first stage, we utilize a PU classifier with our correctly labeled scripted dataset to correct inaccurate labels in our in-the-wild context dataset. In the second stage, we train DeepContext, our novel deep learning architecture, on the in-the-wild dataset with labels that have been corrected by our PU method during the first stage. We now describe the two stages of PUCL in more detail.

5.3.1 Stage 1: Correcting The In-The-Wild Labels

In this stage, PUCL tries to learn reliable label-feature mappings from the more reliable scripted dataset, allowing us to discover incorrect or missing labels in the

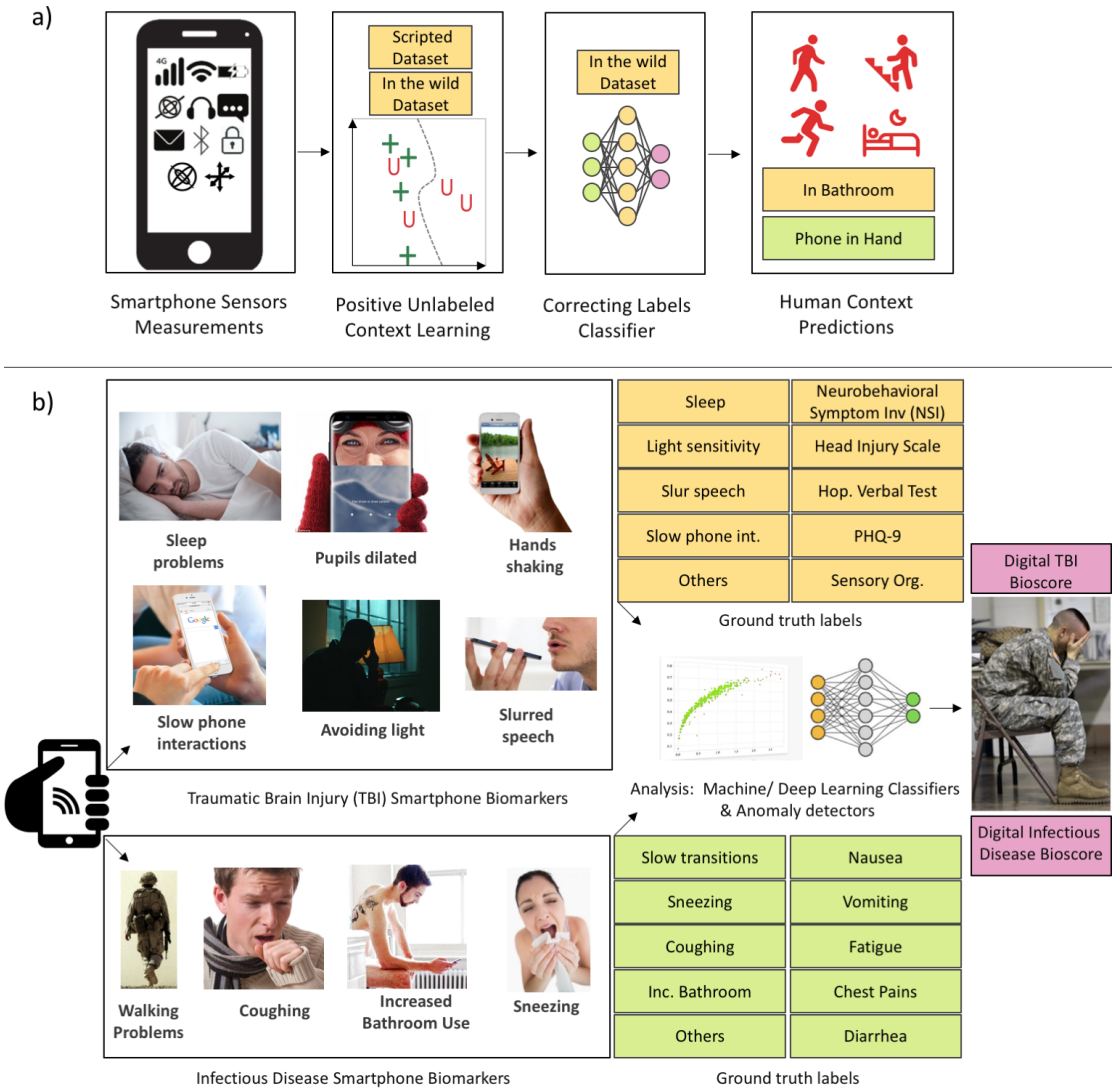


Figure 5.1: a) High-level overview of PUCL. b) Our Envisioned Future Health application, an overview of WASH-WPI's TBI Infectious Disease BioScore Synthesis.

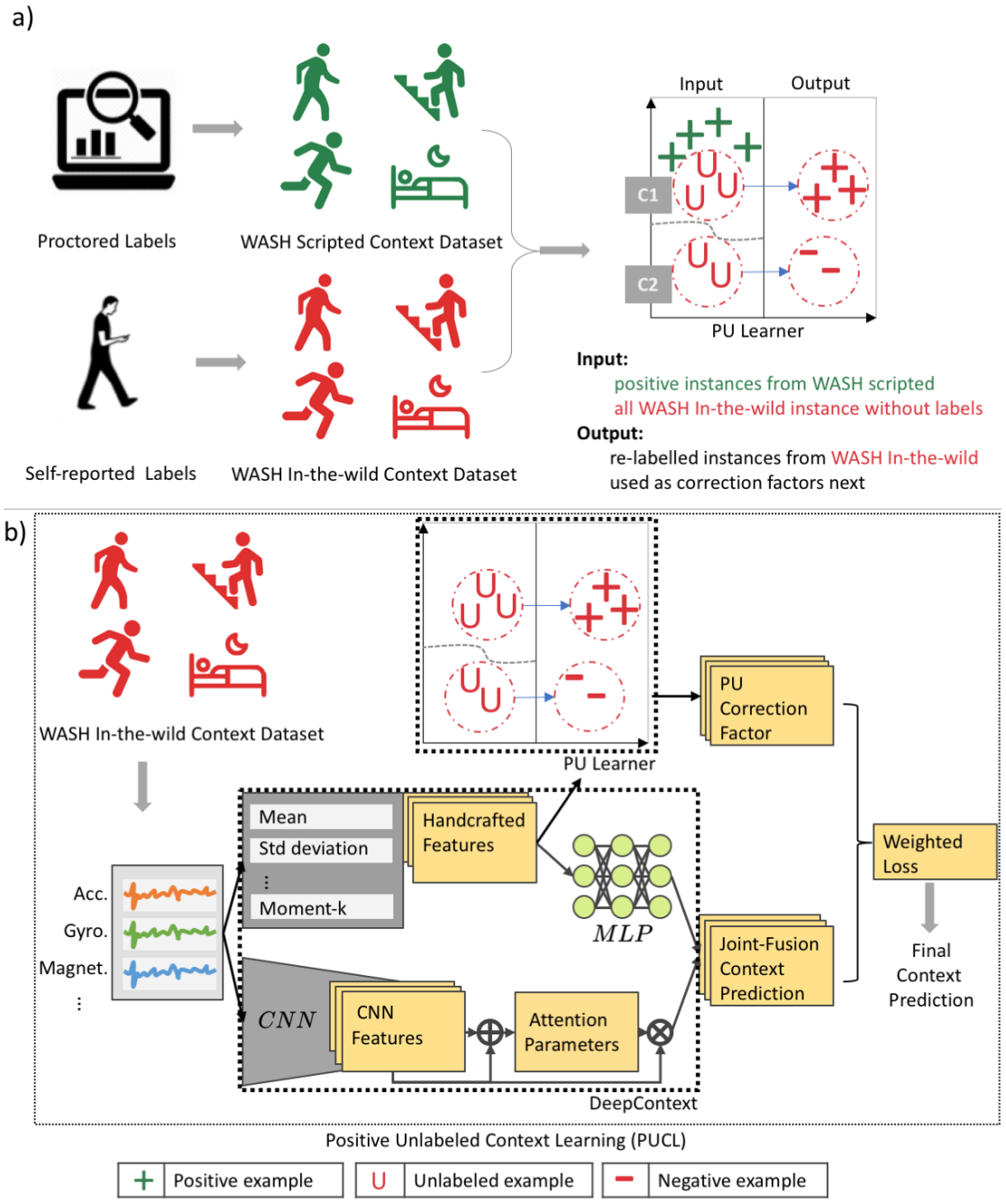


Figure 5.2: Diagram for our Positive Unlabeled Context Learning (PUCL) showing a) PU learning for coincident scripted and in-the-wild human context recognition. A PU learner is fit to identify (+) and (-) instances in the WASH In-the-wild dataset. C1 instances are relabeled as positives due to their proximity to positively-labeled instances of WASH Scripted Context Dataset in feature space2) C2 instances are relabeled as negatives due to their distance from positive instances of the WASH Scripted Context Dataset. b) DeepContext HCR model is trained using the pseudo labels generated as a correcting factor.

in-the-wild dataset. For each class Y that is present in both the scripted and in-the-wild datasets, let $\mathcal{P}$ be the positively-labeled instances of that class in the scripted dataset. Let $\mathcal{U}$ be the entirety of the in-the-wild dataset. We then train a probabilistic PU classifier f_{pu} to predict $Pr(Y = 1|x \in \mathcal{U})$.

While our approach is flexible enough to utilize any PU learning method, we use the PU Bagging algorithm as our classifier. After running the PU Bagging algorithm, all in-the-wild instances would have now been associated with a probability of belonging to the positive class. In addition to guessing the labels of unlabeled instances, our PUCL method can also correct wrongly labeled instances in the in-the-wild dataset. Positive instances that are wrongly labeled as negative can be identified because the PU bagging algorithm will assign them a score that indicates that they have a high probability of belonging to the positive class. Conversely, negative instances that are wrongly labeled as positive will have a score assigned by the PU bagging algorithm, which indicates that they have a low probability of belonging to the positive class.

Building on this intuition, we formulate and estimate a *correcting factor* that corresponds to how much an assigned label in the in-the-wild dataset should be trusted. For each class, e.g. "walking", let y_i be the label of that class associated with the ith instance in the in-the-wild dataset and let PU_i be the PU Bagging score for the class associated with that instance. Then, the correcting factor CF_i is given as:

$$CF_i = 1 - |y_i - PU_i|$$

If PU_i is large while $y_i = 0$, or if PU_i is small while $y_i = 1$, then CF_i will be close to 0. This means that the label associated with the ith instance should not be trusted.

5.3.2 Stage 2: Context Recognition using DeepContext

The goal of this stage is to train a robust context classifier, *DeepContext* [22], a novel deep learning based architecture for multi-label recognition of a smartphone user's current context. Utilizing an attention mechanism, *DeepContext* is able to autonomously learn salient features that discriminate contexts, while suppressing potentially irrelevant parts of the input.

We adapt *DeepContext*, our proposed context classification model but additionally we utilize PUCL to mitigate the negative impact of the inaccurate and missing labels in the in-the-wild dataset. Specifically, *DeepContext* uses the correcting factor in stage 1 to improve its classification results on the in-the-wild dataset. *DeepContext* takes as input both handcrafted-features generated using domain knowledge as well as the raw-sensor values collected by the smartphone. Furthermore, DeepContext utilizes state-of-the-art attention mechanisms to focus on sub-components of the input data that are most predictive of each target class.

Classification accuracy is boosted as the noise present in each input is ignored. In particular, DeepContext is trained using gradient descent on its parameters (denoted as Θ) on the inexact and weakly labeled in-the-wild data by minimizing the following cost function:

Table 5.1: Comparison of our Results with state-of-the-art methods.

a) Results overall

Model	BA	Recall	Precision	Specificity	F1-score
ExtraSensory MLP	0.780161	**0.781946**	0.339242	0.778377	0.472944
PU Context Learning (PUCL)	**0.813777**	0.713059	**0.551373**	**0.914494**	**0.621843**

b) Results per label

Label	ExtraSensory	***PU Context Learning***
Phone on Table, Facing Down	0.8416 ± 0.014	**0.8707 ± 0.011**
Stairs - Going Down	0.8051 ± 0.012	**0.8429 ± 0.011**
Sleeping	0.8294 ± 0.002	**0.8419 ± 0.005**
Stairs - Going Up	0.8141 ± 0.002	**0.8414 ± 0.005**
Laying Down, Phone on Table	0.7913 ± 0.018	**0.8204 ± 0.010**
Phone in Bag	0.7924 ± 0.018	**0.8024 ± 0.010**
Phone in Pocket	0.7878 ± 0.014	**0.7994 ± 0.016**
Typing	0.7736 ± 0.033	**0.7945 ± 0.008**
Walking, Phone in Bag	0.7763 ± 0.010	**0.7910 ± 0.007**
Walking, Phone in Pocket	0.7740 ± 0.008	**0.7895 ± 0.010**
Phone one Table, Facing Up	0.7563 ± 0.011	**0.7888 ± 0.015**
Walking	0.7602 ± 0.003	**0.7796 ± 0.003**
Exercising, Phone in Pocket	0.7532 ± 0.025	**0.7711 ± 0.012**
Laying Down	0.7410 ± 0.021	**0.7701 ± 0.009**
Walking, Phone in Hand	0.7519 ± 0.006	**0.7654 ± 0.010**
Sitting	0.7408 ± 0.008	**0.7577 ± 0.018**
Running	**0.7551 ± 0.058**	0.7539 ± 0.029
Phone in Hand	0.7436 ± 0.009	**0.7512 ± 0.016**
Bathroom, Phone in Pocket	0.7246 ± 0.019	**0.7476 ± 0.010**
Jogging	**0.7720 ± 0.145**	0.7437 ± 0.175
Exercising	0.7185 ± 0.037	**0.7429 ± 0.008**
Jumping	0.7176 ± 0.004	**0.7260 ± 0.002**
Standing	0.7037 ± 0.010	**0.7256 ± 0.002**
Bathroom	0.6614 ± 0.024	**0.6957 ± 0.004**

$$C(\Theta) = \frac{1}{N}\sum_{i=1}^{N} \ell_{\Psi}(f(X_i), Y_i),$$

where N is the number of training samples and ℓ_{Ψ} is the loss function that is weighted by the correcting-factor. More specifically,

$$\ell_{\Psi} = \sum_{y \in Y}\sum_{i=1}^{N}(\frac{1}{Pr(y)} + CF_{i,y}) \cdot [y_i log(f(x_i) + (1 - y_i) log(i - f(x_i))].$$

More intuitively, ℓ_{Ψ} is simply the cross entropy loss (or any other deep learning loss function) multiplied by a weighting factor. The weighting factor is a combination of the inverse class frequency with the correcting factor. In order to account for class imbalance, the weighting factor weights instances of infrequent classes higher than instances of frequent classes and discounts the cost incurred from instances that are likely to have been mislabeled by the annotator. This discounting is applied so as not to punish the network for disagreeing with annotator-assigned labels that are probably wrong.

5.3.3 Context Recognition Results

We compared our model performance against state of the art for HCR (ExtraSensory MLP [59]), which has been applied for a very similar dataset to ours. Due to class imbalance in our context datasets, we utilize *Balanced Accuracy* (BA) as

the metric to evaluate the context recognition performance of *DeepContext* and our novel PUCL method. BA is defined as:

$$BA(\mathcal{D}) = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right)$$

which is also:

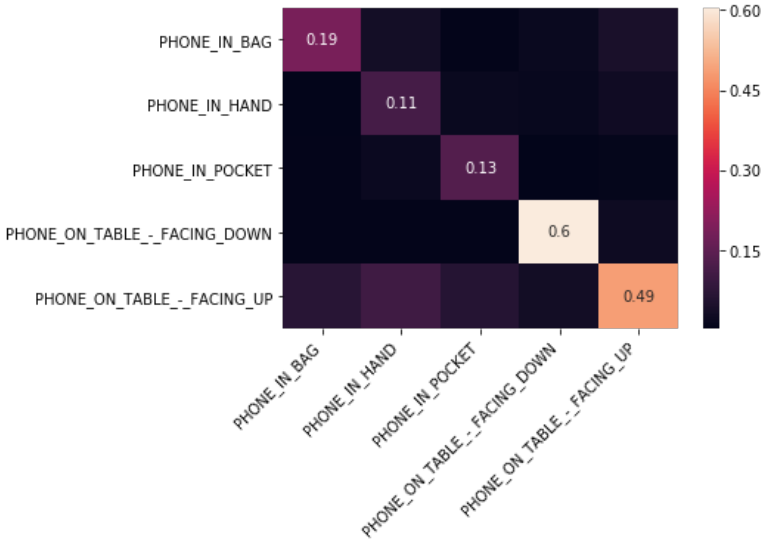
$$BA(\mathcal{D}) = \frac{1}{2}\left(Sensitivity + Specificity\right)$$

Our results are shown in Table (5.1)(a) demonstrate that our approach outperforms the state-of-the-art model on all metrics except recall with lower false positive rates. We speculate that the drop in recall may be due to the amount of mislabeled annotated true positives. Intuitively, the PU Correcting Factor will put less attention on instances that are most likely mislabeled. It would then be expected that the true positive instances will be classified with a higher consistency using guided knowledge gathered from the scripted dataset. Table (5.1)(b) presents performance per label, showing that our approach consistently outperforms state-of-the-art methods across all labels, except Running and Jogging. As can be seen in the results, Running and Jogging have the highest variability scores among user splits. The poor performance in Running and Jogging can be a result of the increased noise resulting from such activities. Figure (5.3) - (a) and (b) we compare confusion matrices showing that our approach achieves consistent improvements over other state-of-the-art methods in detecting phone prioception (placement). Lastly, in Figure (5.3) - (c), we evaluated the impact of the proposed PUCL mechanism on both DeepContext and ExtraSensory MLP. We show that utilizing the PU Correcting factor during training, we achieved a significant

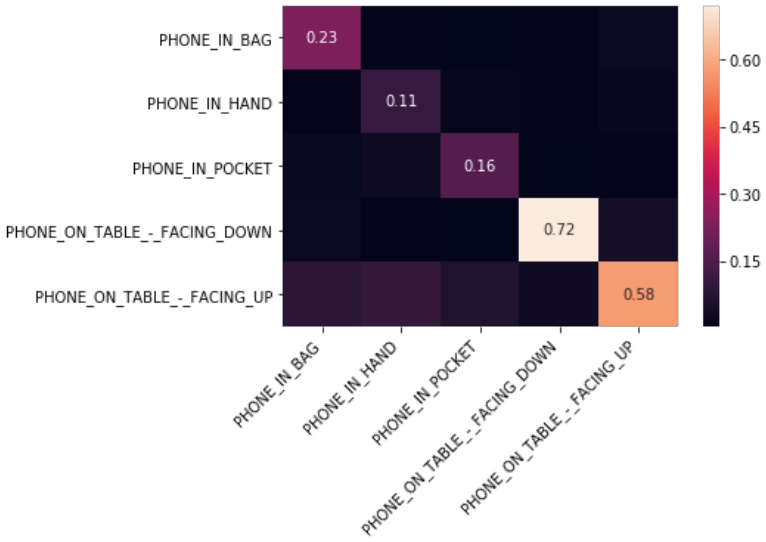
increase in the Balanced Accuracy (BA) of classification in our evaluation of both learning models: ExtraSensory MLP and our proposed model DeepContext.

5.4 Chapter Summary

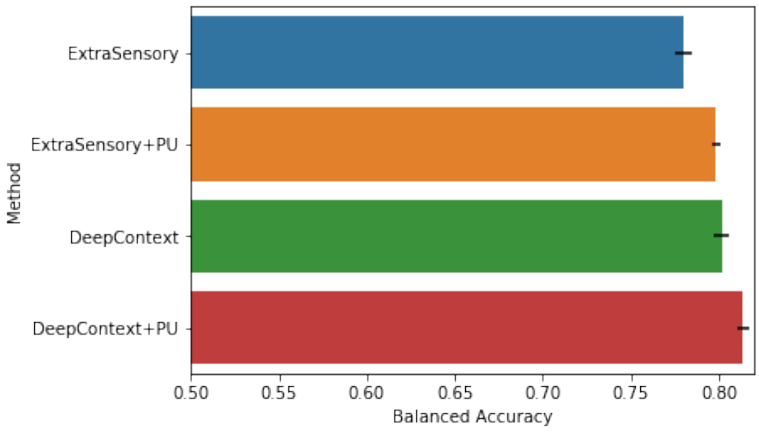
Several issues reduce the performance of machine learning HCR models on in-the-wild datasets, including weak, noisy, or missing labels. We leveraged our coincident context gathering study in designing context recognition models under the weakly supervised learning settings: inaccurate supervision. We introduced a novel PUCL approach for applying transductive PU learning on coincident scripted and in-the-wild human context recognition datasets.



(a) Extrasensory MLP



(b) PU Context Learning (PUCL)



(c) Evaluating the contribution of the PU Correcting Factor

Figure 5.3: (a) and (b) shows confusion matrices for phone prioception, with normalized scores. In (c): the impact of PU correcting factor on the used learning method is depicted.

Chapter 6

Adapting models for Human Context Recognition In the wild under incomplete supervision

We proposed Triplet-based Domain Adaptation for context REcognition (*Triple-DARE*), a deep learning method that is able to leverage the tremendous amounts of unlabeled in-the-wild data, decreasing the need for human-annotated labels. We also utilized coincident scripted and in-the-wild HCR datasets in which similar context labels were gathered in both studies (Chapter 2). These coincident datasets ensure that there exists a feature representation of contexts that is common between the scripted and in-the-wild datasets, a key requirement for this approach.

6.1 Introduction

Lab-to-field methods have recently emerged as viable solutions to achieve good HCR performance on in-the-wild datasets that have noisy, low-quality labels [37]. Lab-to-field approaches try to train highly accurate machine learning models on

scripted datasets, which are adapted for *in-the-wild* datasets with the hope of maintaining good performance. However, in general, the performance of HCR models trained naively on scripted datasets often drops when tested on in-the-wild datasets *(Going from Lab-to-field)*. This performance drop is because in addition to the in-the-wild dataset issues listed above, the contexts visited by subjects in the scripted study, as well as their context visit order and visit duration differ significantly from in-the-wild scenarios. Consequently, there are significant differences between the distribution of features extracted from scripted vs. in-the-wild datasets, also known as the *covariate shift problem* [35, 36, 37].

Domain Adaptation (DA) is a transductive transfer learning method and one of the main solutions used to adapt neural networks to mitigate the covariate shift problem. DA has been employed in various related domains, including object detection in computer vision and the problems caused by the variability of wearable sensor placement in ubiquitous computing[36, 24]. Unsupervised DA (UDA) tries to learn a deep learning model using a combination of a labeled source (e.g. scripted dataset) and unlabeled target (e.g. in-the-wild) samples with different distributions to achieve accurate predictions on previously unseen, unlabeled (e.g. in-the-wild) samples [107, 24]. Figure (6.3) provides a high-level overview of the problem, challenges, and our approach.

Challenges. Two key challenges must be addressed for using UDA for Lab-to-Field generalization of smartphone context recognition. First, the previously described data issues with in-the-wild datasets (the diversity of placements, weak and noisy labels, and diverse smartphone types - Chapter 2) must be overcome. Secondly, it is challenging to develop a robust method for transferring knowledge

from a scripted dataset to a more realistic but considerably noisier in-the-wild dataset with sparse labels.

State-of-the-art limitations. There is very little work on lab-to-field generalization for HCR. Previously proposed lab-to-field methods include importance re-weighting [37, 131] and Positive Unlabeled (PU) classifiers [132]. DA has previously been used to address the issue of variability in the placement of wearable sensors [24, 103] but not for HCR. The majority of prior DA work for wearable sensors focuses on reducing the global distribution discrepancy across domains while learning common feature representations [103, 24]. However, we observe that even if the global distribution is effectively aligned, samples from different domains with the same label may be mapped widely apart in feature space. Thus, in addition to using a domain alignment loss [100, 101], *Triple-DARE* improves intra-class compactness and inter-class separability by utilizing a joint fusion triplet loss [107, 108] designed for multi-labeled datasets. Moreover, unlike other existing methods for dealing with domain shifts [37, 103, 132, 133], we do not utilize target labels in the target (in-the-wild) dataset, instead following the UDA problem setting in [24].

Our approach. We are motivated by the recent empirical success of triplet loss in face identification [108, 134], where variations of the same person's face images are mapped closely in the learned embedding space. We believe sensor data can benefit from the same approach where there is often variation in sensor signatures corresponding to the same context. Our belief is also consistent with Khaertdinov *et al.*'s findings where triplet loss was applied recently to mitigate the effects of subject heterogeneity and improve model generalizability [135].

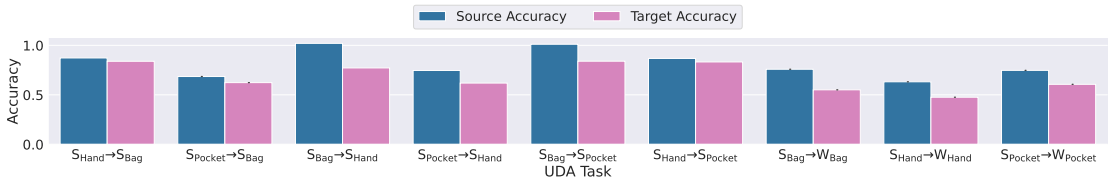


Figure 6.1: This figure demonstrates the reduction in the accuracy of predicting context for models trained solely on context labels from a single subset of our context dataset. $S_{Prioception}$ is denoted for the scripted context dataset and $W_{Prioception}$ for the in-the-wild dataset, e.g. S_{Bag} refers to scripted contexts, annotated with "Phone In Bag".

We propose *Triple-DARE*, a deep Lab-to-field UDA method that is able to leverage the tremendous amounts of unlabeled in-the-wild smartphone HCR data, decreasing the need for human-annotated labels. To facilitate our DA approach, we utilized coincident scripted and in-the-wild HCR datasets in which similar context labels were gathered in both studies [132]. These coincident datasets and similar context labels ensure that there exists a feature representation of contexts that is common between the scripted and in-the-wild datasets, a key requirement for the DA approach. We demonstrate our method's applicability to HCR models deployed in realistic environments by using context labels gathered in a scripted study only during model development and using DA to mitigate the influence of potentially noisy labels and retain HCR performance on an in-the-wild dataset. *Triple-DARE* outperforms state-of-the-art baselines with 3.79% and 1.89% increases in F1-score and classification accuracy, respectively, and also achieves improvements of 39% and 14.7% in F1-score and classification accuracy, respectively, HCR models without *Triple-DARE*.

Contributions. The main contributions are:

1. We proposed *Triple-DARE*, a novel UDA deep-learning framework, which utilizes a domain alignment loss to learn domain-invariant features, a clas-

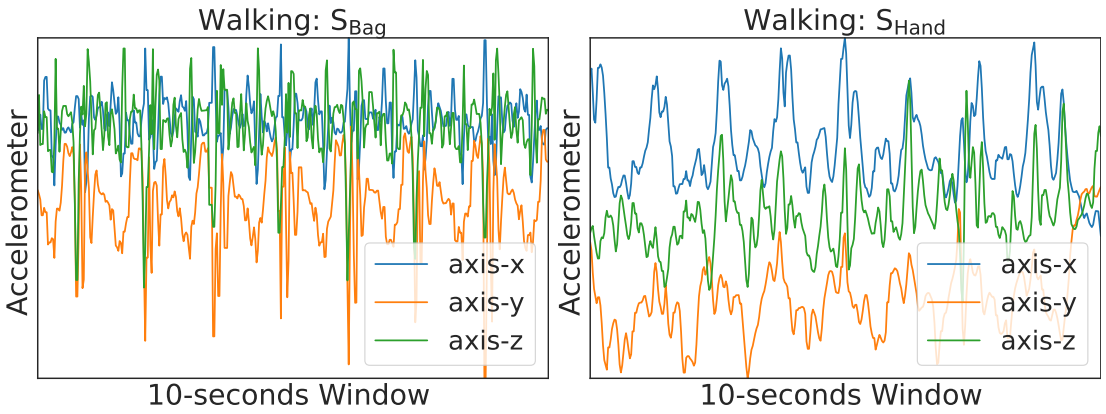


Figure 6.2: The influence of diverse phone placements on sensor data is observable in triaxial accelerometer tracings for the same walking activity but with different prioceptions.

sification loss to maintain task-discriminative features, and a joint fusion triplet loss to increase intra-class compactness and inter-class separation. A scripted dataset is used to improve the HCR accuracy of predicting in-the-wild contexts.

2. We rigorously evaluated *Triple-DARE*, comparing it to multiple state-of-the-art unsupervised domain methods, including DAN [100], CORAL [101], and HDCNN [103], and bench-marking improvements in HCR performance on target domains in several use cases.

3. We demonstrate that *Triple-DARE* mitigates in-the-wild dataset challenges when compared to state-of-the-art DA methods, achieving high prediction scores on the target domain without the need for large amounts of source labeled samples. Our ablation study demonstrates that all component of *Triple-DARE* contributes non-trivially.

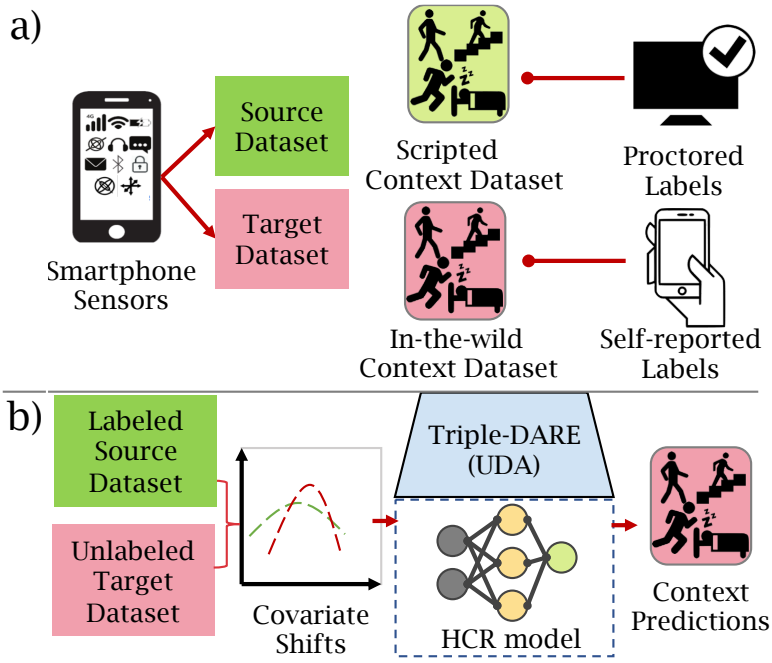


Figure 6.3: a) The nature of the two smartphone context data we use in this work. b) A high-level overview for *Triple-DARE*'s problem and approach.

6.2 Related Work

6.2.1 Lab-to-field Generalization

Our Lab-to-field method tries to leverage a scripted dataset with high-quality, relatively cheaper to obtain, ground truth labels to improve HCR model performance on an in-the-wild dataset [37]. Lab-to-field methods previously proposed to handle covariate shifts include importance re-weighting [37, 131] and Positive Unlabeled (PU) classifiers [132]. There is very little work on lab-to-field generalization for HCR. However, a related work that dealt with this problem on wearable electrocardiogram (ECG) data [37], used importance re-weighting to adapt a linear logistic regression model. However, these methods have achieved lower performance when applied to deep neural networks [136]. Unlike other existing methods for dealing with domain shifts [37, 103, 132, 133], our approach does not require

target domain labels.

6.3 Proposed *Triple-DARE* Methodology

6.3.1 Problem Formulation

In this work, we utilize data from two context datasets: 1) a scripted dataset (source) with high-quality labels and 2) an in-the-wild dataset (target) in which data was annotated with the same context ⟨Activity, Phone Prioception⟩ labels shown in Table 6.2. With regards to UDA, there are labeled samples for the source domain and unlabeled samples for the target domain, which have different data distributions. Our goal is to learn a classifier that generalizes well on the target domain, using labeled source data and unlabeled target data. Formally, we have labeled samples $\mathscr{D}_s = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^{n_s}$ and unlabeled samples $\mathscr{D}_t = \{(\mathbf{x}_i^t)\}_{i=1}^{n_t}$ where n_s and n_t represent the number of samples in source and target domains, respectively. Both the source and the target domain share the same feature space $\mathcal{X}_s = \mathcal{X}_t$ and label space $\mathcal{Y}_s = \mathcal{Y}_t$, but they differ in the marginal distribution ($P_s(x_s) \neq P_t(x_t)$) and the conditional distributions are presumed to be equal $P_s(y_t|x_s) = P_t(y_t|x_t)$. We denote $\boldsymbol{x}$ as a feature vector and $\boldsymbol{y}$ as a human context represented by a multi-label output vector, where each label produced is a binary output (E.g walking vs not walking). The source and target tasks are presumed to be the same. Initially, we train the HCR model using the labeled source dataset. Afterward, the trained HCR model is utilized to recognize unlabeled contexts in the target dataset by incorporating unlabeled data from the target dataset.

6.3.2 Overview

The framework of *Triple-DARE* is illustrated in Figure (6.4). *Triple-DARE* has two types of feature sources extracted from both the source scripted and target in-the-wild datasets: 1) Time and frequency based handcrafted features handled by a feed-forward network and 2) raw three axial sensors fed into a CNN that extracts salient features from raw sensor data using a soft attention mechanism. *Triple-DARE* has three major learning components: 1) A domain alignment loss $\mathcal{L}_{\text{d}}$ to extract embeddings that are invariant across domains. 2) a classification loss $\mathcal{L}_{\text{cls}}$ to maintain task-discriminative features, 3) a joint fusion triplet loss $\mathcal{L}_{\text{tri}}$ to increase intra-class compactness and inter-class separation in the learned embedding space by learning similar contexts represented by variations of sensor inputs. The final output is used for multi-labeled context predictions. For example, based on our definition of context as an <Activity, Phone placement>, a context could be ⟨"Sitting","In Bathroom" with "Phone In Hand"⟩. In order to perform our context predictions by learning discriminative and domain-invariant embeddings, our final objective is to minimize the cost function $C(\cdot)$

$$C(\theta) = \lambda_1 \mathcal{L}^{\theta}_{\text{cls}} + \lambda_2 \mathcal{L}^{\theta}_{\text{d}} + \lambda_3 \mathcal{L}^{\theta}_{\text{tri}}, \tag{6.1}$$

where θ are model parameters, λ_1, λ_2, and λ_3 are balancing coefficients. This process and each type of loss are described in more detail in subsequent subsections.

6.3.3 Feature Generation

For a given smartphone context dataset, we create two views from the raw sensory inputs. The first one is a vector obtained by applying handcrafted features on all available sensors. These handcrafted features are used to construct a vector fed into a feed-forward network. A total of 188 features adopted from [14], were utilized with some examples provided in Table (7.2). The second view consists merely of the raw three-axial sensors. We use different feature encoders for each type of input view. Specifically, 1) Multi-Layer Perceptron (MLP) encoder for handcrafted features, which is adapted from [23], 2) attention-based CNN encoder [22] for raw sensor data. Finally, a joint fusion encoding is obtained by concatenating the two produced feature encodings.

The raw sensor data from three axial sensors (accelerometer, gyroscope, and magnetometer) is utilized for CNN's auto-learning features. Adapted from the DeepContext architecture [22], the CNN we leveraged has a soft attention mechanism that helps learn salient features, giving higher weights (importance) to regions of the raw sensor data that are more predictive of the user's context. The intuition behind the design of this attention mechanism is similar to that proposed by [117] and [22]. The effectiveness of this architecture comes from applying attention layers on features generated by single-sensor CNNs and on features generated by CNNs that analyzed the merged sensor outputs. This enables the model to highlight discriminative CNN features for different contexts. For more details about the DeepContext CNN architecture, we refer the reader to [22].

Table 6.1: A sample list of handcrafted features used for our sensor data, adopted from [84, 14].

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	$\mathbb{E}\left[(s-\bar{s})^4\right] / \mathbb{E}\left[(s-\bar{s})^2\right]^2$
Signal magnitude area	$\frac{1}{3}\sum_{i=1}^{3}\sum_{j=1}^{N}\lvert s_{i,j}\rvert$
Pearson Correlation coefficient	$C_{1,2}/\sqrt{C_{1,1}C_{2,2}}, C = \mathrm{cov}(s_1, s_2)$
Spectral energy of a frequency band [a, b]	$\frac{1}{a-b+1}\sum_{i=a}^{b} s_i^2$
s: signal vector, N: signal vector length Q: quartile, cov: covariance	
GPS Features	
Significant changes from the previous location state	
Estimated speed	
Changes in latitude and longitude	
Phone State Features	
Is phone screen unlocked?	Is battery charging?
Is ringer mode set to silent?	Is phone connected to WIFI?

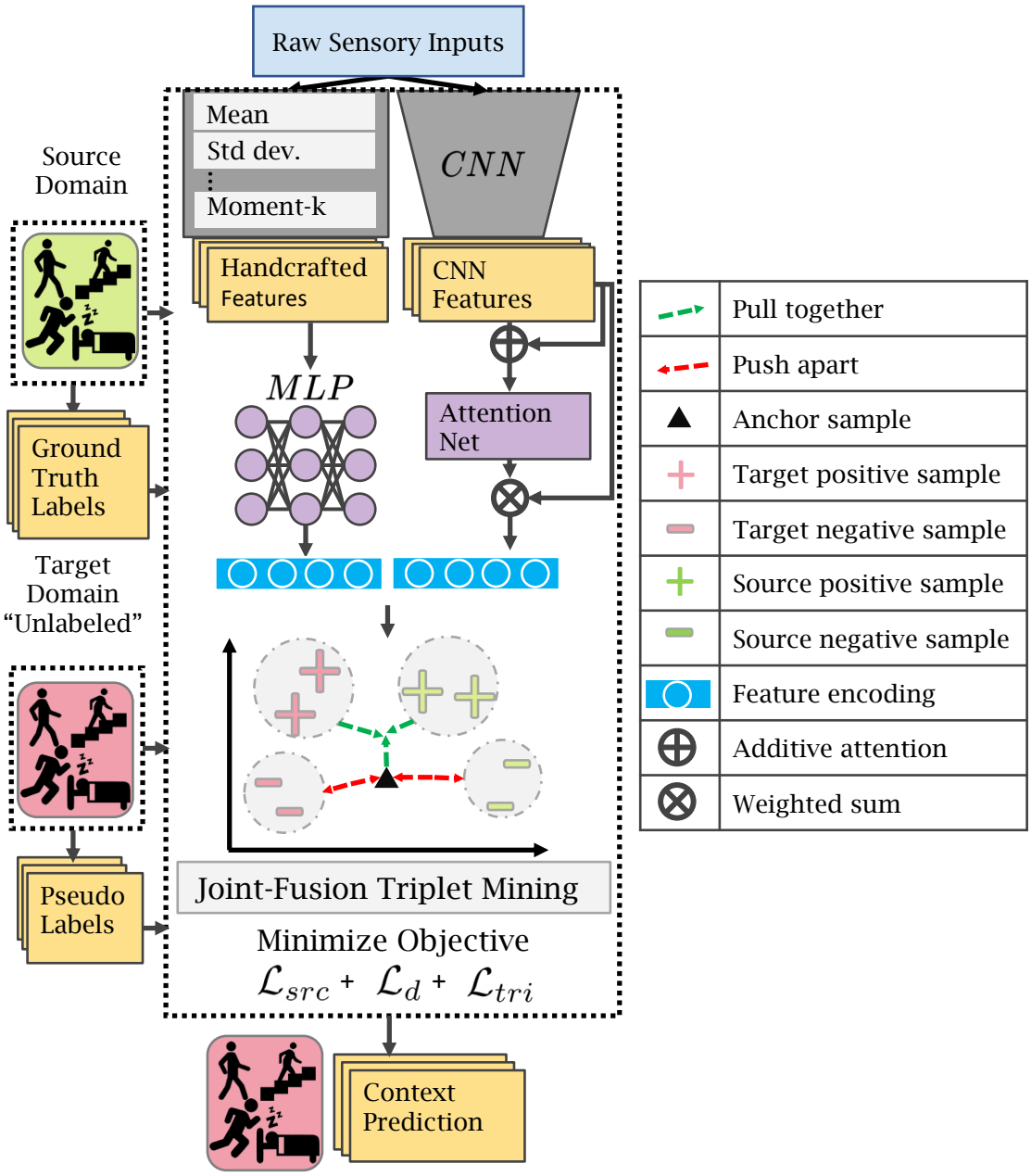


Figure 6.4: *Triple-DARE*'s Framework.

6.3.4 Domain Alignment Loss

The goal of the domain alignment loss is to map the source and target feature encoding into a standard feature distribution space to learn common feature representations across domains. For our approach, we utilize Multi Kernel Maximum Discrepancy Mean (MK-MMD), an extension of the Maximum Discrepancy Mean (MMD), introduced by Gretton *et al.* [137]. MMD is a non-parametric distance measure that may be used to assess the discrepancy between marginal distributions [100]. 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 [100].

Let $\phi(\cdot)$ be a feature map defined as a combination of G positive kernels k_u with their associated bandwidth $\beta_u \geqslant 0$, given as the following:

$$k = \sum_{u=1}^{G} \beta_u k_u, \tag{6.2}$$

$$\phi\left(x^s, x^t\right) = k\left(x^s, x^t\right), \tag{6.3}$$

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

$$\mathrm{q}(\mathcal{X}^s, \mathcal{X}^t) = \left\| E_{\mathcal{X}^s}\left[\phi\left(x^s\right)\right] - E_{\mathcal{X}^t}\left[\phi\left(x^t\right)\right] \right\|_H, \tag{6.4}$$

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

$$\mathcal{L}_{\mathrm{d}}^{\theta} = \sum_{l \in N^l} \mathrm{q}^2 \left(\mathcal{X}_l^s, \mathcal{X}_l^t \right). \tag{6.5}$$

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 lth layer in the network. $d\left(\mathcal{X}_s^l, \mathcal{X}_t^l\right)$ is the MK-MMD calculated by Equation (6.4) between the source and target domain distributions evaluated on the $\underline{l}$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.

6.3.5 Classification Loss

The classification loss aims to leverage source domain labels in discovering discriminative features for context predictions. The context labels for classification are the same in both domains. Optimizing our model for classifying contexts on the source domain guides the overall learning process. Since the labels of D_s are available, the classification loss is defined as:

$$\mathcal{L}_{\mathrm{cls}}^{\theta} = \frac{1}{N_s} \sum_{i=1}^{N_s} \ell_{\Psi}(f_{\phi}(x_i^s), y_i^s), \tag{6.6}$$

where $f_{\phi}(\cdot)$ is a classifier, N_s is the number of labeled training samples and ℓ_{Ψ} is a binary cross-entropy function weighted by inverse class frequency to account for class imbalance where infrequently occurring classes get higher weights than

frequently occurring classes and $(xi^s), y^s)$ represents labeled context data sampled from source domain data.

6.3.6 Triplet Loss

The triplet loss is mainly used to pull samples belonging to the same or similar classes together and push away samples belonging to different classes in an embedding space. It achieved empirical success in face identification, where variations of the same person images are mapped closely in the learned embedding space [108, 134]. We believe sensor data can benefit from the same approach as numerous variations in the sensor inputs can represent the same context. Given three types of samples: 1) an anchor sample x_a (i.e. a query sample), 2) a positive sample x_p (i.e., a sample shares the same class as the anchor), and 3) a negative sample x_n (i.e., a sample with a different class from the anchor). Along with a distance function d, triplet loss is defined as the following:

$$\mathcal{L}_{\text{tri}}^{\theta} = \sum_{i}^{N} \left[d(x_a^i, x_p^i) - d(x_a^i, x_n^i) + \alpha \right]_+ \tag{6.7}$$

Where α is a parameter for the margin between positive and negative samples, and x here is used to represent an embedding of x for notation simplicity. We reduce the triplet loss by pushing $d(x_i^a, x_i^p)$ towards zero and making $d(x_i^a, x_i^n)$ to be greater than $d(x_i^a, x_i^p) + \alpha$. In other words, pairs of positive samples are jointly pulled together, while pairs of positive and negative samples are pushed away by some margin α.

6.3.7 Joint-Fusion Triplet Mining

Triplet mining is the process of constructing triplets (anchor, positive and negative) for triplet loss calculations.

Following the practice in [108], we adopt the online triplet mining strategy which does not require a complete pass on the training set beforehand. Since finding triplets across two domains mandates the existence of target domain labels, the classifier trained on the source domain is used to construct pseudo labels for target domain samples during training of the classifier, which is one of the most common solutions for UDA problems [107]. During this procedure, it is vital to remember that the pseudo labels generated may not be accurate. However, we re-assign pseudo labels every few iterations because the classifier will steadily increase its accuracy on the target dataset during training. Additionally, the domain alignment loss can also help improve the classifier's accuracy on the target dataset by lowering the distribution disparity. As a result, the quality of the pseudo label can automatically improve.

Our joint-fusion triplet mining strategy works as the following: We create triplets from two mini-batches of samples from the source and target domains after concatenating them into one mini-batch. For constructing triplets suitable to our multi-labeling settings, we need a notion of similarity between multi-labeled vectors. We first define a compatibility score between two contexts y_1, y_2 that are both represented as binary labels, as the dot product between them:

$$c(y_1, y_2) = y_1 \cdot y_2 \tag{6.8}$$

Because our dataset is overly imbalanced, we consider all the positive examples in constructing the triplets. We follow a similar strategy to [108] that focuses on triplets contributing the most to the learning process but modified using our compatibility score to select triplets that satisfy this condition:

$$d(x_a, x_p) + \alpha > d(x_a, x_n) \;\; \& \;\; c(y_a, y_p) > c(y_a, y_n) \tag{6.9}$$

Our triplet mining strategy is detailed in Algorithm (1).

Algorithm 1: Joint-fusion Online Triplet Mining finds triplets with multi-labeled vectors.

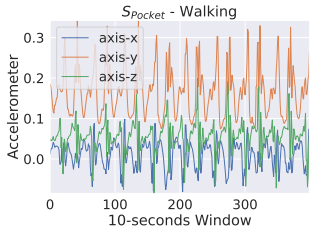
Input: Number of samples in a batch m, classifier $f(\cdot)$, D_s, D_t , distance d, compatibility score (Eq (6.8)) c, α, and sample size k
Output: List of triplets [(a, p, n)]
$\{x_i^s, y_i^s\}_{i=1}^m \leftarrow$ Read next mini-batch(D_s) ;
$\{x_i^t\}_{i=1}^m \leftarrow$ Random sample mini-batch(D_t) ;
$\{y_i^t\}_{i=1}^m \leftarrow$ 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**
      Eq (6.9):
      if $d(a, p) + \alpha > d(a, n) \;\; \& \;\; c(a, p) > c(a, n)$ **then**
        $triplets$.add ((a, p, n)) ;
      end
    end
  end
end
return $triplets$

Table 6.2: The percentage of positively labeled contexts.

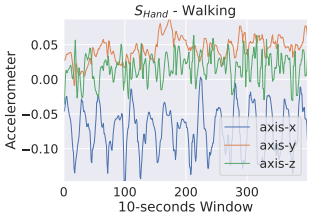
Contexts	**Scripted % P**	**In-the-wild % P**
Bathroom	3.15%	2.17%
Jogging	2.04%	0.27%
Lying Down	1.10%	16.24%
Running	1.95%	0.37%
Sitting	11.99%	38.71%
Sleeping	2.19%	37.69%
Stairs - Going Down	2.52%	2.00%
Stairs - Going Up	0.89%	1.92%
Standing	1.71%	8.46%
Talking On Phone	1.41%	1.27%
Typing	3.65%	6.45%
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_{Bag} refers to scripted contexts, annotated with "Phone In Bag"		

6.4 Experiments

We evaluated Triple-DARE and baseline models for performing multiple UDA use cases on scripted and in-the-wild smartphone HCR datasets. The overarching goal was to use *Triple-DARE* to learn a robust representation from the scripted dataset (source), which is then used to improve HCR on the in-the-wild dataset (target). Figure 6.5 displays data extracted from the two datasets, displaying only the accelerometer sensor readings for three context examples.

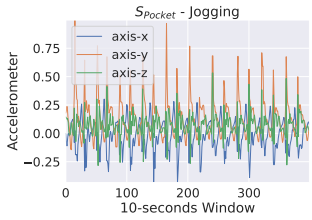


(a) S_{Pocket} - Walking

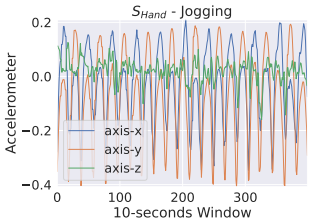


(b) S_{Hand} - Walking

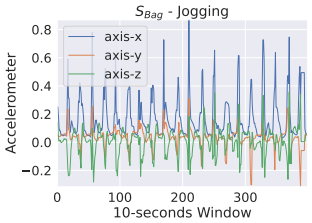
(c) S_{Bag} - Walking



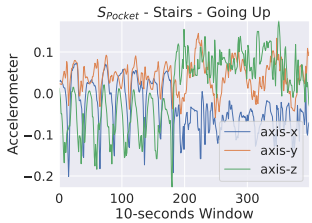
(d) S_{Pocket} - Jogging



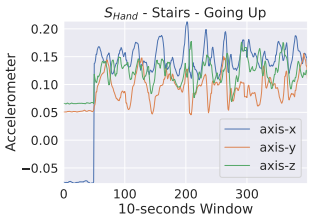
(e) S_{Hand} - Jogging



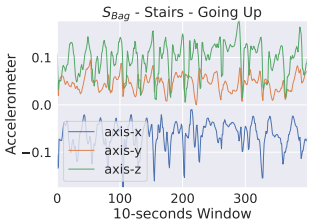
(f) S_{Bag} - Jogging



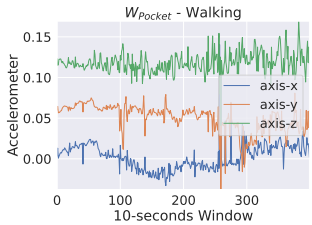
(g) S_{Pocket} - Stairs - Going Up



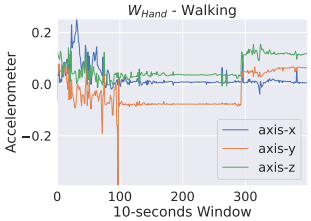
(h) S_{Hand} - Stairs - Going Up



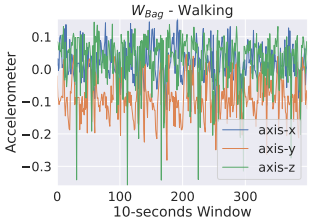
(i) S_{Bag} - Stairs - Going Up



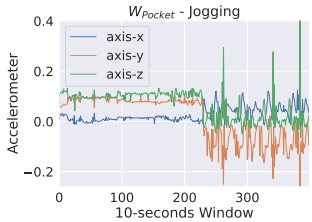
(j) W_{Pocket} - Walking



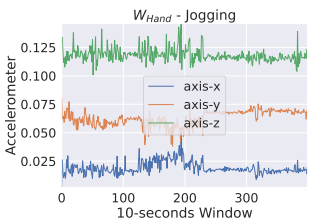
(k) W_{Hand} - Walking



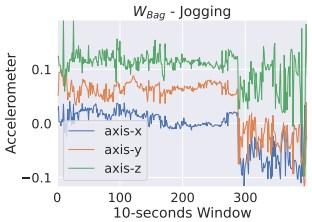
(l) W_{Bag} - Walking



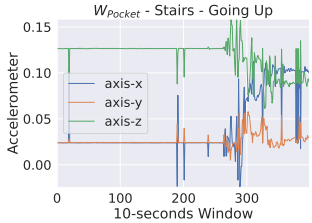
(m) W_{Pocket} - Jogging



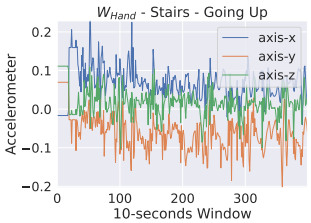
(n) W_{Hand} - Jogging



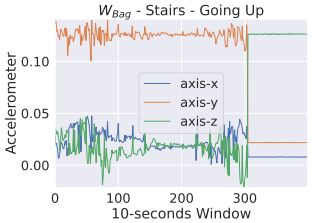
(o) W_{Bag} - Jogging



(p) W_{Pocket} - Stairs Going Up



(q) W_{Hand} - Stairs Going Up



(r) W_{Bag} - Stairs Going Up

Figure 6.5: Raw accelerometer tracings sampled from Walking, Jogging, and Stairs Going up contexts within each dataset.

6.4.1 Baselines

We compared *Triple-DARE* to state-of-the-art deep-learning based DA models : 1) *CORAL* [101]: A state-of-the-art UDA model that utilizing deep-coral discrepancy loss as baseline. 2) *DAN* [100]: A model with only our MK-MMD domain alignment loss. 3) *HDCNN* [103]: a state-of-the-art baseline DA method previously applied on smartphone sensor data. *HDCNN* is a transductive transfer learning model with KL divergence loss on the obtained feature vectors across domains. 4) *SOURCE*: A model trained on the source domain without any adaptation to the target domain. Our proposed model, *Triple-DARE*, which uses our joint-fusion triplet loss.

6.4.2 Implementation and Experimental Settings

1) Hyper-parameters We tuned the hyper-parameters of MLP and CNN using grid search. The learning rate is initialized at 1e-1, balancing coefficients are initialized as $\lambda_1 = 1$, $\lambda_2 = 0$, and $\lambda_3 = 0$. The balancing coefficients and the learning rate are increased or decreased following the schedule mentioned in [102], making our model highly confident on source labels and less sensitive to low-quality pseudo labels at the early stages of the training. The batch size is set at 256. The Adam optimizer was used. The back-bone layers used in our DA method are shared across all experiments: handcrafted-features MLP with two layers, both have 16 hidden dimensions, MLP domain classifier with one layer that has 32 hidden dimensions, and CNN that has attention blocks for separate and merged sensors layers, followed by an average pooling layer, adapted from [22]. All raw

Table 6.3: Overall context prediction.

Overall UDA Tasks	Accuracy	F1-micro
Triple-DARE	**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
Triple-DARE	**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

sensor data were input to a 3-layer CNN. Then their outputs are concatenated and forwarded to another 3-layer CNN. Attention blocks are used to focus on salient regions in inputs [117, 138]. Euclidean distance is used for computing pairwise distances in triplet mining and α is set to 0.1. The final context prediction layer has LeakyReLU activation, followed by Sigmoid activation.

2) Evaluation Protocol Due to the class imbalance in our context datasets, we used the *F1* metric to evaluate HCR performance in the UDA setting in addition to reporting classification accuracy. As the sizes of the source and target domain datasets might not be the same, we iterate through the target domain dataset with random sampling. However, we evaluate our model on all samples in the target domain dataset.

6.4.3 Results and Findings

1) Overall Results: In Table (6.3), we report the overall performance scores for our *Triple-DARE* compared to baseline models. *Triple-DARE* outperforms the baseline methods in the overall UDA tasks and Lab-to-field UDA task by 4.5%

Table 6.4: F-1 scores - comparing different methods for scripted contexts with cross-prioception UDA tasks, varying the amounts of used source labels.

		Scripted contexts with cross-prioception UDA tasks						
Training %	Method	$S_{Bag} \rightarrow S_{Hand}$	$S_{Bag} \rightarrow S_{Pocket}$	$S_{Hand} \rightarrow S_{Bag}$	$S_{Hand} \rightarrow S_{Pocket}$	$S_{Pocket} \rightarrow S_{Bag}$	$S_{Pocket} \rightarrow S_{Hand}$	Avg
0.2	*Triple-DARE*	**0.500**	**0.651**	0.213	0.318	**0.652**	**0.467**	**0.467**
	CORAL	0.357	0.328	0.357	0.428	0.352	0.378	0.367
	DAN	0.341	0.436	0.285	0.265	0.418	0.403	0.358
	HDCNN	0.341	0.492	**0.472**	**0.470**	0.468	0.380	0.437
0.4	*Triple-DARE*	**0.557**	**0.617**	0.444	**0.492**	**0.767**	**0.511**	**0.565**
	CORAL	0.380	0.584	**0.455**	0.457	0.633	0.484	0.499
	DAN	0.452	0.509	0.418	0.451	0.721	0.459	0.502
	HDCNN	0.424	0.580	0.504	0.558	0.704	0.441	0.535
0.6	*Triple-DARE*	0.497	0.588	**0.570**	**0.653**	0.744	**0.542**	**0.599**
	CORAL	**0.577**	**0.688**	0.505	0.505	**0.754**	0.448	0.580
	DAN	0.540	0.634	0.428	0.459	0.653	0.429	0.524
	HDCNN	0.345	0.561	0.575	0.540	0.645	0.445	0.518
Average	*Triple-DARE*	**0.518**	**0.619**	0.409	0.488	**0.721**	**0.507**	**0.544**
	CORAL	0.438	0.533	0.439	0.463	0.580	0.437	0.482
	DAN	0.440	0.526	0.377	0.392	0.597	0.430	0.461
	HDCNN	0.370	0.544	**0.517**	**0.523**	0.606	0.422	0.497
-	No Adaptation	0.319	0.469	0.476	0.470	0.260	0.315	0.385

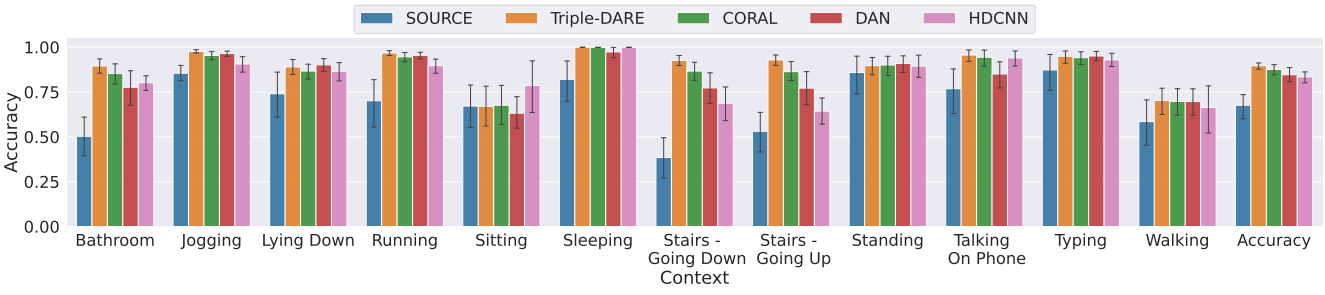


Figure 6.6: Target predictions scores per label averaged across different UDA task domains.

Table 6.5: F-1 scores - comparing different methods for Lab-to-field UDA tasks, varying the amounts of used source labels.

		Lab-to-field UDA Tasks			
Training %	Method	$S_{Bag} \rightarrow W_{Bag}$	$S_{Hand} \rightarrow W_{Hand}$	$S_{Pocket} \rightarrow W_{Pocket}$	Avg
0.2	*Triple-DARE*	**0.101**	0.080	**0.326**	**0.169**
	CORAL	0.089	**0.087**	0.150	0.109
	DAN	0.079	0.077	0.165	0.107
	HDCNN	0.087	0.084	0.181	0.117
0.4	*Triple-DARE*	**0.118**	**0.143**	**0.359**	**0.207**
	CORAL	0.092	0.075	0.165	0.111
	DAN	0.101	0.093	0.244	0.146
	HDCNN	0.106	0.108	0.266	0.160
0.6	*Triple-DARE*	**0.111**	0.112	**0.341**	**0.188**
	CORAL	0.110	**0.123**	0.210	0.148
	DAN	0.100	0.084	0.209	0.127
	HDCNN	0.094	0.102	0.285	0.160
Average	*Triple-DARE*	**0.111**	**0.112**	**0.341**	**0.188**
	CORAL	0.097	0.093	0.173	0.122
	DAN	0.096	0.087	0.198	0.127
	HDCNN	0.096	0.098	0.244	0.146
-	No Adaptation	0.108	0.110	0.180	0.133

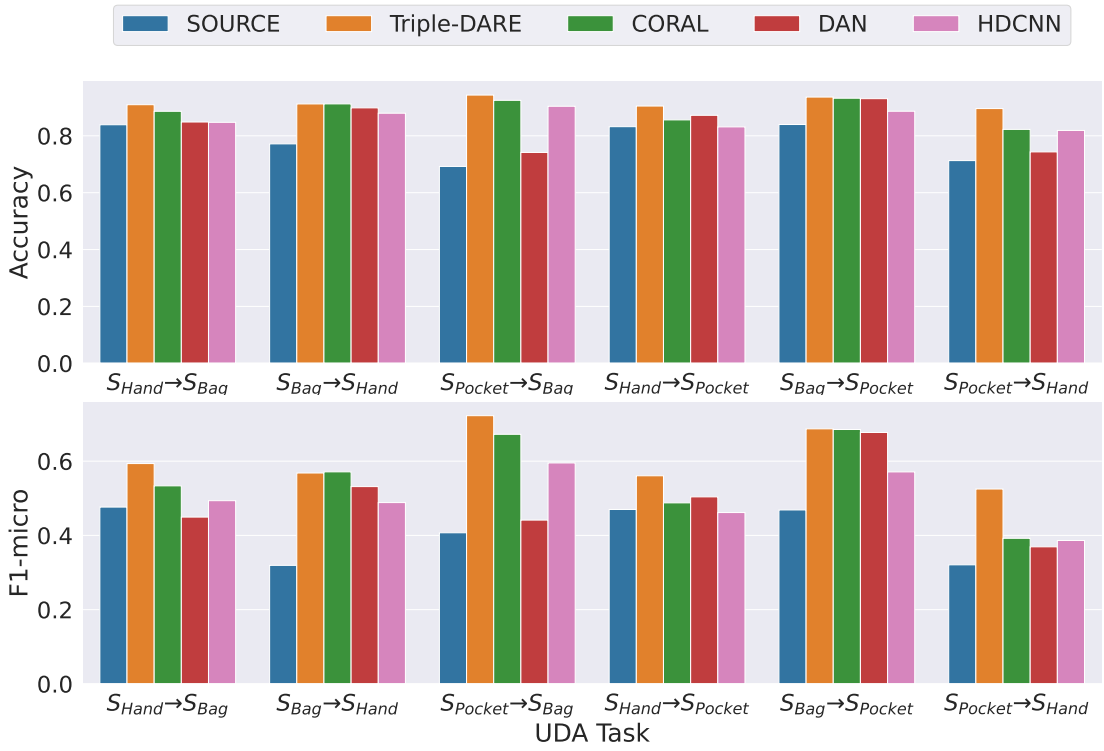


Figure 6.7: Scripted context data with cross-prioception UDA tasks.

increase in F1-score and 6.3% increase in classification accuracy. The results are shown in Figure (6.6) demonstrate the performance per label aggregated over all the UDA tasks, showing that our approach outperforms state-of-the-art methods across several context labels. In general, the advantage of using UDA methods can be observed over classifiers that are solely trained on the source domain without leveraging unlabeled data. In particular, UDA methods helped a lot in the Jogging, Running, Going Up and Down Stairs labels where the user is likely to stop providing labels while performing these activities in the wild. However, our approach takes advantage of the high fidelity labels acquired in the scripted study and improves adaptation. We also can see that predictions for Sitting and Walking are the most difficult compared to other labels, which could be due to a significant difference in target label distributions (see Table (6.2)).

2) Scripted contexts with cross-prioception UDA tasks: In Figure (6.7), Triple-

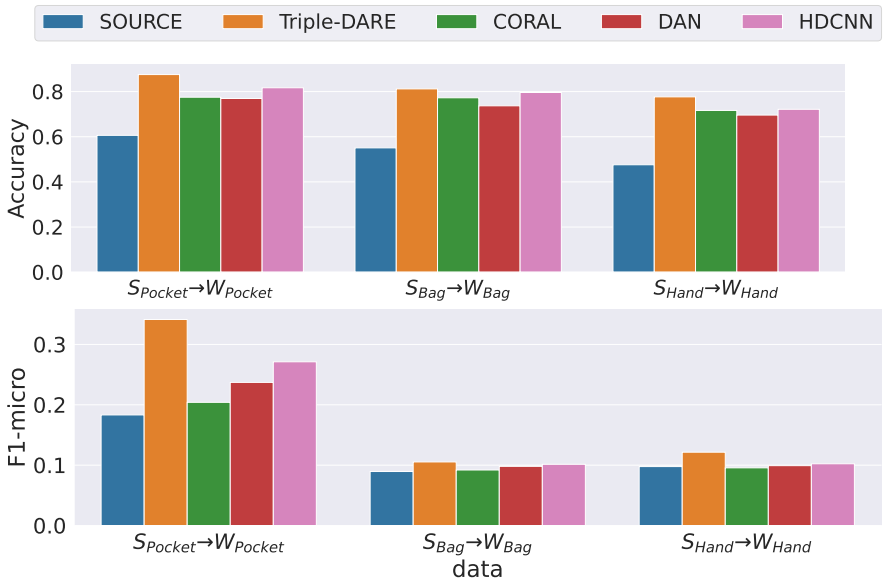


Figure 6.8: Scripted context to In-The Wild UDA tasks scores.

DARE consistently outperforms the baseline methods across the different cross-prioception UDA tasks. The UDA tasks with "Phone In Hand" as the target domain have significantly benefited from the adaptation process. This benefit is due to the amount of signal noise introduced when the phone is not stationary. We also notice that *CORAL* performs better than *DAN* in most cases.

3) Lab-to-field generalization UDA tasks: We report the scores for our lab-to-field generalization UDA tasks in Figure (6.8). Another clue we get about diversity placements is by looking at huge differences in the scores obtained from "Phone in Pocket" compared to "Phone In Bag" and "Hand". We speculate that when the phone is placed in a bag or in hand, the model can hardly map data collected from scripted and in-the-wild datasets to a common feature space. However, we see a noticeable improvement over state-of-the-art baseline methods in adapting models learned on scripted data to make context predictions on in-the-wild data with a "Phone in Pocket" prioception.

4) Training under insufficient labels: We studied the performance of our model

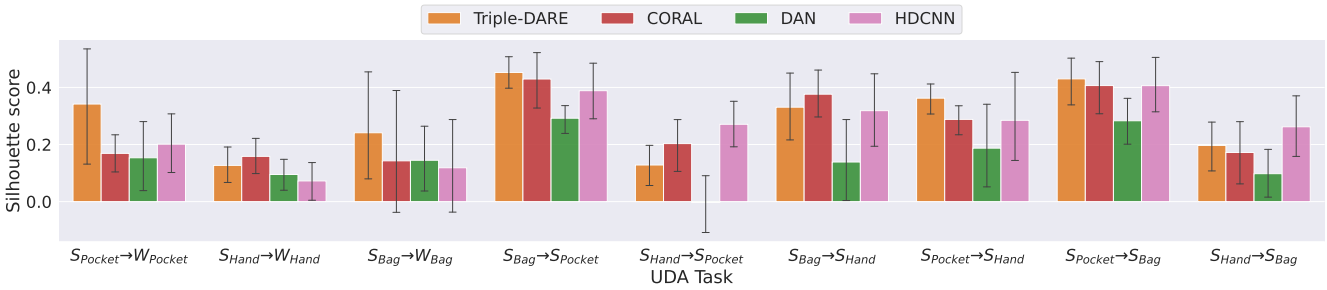


Figure 6.9: Compactness measure on feature embeddings.

when the number of labels from the source domain is varied. In Figure (6.10), we plot the prediction scores, obtained on multiple scripted cross-prioception domains, averaged over different source domains. In Tables (6.4) and (6.5), a more detailed version for this experiment is provided. We can see that our Triple-DARE achieves higher prediction scores on the target domain, with minor amounts of source labels, outperforming baseline methods in almost all UDA tasks.

5) Intra-class compactness and inter-class separation: To provide a measure of compactness and separation in the learned feature embeddings, we utilized the Silhouettescore Score $= \frac{b_i - a_i}{\max(b_i, a_i)}$, where b_i is the shortest mean distance between a point to all other points in any other cluster, whereas a_i is the mean distance of i and all data points from the same cluster. This score measures both compactness and separation. To calculate the Silhouette scores on the learned feature embeddings, we assign each instance with cluster labels using one of the binary context labels. Then, we average the scores over labels. The scores are reported in Figure (6.9), which shows our *Triple-DARE* achieves better compactness and separation scores in most UDA tasks. Also, CORAL achieves higher scores than DAN in most cases. Additionally, we can visually see the quality of the learned feature embeddings in Figure (6.11), which depicts the same context instances represented by feature embeddings learned using DAN and *Triple-DARE*. The visualization

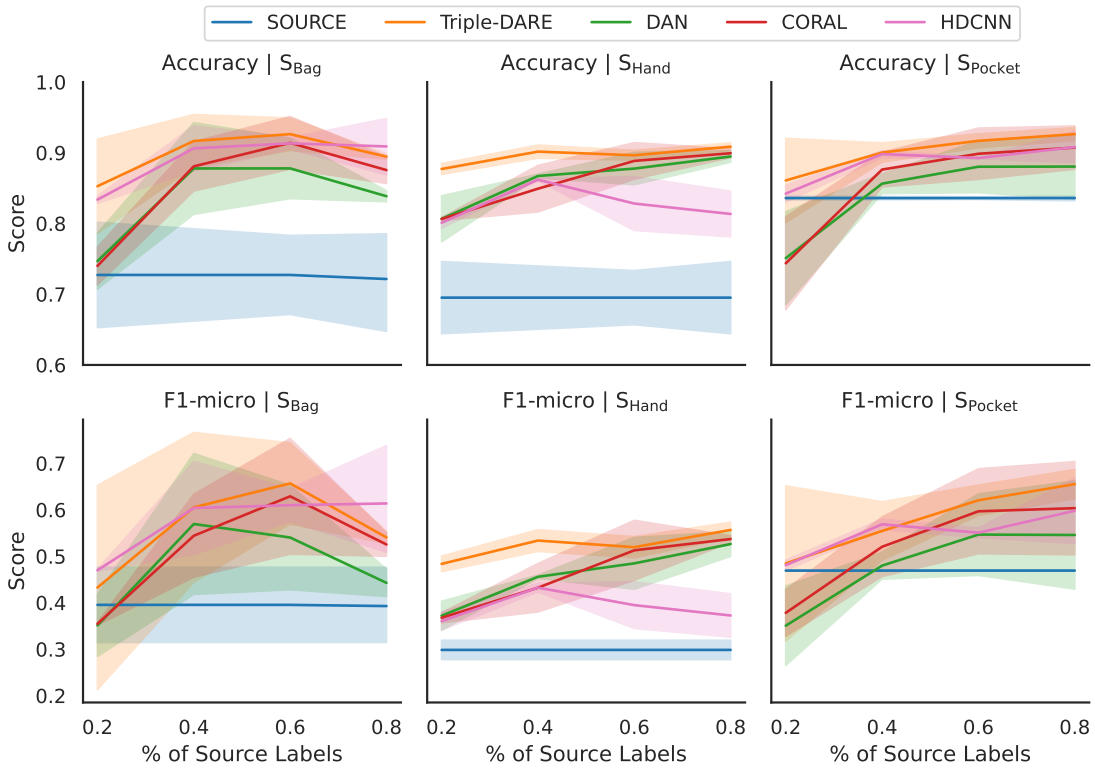


Figure 6.10: Scores for each source domain in scripted contexts with cross-prioception UDA tasks, averaged over each target, varying the number of labels from the source domain.

is obtained by projecting feature embeddings into a two-dimensional space using the T-distributed Stochastic Neighbor Embedding (TSNE) [139].

(7) Ablation Study: We conducted an experimental ablation (shown in Figure (6.12)) 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.

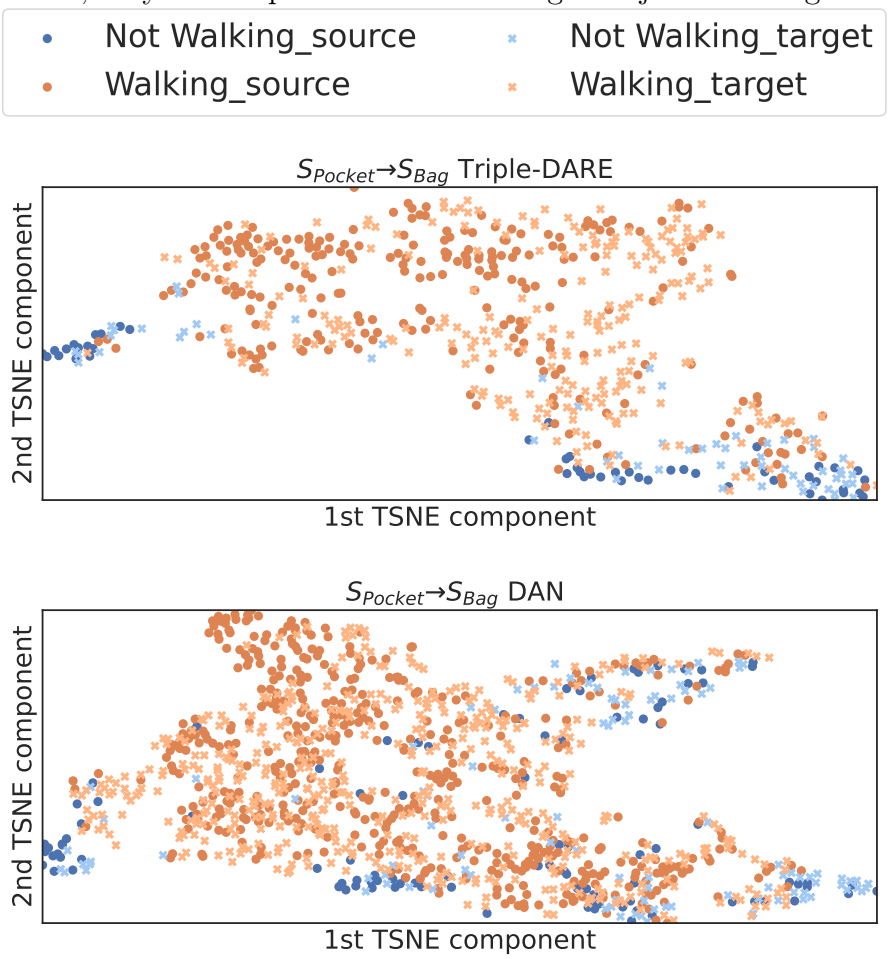


Figure 6.11: Visualization of the learned feature embeddings for TripleDARE (top) and DAN (bottom), using TSNE dimensional reduction.

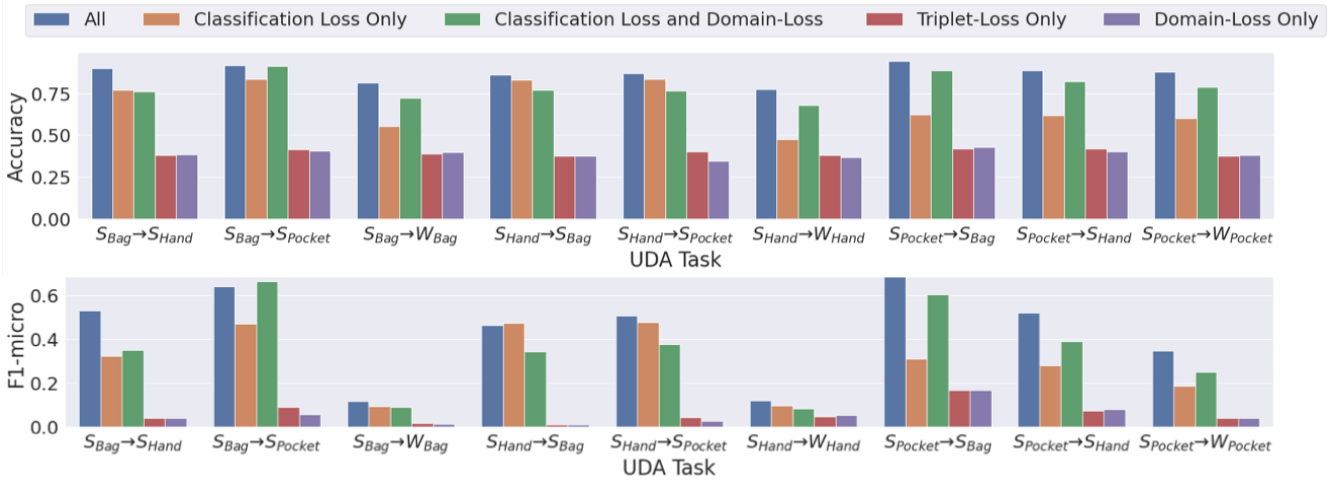


Figure 6.12: Ablation study, evaluating the contribution of *Triple-DARE*'s each component.

6.5 Chapter Summary

Several issues reduce the performance of machine learning HCR models on in-the-wild datasets, including the diversity of phone placements and smartphone models. Lab-to-field methods try to improve the performance of HCR models by first training them on similar scripted datasets, then adapting them for use in predicting context labels in in-the-wild datasets. We designed DA strategies that are susceptible to covariate shifts between the scripted and in-the-wild datasets, improving lab-to-field generalization. We proposed Triple-DARE, a UDA deep-learning model for HCR on smartphones, comprised of three parts: 1) domain alignment loss using MK-MMD 2) a classification loss and, 3) joint-fusion triplet loss designed for multi-labeled datasets. Triple-DARE learns domain-invariant features common to both datasets, reducing the influence of highly noisy in-the-wild data by using its attention mechanism to focus on salient regions in sensor inputs, achieving a high F1-score for various UDA tasks on our scripted and in-

the-wild context datasets. Using its domain alignment loss, Triple-DARE is able to map the source and target feature encoding into a standard feature distribution space with better performance than state-of-the-art baseline methods. Furthermore, the triplet loss improves discrimination, increasing intra-class compactness and inter-class separation while leveraging massive amounts of unlabeled data. *Triple-DARE* outperforms other state-of-the-art DA baselines, improving on their F1-score and classification accuracy by 4.6% and 1.89%, respectively, and improving on models with no adaptations by 10.7% and 14.7%, respectively.

Part IV

Robust Representations under Adversarial Attacks

Chapter 7

Adversarial Human Context Recognition: Evasion Attacks and Defenses

7.1 Introduction

Recent state-of-the-art smartphone HCR methods utilize datasets gathered in the wild, wherein sensor data is collected continuously from users' smartphones as they live their lives. Users then submit self-reports of contexts visited that are used to annotate the sensor data [14, 26]. However, despite the fact that these study designs collect more realistic data, they also make it easier for adversaries to attack the system. Specifically, adversaries can send modified data samples directly to the dataset aggregator to mislead the recognition system and produce inaccurate results [140] . However, these samples meant to degrade the performance can easily be detected by outlier detection methods [71, 140] unless these samples are carefully adjusted and generated as *adversarial examples*, small perturbations on inputs cause models to produce erroneous classifications with high confidence. It is also worth mentioning that the difference between adversarial

perturbations and random noise is that adversarial examples are misclassified far more frequently than samples that have been disrupted by random noise, even though the quantity of random noise is substantially more significant than the adversarial perturbation [141].

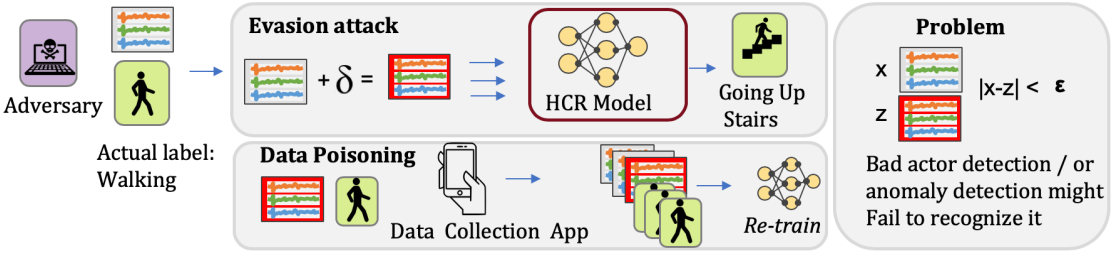


Figure 7.1: The general evasion adversarial attack problem.

Prior work and challenges. This work examines attacks and defenses against HCR systems. Researchers have studied *adversarial examples* vulnerabilities in computer vision [43], natural language processing [44], and speech recognition [45], the focus of exploring adversarial vulnerabilities has only recently shifted to time-series-based models [46, 47, 48] or sensory-based classifications [49, 112, 50]. Sah *et al.* recently studied utilizing wearables for activity recognition, investigating the transferability and generation of adversarial examples. However, they did not propose any countermeasures [50]. Moreover, in comparison to wearables, the nature of HCR data collected by smartphones is much more complicated. For instance, sensor signals for the same activity have different characteristics when the phone is held in various prioceptions [24, 25]. In fact, prioception has the greatest impact in terms of differences in smartphone context sensor data [59]. When performing a particular activity, smartphone owners may choose to either hold the smartphone in their hand or place it in their pants or coat pocket. Therefore, methods to evaluate the security of HCR data collected by smartphones in the wild are needed,

especially for such large amounts of data with many diversities. To understand the impact of this problem, imagine a detection algorithm failing to detect an older person's fall or a warfighter with TBI not being detected on time due to an adversarial attack. Other sensor-based human activity recognition models, such as WiFi-based models [142], are used in smart homes; malicious manipulation of sensing data can deceive the access control system, posing a security risk. There is a dire need to define and evaluate the effects of adversarial examples for smartphone HCRs, which could have fatal outcomes if perpetuated in CA systems for mobile health and behavioral medicine.

Potential adversarial attacks. Poisoning and Evasion are two major adversarial attacks [41]. *Poisoning* or training-stage attacks contaminate training data. When adversaries can access a model's training data, they may try to undermine the machine learning model by manipulating the data or labels. *Evasion attacks*, or inference-based attacks, are common and well-studied. We focus on evasion attacks, which occur when an attacker manipulates test data to fool classifiers. Adversarial attacks can be divided into white-box and black-box attacks based on system knowledge and access [41]. White-box attacks require model parameters [113, 53]. Black-box attacks (listed in Figure (3.1)) require the ability to query the model with arbitrary inputs [114, 115]. We focus on score-based and label-based evasion attack generation methods in accordance with plausible scenarios of possible adversarial attacks. These methods can generate adversarial perturbations using only class confidence scores (Zoo attack) or class decisions (HSJ attack).

Problem description. HCR models deployed in the wild are vulnerable to

adversarial threats such as data poisoning, which targets the training stage. or evasion attacks that occur during model inference. Using adversarial threats as a measure of model robustness is also crucial. Concerning the feasibility of these threats, it is important to note what attacks an adversary can perpetuate on a smartphone HCR classifier under two threat models: a) a score-based threat model assumes access to class confidence scores; b) a label-based threat model assumes only access to the predicted label. An adversary can query a pre-trained HCR classifier with arbitrary inputs and observe model output (e.g., class confidence scores or only class labels). During in-the-wild data collection, the adversary can send sensor inputs and self-reported annotations. In Figure (7.1), we showcase a scenario of an evasion attack against our smartphone HCR system and how that can lead to a potential poisonous attack. Evasion attacks can be used to create poisonous examples, particularly inputs that look similar to legitimate inputs but have misleading labels. This study focuses on evasion attacks. *To the best of our knowledge, this is the first work that studies black-box based evasion attacks against smartphone based HCRs.*

Proposed research In this work, we formulate potential types of HCR evasion attacks as well as propose and demonstrate the effectiveness of specific, practical defenses. More broadly, our work is a step towards improving the robustness and generalizability of HCR models when deployed in the real world. According to Geirhos *et al.* , one way to evaluate how reliable and robust deep neural networks are when deployed in natural environments is to test them against *adversarial examples*, which motivated us even more to conduct this study [40].

Contributions. These are the primary contributions of this work:

1. Rigorous formulation of the adversarial HCR problem

2. Definition of a suite of black-box adversarial evasive attacks of deployed smartphone HCR models, which require minimal access to the HCR model and data. Primarily, we experiment with black-box evasion attack generation methods under two assumptions: a) if class confidence scores are available to the adversary, we use the Zoo score-based method; b) if only class decisions are available to the adversary, we use the HopSkipJump label-based method.

3. Comprehensive evaluation of the vulnerability of smartphone HCR models to black box evasive attacks.

4. Definition of a suite of novel defenses to adversarial HCR attacks that are based on provable defense strategies. We propose *RobustHCR*, which adapts a duality-based method to improve the neural network's robustness, which can be provably resilient to norm-bounded perturbations [112].

5. Comprehensive evaluation via extensive empirical experiments of our proposed method *RobustHCR* and comparisons to state-of-the-art baseline defensive methods.

The remaining sections of this chapter are structured as follows. Section 7.2 lists related work. Section 7.3 contains our threat model. Section 7.4 introduces our proposed approach. Section 7.5 demonstrates our evaluation. Results and Discussion are presented in section 7.6. Finally, Section 7.8 summarizes all the findings.

7.2 Background & Related Work

7.2.1 Smartphone-based Mission-critical Applications

Smartphone-based recognition of user context and ambulatory activities [82] has several practical, mission-critical applications. Compromising such systems could have serious ramifications. For instance, smartphone-based HCRs may be utilized to continuously track and monitor the health of soldiers or veterans to detect Traumatic Brain Injuries (TBI) or an infectious diseases (e.g. Covid-19). By monitoring smartphone health biomarkers, abnormal user behavior, physiological indicators, activities, and context visit patterns can be identified [26]. Sun *et al.* showed that smartphone-based activity recognition could detect aggravated assaults. They created iProtect to identify abuse and kidnapping. iProtect uses smartphone accelerometers to record and identify physical assaults. The importance of real-time assault detection for personal safety cannot be overstated [83]. These applications require accurate smartphone HCR predictions, which we aim to improve.

7.2.2 Black-box Evasion Attacks

One of the following methods is used to launch a black-box evasion attack: 1) Utilizing a transferable attack strategy [143, 50], an adversary may decide to train a substitute model to imitate the original model. For weight estimation, the attacker can use an architecture that is significantly superior to the original. In the absence of a substitute model, however, the attacker opts for 2) a query feedback mechanism [114, 115, 144], wherein the attacker continuously generates modified

inputs while querying the model under attack. We focus on two common black-box attacks that follow the query feedback mechanism (Zoo and HSJ), which were employed to generate successful imperceptible adversarial images to the human eye for image classification tasks [114, 115].

7.2.3 Evasion Defenses

There are three common types of evasion defense methods: 1) adversarial training, 2) provable defenses, and 3) regularization approaches [145]. Adversarial training defense augments adversarial examples during training to improve model robustness and stability [146]. Provable defenses aim to provide robust guarantees that there are no adversarial examples within a l_p-bounded region, which provides reliable predictions, especially needed for critical applications [112, 145]. Regularization techniques concentrate on making minor adjustments to the learning algorithm so that it can generalize more effectively. With regards to achieving robustness to adversarial examples, regularization approaches concentrate on avoiding small input variations that can result in algorithmic decision changes. This is accomplished by either enlarging the decision boundaries or restricting changes in the model's gradient.

To protect against evasion attacks on sensor data, state-of-the-art methods proposed include Adar [147] for data collected from wearable sensors and RobustSense [142] for data collected via WiFi-based activity recognition. These two methods employ adversarial training by incorporating adversarial examples during training and attempting to reduce the disparity between the distributions of clean and adversarial examples. In particular, Adar uses a white-box attack to generate

adversarial examples that are then used in the data augmentation process [147]. Additionally, Yang *et al.* tried to minimize the sum of the Kullback-Leibler (KL) divergences between every possible pair of clean and associated adversarial examples, with the goal of reducing the prediction gap and increasing the model's robustness [142]. Empirical evidence suggests that state-of-the-art approaches have a good chance of success, but only provable defenses can ensure reliability.

7.3 Threat Model

Assumptions. In this section, we define specific aspects and assumptions for our threat model [145] concerning smartphone HCRs. First, *the timing* of evasion attacks is during inference, or more specifically, when an adversary queries a classifier with arbitrary inputs. Second, in terms of *accessible information*, we consider only the black-box threat model, where an adversary can query the pre-trained HCR classifier with arbitrary inputs and observe model outputs. We test two possibilities, when class confidence scores are available and when only class decision labels are available. At the same time, the adversary can have indirect access to the training data by sending sensor inputs along with self-reported annotations as part of the in-the-wild data-gathering study. Also, we consider the *attack frequency* to happen without any constraints. Moreover, as a standard measure of an attack's effectiveness, we use a *perturbation measurement* of norm-bounded perturbations [144, 112, 114, 115]. Generally, adversarial attacks are evaluated based on the number of queries converging on model parameters. The attack will be more effective with fewer queries, as the time required to initiate an attack will

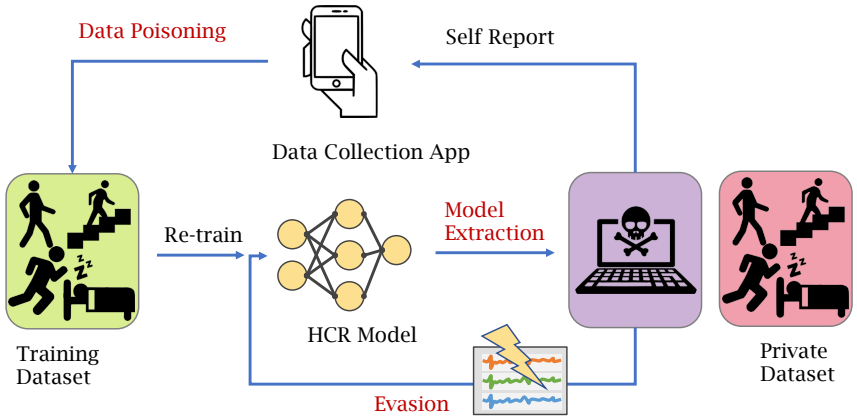


Figure 7.2: Adversarial attack types within the context of our HCR framework: Evasion: Attackers can launch evasion attacks by tampering with test data in an attempt to deceive trained classifiers, Data poisoning: adversaries may manipulate the model's training data or labels to undermine machine learning during deployment, and Model Extraction: involves a process of reverse-engineering the learning algorithm, seek to gain unauthorized entry to a sensitive system, user, or data information where data security and confidentiality are major concerns. *This work focuses on Evasion attacks only.*

be reduced. The perturbation norm is a standard metric of attack effectiveness, along with the amount of time needed to construct the attack. Standard error measures l_2 and l_∞ are the most frequently employed perturbation norms [144]. An adversary's *final objective* is to design an adversarial example that causes the HCR to make incorrect predictions, assuming that they already know the original input and the label produced by the classifier.

Potential adversarial attackers: Evasion attacks described above can be perpetrated by *adversary data scientists:* these are expert users knowledgeable in applying data science techniques with clear adversary intentions. They could inject malicious data that could effectively degrade the training or testing processes, which could not be detected easily by traditional anomaly detection or data sensitization techniques [55, 54]. They could also craft adversarial examples to fool HCR models into doing something else [50]. For instance, in data collection studies, one subject could avoid labeling quality procedures that ensure subjects are

doing what they are supposed to do. Another possibility for this attack is that someone under remote health monitoring pretends to have some sort of infectious disease by forcing the classifier to predict arbitrary labels (e.g., In Bathroom). In Figure (7.2), a demonstration of adversarial attack types within the context of our HCR framework is depicted.

7.4 Methodology

Inspired by the conceptual framework for building secure machine learning models outlined by Biggio *et al.* [55], we propose the following research methods that are aligned with the proactive approach, which involves simulating attacks on machine learning systems and doing a comprehensive analysis of the security profile of these systems by following these steps:

1. Identifying and describing significant evasion threats to an HCR machine learning system under development
2. Modeling attacks matching such evasion threats.
3. Launching simulated evasion attacks against the HCR model.
4. Developing appropriate countermeasures to evasion attacks.
5. Repetitively evaluating the performance of the HCR system until the evasion threat is mitigated prior to system deployment.

.

The methodology outlined above yields *RobustHCR*, our proposed method that has

two major components: 1) simulating threats (Adversarial attacks generation) and 2) Adversarial defenses to evasion attacks, which put security measures in place.

7.4.1 HCR Evasion Attack Problem Formulation

First, we introduce essential notations and outline an optimization framework for computing and defending against adversarial examples. With regards to smartphone HCRs, we have the dataset $\mathscr{D} = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^m$ where $\boldsymbol{x}$ represents a feature vector and $\boldsymbol{y}$ represents a human context, an output vector with multiple labels, where each label generated is expressed in a binary form (E.g walking vs not walking), and m represents the number of samples in the dataset. A deep neural network $f_\theta(x)$ that is able to recognize human contexts y given sensory inputs x, is trained. In a standard supervised learning setting, in order to train the classifier $f(x)$ to recognize human contexts. The cross-entropy loss function is minimized to achieve equation 7.1.

$$\mathcal{L}_{\text{cls}}^\theta = \frac{1}{m}\sum_{i=1}^{m} \ell_\Psi(f_\phi(x_i), y_i), \tag{7.1}$$

where ℓ is a cross-entropy function.

Our main objective is to simulate possible evasion attacks against $f_\theta(x)$ using *adversarial attack generation*, evaluate potential damage to HCR model performance, and then secure our model against this vulnerability with *adversarial defenses* that defend not only against potential adversarial perturbation but also, more generally, improve the consistency of predictions generated by our HCR classifier when given similar inputs.

The objective of the *adversarial attack generation* stage is to provide samples that resemble legitimate inputs, but cause the HCR classifier to make inaccurate predictions. Formally, given a target pre-trained model $f(\theta)$ and an original input $x \in X$ with the associated class c, the goal is to produce an undetectable perturbation δ to create an adversarial example $\bar{x} = x + \delta$ and let the target model classify $\bar{x}$ as $\bar{c} \neq c$. In order to ensure imperceptibility and to produce samples that look similar to the legitimate input, the amount of perturbation allowed is constrained, typically enforcing δ to be within l_p ball such as l_2 [114, 115] or l_∞ [115]. On the other hand, the goal of adversarial defense is instead of minimizing the loss at only the legitimate examples; we utilize a standard robust optimization loss by minimizing an upper bound on the worst possible perturbation in a p-ball around each x_i example, following the approach in [148]:

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^{N} \max_{\|\Delta\|_p \leq \epsilon} L\left(f_\theta\left(x_i + \Delta\right), y_i\right) \tag{7.2}$$

However, this solution may be intractable for deep, non-linear neural networks. Thus, we adopt an approximate solution based on duality-based networks [112], which minimizes an upper bound on the worst-case adversarial example within a perturbation region $\mathcal{B}_\epsilon(x)$:

$$\mathcal{B}_\epsilon(x) = \left\{x + \delta : \|\delta\|_p \leq \epsilon\right\}, \tag{7.3}$$

where p is l_2 or l_∞. Next, we provide details for each step.

7.4.2 Adversarial Attacks Generation

The majority of prior research work on evasion attack generation methods applied to sensor data focused more on white-box settings [50, 112], which require a level of system access that is impractical and hence make them rare in practice. In order to better align with more realistic scenarios, we instead focus on black-box attack scenarios. Unlike prior work that exploits the transferability property of adversarial examples [143, 50], We concentrate on methods that only require access to query the model with an arbitrary input: 1) Zoo: zeroth order optimization [114] which utilizes class confidence scores; and 2) Hopskipjump: a label-based attack [115] that requires only the class decisions.

Zoo: Zeroth Order Optimization

The difficulty of optimizing a function f based only on access to function outputs $f(x)$, as opposed to gradient values $\Delta f(x)$, is known as zeroth-order optimization [149, 150]. Chen *et al.* applied a zeroth-order algorithm that enables calculation of the classifier gradient without having access to the classifier or utilizing the attack transferability of surrogate models, making it perfect for generating adversarial examples in a black-box scenario [114, 151]. This method uses the technique of the symmetric difference quotient through two queries to estimate the model gradients. Intuitively, model gradients in the direction of vector v can be estimated by calculating the objective function values at two extremely close points $f(x+\epsilon v)$ and $f(x-\epsilon v)$ using a small constant ϵ. As shown below, partial model gradients

can be computed:

$$\frac{\partial f(x)}{\partial x_i} = \frac{f\left(x + \epsilon z_i\right) - f\left(x - \epsilon z_i\right)}{2\epsilon} \tag{7.4}$$

where ϵ is a small constant (e.g., $= 0.0001$ in our experiments), and $z_i \in \{0, 1\}_n$ represents a unit vector where i-th value is 1 and the rest are 0.

HopSkipJump (HSJ): a Query-efficient Label-based Black-box Attack

HSJ is a decision-based evasion attack proposed by Chen *et al.* and applied for attacking image classification tasks in a black-box setting [115]. It is a decision-based attack that only has access to the predicted output class and proposes a new method for estimating the gradient direction along the decision boundary. In terms of effectiveness, an HSJ attack can deceive an image classifier with a 6-bit per pixel perturbation in 50% of time in a large-scale dataset [115]. The attack is a decision-based attack that produces an adversarial example $\bar{x}$ by solving the subsequent optimization problem:

$$\min_{\bar{x}} \left\|\bar{x} - x\right\|_p, \text{s.t. } \phi_x\left(\bar{x}\right) = 1, \tag{7.5}$$

where x is the original instance, $\bar{x}$ is the perturbed input, p is either l_2 or l_∞ distance and ϕ_x is given by the following:

$$\phi_x\left(\bar{x}\right) = \text{sign}\left(S_{x^*}\left(\bar{x}\right)\right) = \begin{cases} 1, & S_{x^*}\left(\bar{x}\right) > 1 \\ -1, & \text{otherwise} \end{cases}, \tag{7.6}$$

and S_{x^*} is defined as

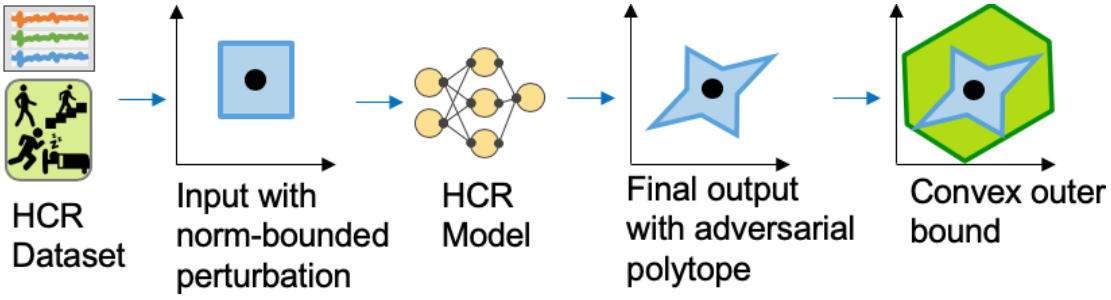


Figure 7.3: Training robust models against adversarial examples using Convex Adversarial Polytope. The idea is to bound the output of a neural network and find an upper bound for worst-case adversarial examples.

$$S_{x^*}(\bar{x}) = \max_{c \neq c^\star} F_c(\bar{x}) - F_{c^\star}(\bar{x}), \tag{7.7}$$

where $c^\star$ is original classifier decision.

The algorithm uses the binary information at the decision boundary to estimate the gradient direction, which is used to generate adversarial examples. Each iteration of the method consists of two steps, and iterations are performed repeatedly. The initial step entails conducting a binary search between the input and target samples to identify the sample that is closest to the model's boundary. Following this, the obtained sample is randomly manipulated. These modified samples are queried, and the resulting information is used to estimate the gradient using the collected labels. Finally, the obtained gradient is applied to the boundary sample to perturb it. All steps are performed until the maximum number of iterations specified by the attacker has been reached.

7.4.3 Adversarial Defenses

To improve the neural network's robustness and obtain performance guarantees, we adapt a duality-based neural network, a technique for training deep ReLU

networks that is demonstrably resistant to norm-bounded perturbations, as proposed by Wong *et al.* [112]. This method is designed for ReLU-based neural networks, and first they define an adversarial polytope for deep ReLU networks, as illustrated in Figure (7.3), which is hard to optimize, so they come up with a convex outer bound (green region), which can be solved using convex optimization. However, for training robust models efficiently using backpropagation, we utilized their approach using the dual problem of the corresponding linear program. By leveraging this method, we aim to discover models for which we can guarantee accuracy not only for the input example x, but also for the entire perturbation region $\mathcal{B}_\epsilon(x)$. In particular, these methods can compute an upper bound $\mathcal{J}(x)$ for a given neural network f subject to some perturbation set $\mathcal{B}_\epsilon(x)$ around input x.

$$\max_{z \in \mathcal{B}_\epsilon(x)} f(z) \cdot \gamma \leq J(x; \gamma), \tag{7.8}$$

for any constant $\gamma \in \{-1, 1\}$.

Specifically, let (ℓ, u) represent the lower and upper bounds on the output of a network f subject to perturbations in $\mathcal{B}_\epsilon(x)$ near the example x:

$$\ell \leq \min_{z \in \mathcal{B}_\epsilon(x)} f(z), \quad \max_{z \in \mathcal{B}_\epsilon(x)} f(z) \leq u \tag{7.9}$$

Eventually, the idea is that instead of trying to find an exact solution in (7.2), we are able to find an upper bound for the worst-case adversarial example by replacing the robust optimization loss with the following objective:

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^{N} L\left(-J\left(x_i, g_\theta\left(e_{y_i} 1^T - I\right)\right), y_i\right) \tag{7.10}$$

where $J\left(x, g_\theta(c)\right)$ is the dual objective, e_y represents logit scores learned for class y, I is the identity matrix, and $g_\theta(c)$ is the bounded version of classifier f determined by the computed bounds obtained using an open-source package, see 7.5 Experimental Evaluation - Implementation section. Using that method, we can guarantee that no adversaries can attack the model further with more extreme examples than the worst found adversarial example.

7.5 Experimental Evaluation

RobustHCR and baseline models for context recognition were evaluated against two black-box evasion attacks (Zoo and HSH) using multiple smartphone HCR datasets. We examine the potential harm to the performance of HCR models that could be caused by adversaries, as well as the trade-off between performance on clean inputs and security against evasion attacks. Lastly, we assess the quality of adversarial examples generated and how closely they resemble legitimate inputs.

7.5.1 Research Questions

We examine the subsequent research questions:

1. RQ1: How much does each of the adversarial attacks reduce the performance of HCR classifiers?

2. RQ2: How similar are these adversarial examples to the original inputs?

3. RQ3: How well does our proposed adversarial defense method improve the robustness of HCR models against adversarial attacks?

7.5.2 Datasets

ExtraSensory dataset. Extrasensory is an HCR dataset collected from UCSD college students' smartphones and smartwatches as they went about their daily lives. Statistical features were extracted, which correspond to at least one of 51 possible context classes [14]. However, only the smartphone sensor features were used in our experiments.

Scripted and In-the-wild datasets. These datasets were collected by the WASH team and closely followed ExtraSensory's data collection and label annotation methodology. *The scripted data* was collected in specific college buildings, laboratories, and routes. The smartphone application collected information from 100 participants who performed 16 activities with four different phone placements. During the data collection session, which lasted about one hour per subject, human proctors oversaw and manually annotated the data. *In-the-wild dataset:* 103 participants downloaded a smartphone app that passively collected data for two weeks as they lived their daily lives. Periodically, subjects were asked to self-report the context labels listed in Table (7.1) that they visited. By collecting data using individuals' smartphones, our in-the-wild dataset reflected a variety of manufacturer hardware and realistic user contexts.

Table 7.1: Contexts for which data was gathered in our Smartphone HCR Study Collected Contexts.

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

7.5.3 Data Preprocessing and Feature Extraction

All datasets were uniformly preprocessed. Segments were generated using a 20-second time window, and contexts were formulated as multi-label vectors. We used information gathered from five different sensors: the accelerometer, gyroscope, global positioning system (GPS), magnetometer, and the phone's status (discrete attributes such as whether or not the screen is locked on the device). Statistical, temporal, and frequency-based features were computed for each sensor modality. Thereafter, Z-score normalization was applied by taking the difference between the mean and the standard deviation and dividing by the latter $z_i = \frac{x_i - \bar{x}}{s}$. To train a feed-forward network, these manually extracted features are utilized to build a vector. Vaizman *et al.*'s open-source work on github.com/cal-ucsd/ExtraSensoryAndroid was used to implement 188 new features [14].

Table 7.2: A sample list of handcrafted features used on our sensor data [14].

Feature	**Formulation**
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	$\mathbb{E}\left[(s-\bar{s})^4\right] / \mathbb{E}\left[(s-\bar{s})^2\right]^2$
Signal magnitude area	$\frac{1}{3}\sum_{i=1}^{3}\sum_{j=1}^{N}\lvert s_{i,j}\rvert$
Pearson Correlation coefficient	$C_{1,2}/\sqrt{C_{1,1}C_{2,2}}, C = \mathrm{cov}(s_1, s_2)$
Spectral energy of a frequency band [a, b]	$\frac{1}{a-b+1}\sum_{i=a}^{b} s_i^2$

s: signal vector, N: signal vector length Q: quartile

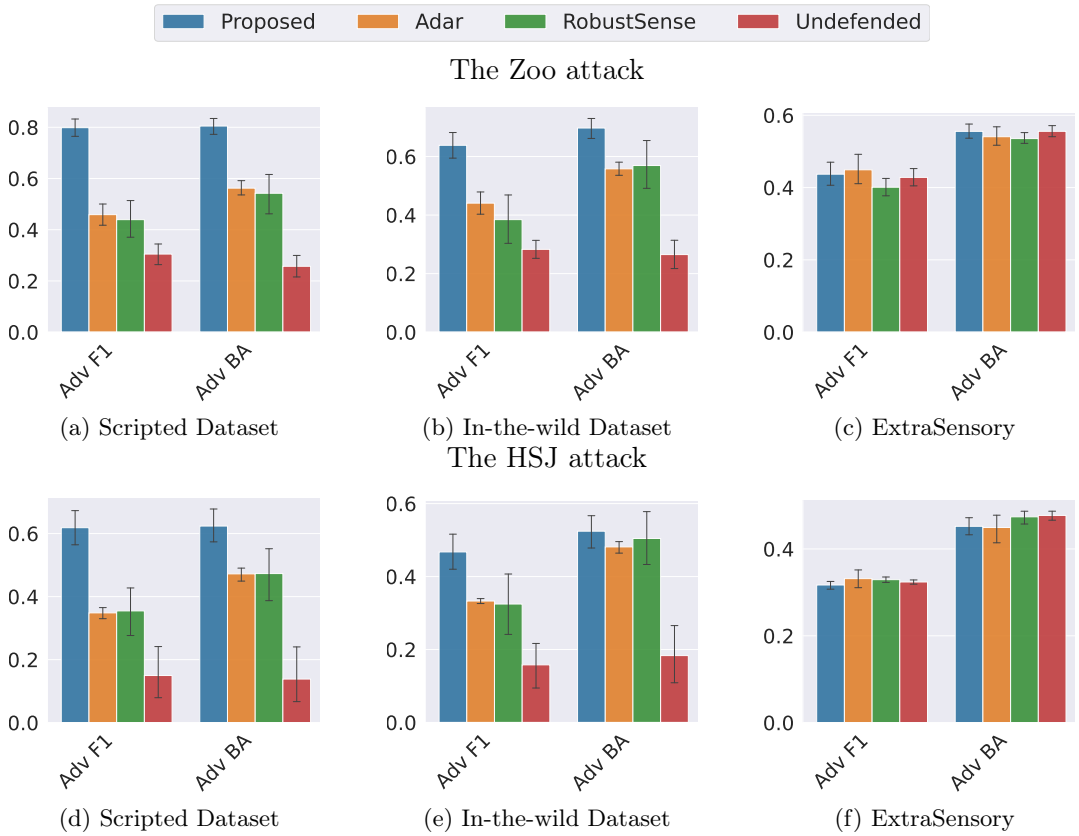


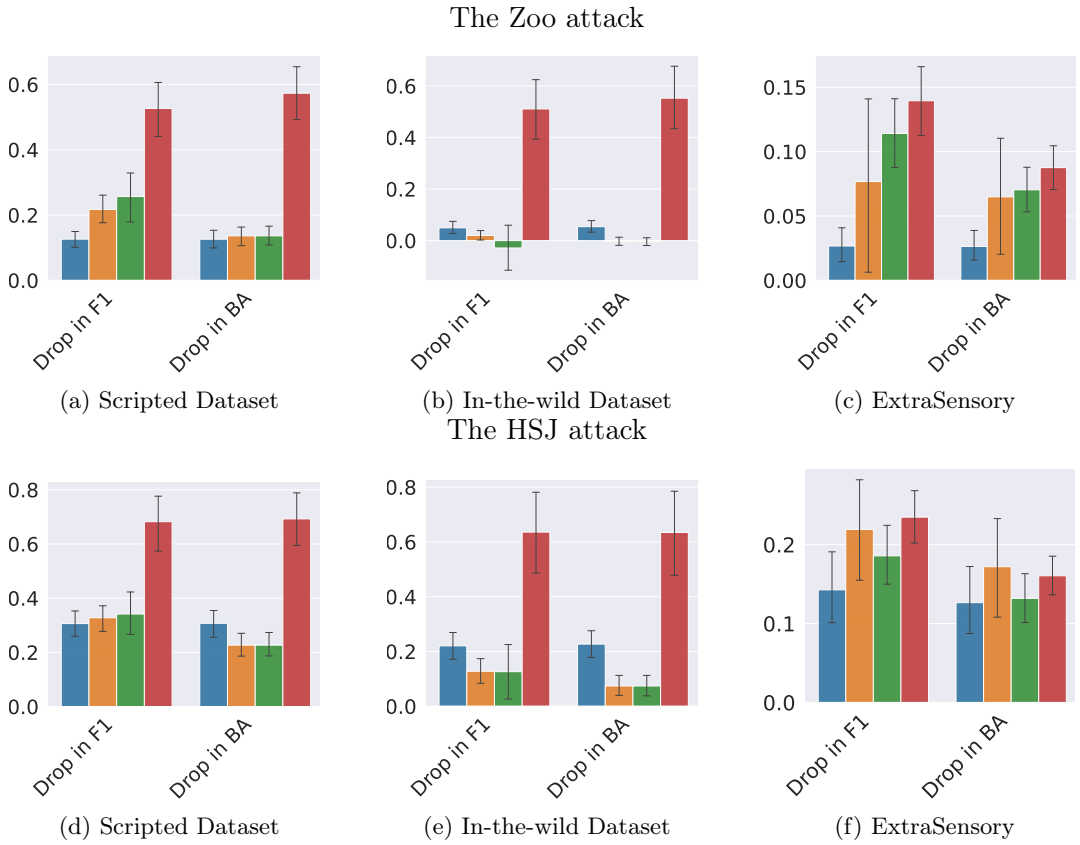
Figure 7.4: The overall performance results were obtained using the Zoo (top) & HSJ (bottom) (*higher is better*).

7.5.4 Evaluation Protocol

In order to improve the generalizability of our model to unseen participants, we utilized subject-wise cross-validation, where all of a subject's data appeared either in the training or test sets but not both. In each adversarial HCR experiment, we use 80% of the data to train the HCR and the remaining 20% data is used for testing. We only use the testing data to create adversarial examples that capture a real-world use case. Two metrics were used: Balanced accuracy and F-1 score.

Where balanced accuracy is defined as :

$$BA = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right)$$



(a) Scripted Dataset (b) In-the-wild Dataset (c) ExtraSensory

(d) Scripted Dataset (e) In-the-wild Dataset (f) ExtraSensory

Figure 7.5: The overall drop in performance under the Zoo (top) & HSJ (bottom) attacks (*lower is better*).

In each experiment, a binary classifier is trained for each label in the dataset. Then, adversarial examples are generated using one of the mentioned black-box attacks (Zoo and HSJ) using the testing split. Also, both the testing score and the adversarial score, defined as the following (for Balanced accuracy) are reported:

$$BA_{ADV} = BA_{\text{testing}} - BA_{\text{adversarial examples}}$$

7.5.5 Implementation

We implemented *RobustHCR* on a Dell R450 Poweredge server equipped with a single NVIDIA V100 GPU. For training our neural networks, we used the Pytorch auto differentiation framework. We used a feed-forward neural network

with two ReLU hidden layers, each consisting of 16 nodes, adapted from Vaizman *et al.*'s classifier used on the Extrasensory dataset [23]. For adversarial attack generation, we utilized the Adversarial Robustness toolbox by IBM to generate the Zoo and HSJ attacks [152]. The necessary parameters for producing an adversarial attack are as follows: Zoo attack(learning rate=1e-, max iter=20, binary search steps=10, initial const=1e-3, abort early=True, variable h=0.2) and HopSkipJump(norm = 2, max iter = 50, max eval = 10000, init eval = 100, init size = 100). All networks are optimized by a mini-batch Adam optimizer with a 0.01-percent learning rate and a momentum of 0.90. We ran a total of 100 epochs using a batch size of 64 in order to ensure sufficient training iterations for HCR tasks. For the adversarial defense, the open-source implementation of the duality-based method is used to compute the robust bounds for ReLU-based neural network: github.com/locuslab/convex_adversarial. Through empirical evaluation, we discovered the best size of the perturbation region to be robust against adversarial examples, which was set at 0.1.

7.5.6 Baselines

We compare our work to Adar [147] and RobustSense [142], the two closest methods proposed to defend against evasion attacks on sensor data. During training, Adar augmented adversarial examples generated by a white-box attack [147]. Yang *et al.* proposed RobustSense, which is designed to defend WiFi-based Human Activity Recognition against evasion attacks [142]. This method also falls under the adversarial training category as it augments adversarial examples during training and forces the model to generate consistent predictions for original inputs and

Table 7.3: Adversarial F-1 scores per label on the Scripted HCR dataset were obtained using the Zoo attack.

Method / Label	Proposed	Adar	RobustSense	Undefended
Bathroom	**0.62**	0.33	0.21	0.25
Coughing	**0.65**	0.40	0.36	0.30
Jogging	**0.65**	0.33	0.41	0.35
Jumping	**0.71**	0.33	0.41	0.30
Laying Down (action)	**0.67**	0.36	0.28	0.48
Lying Down	**0.66**	0.33	0.29	0.39
Phone in Bag	0.43	**0.53**	0.50	0.13
Phone in Hand	0.56	0.74	**0.88**	0.07
Phone in Pocket	0.53	**0.66**	0.44	0.11
Phone on Table	**0.57**	0.33	0.48	0.31
Running	**0.65**	0.33	0.27	0.29
Sitting	0.60	0.81	**0.85**	0.30
Sitting Down (action)	**0.70**	0.45	0.41	0.59
Sitting Up (action)	**0.67**	0.40	0.28	0.45
Sleeping	**0.65**	0.33	0.43	0.42
Sneezing	**0.67**	0.40	0.29	0.49
Stairs	**0.66**	0.34	0.17	0.34
Standing	**0.65**	0.33	0.17	0.41
Standing up (action)	**0.70**	0.44	0.57	0.46
Talking On Phone	**0.65**	0.33	0.25	0.25
Trembling	**0.61**	0.35	0.57	0.23
Typing	**0.63**	0.33	0.50	0.28
Walking	0.62	**0.86**	0.86	0.10

their adversarial variants by minimizing the sum of the Kullback–Leibler (KL) divergences $D_{KL}(p\|q) = \int_x p(x) \log \frac{p(x)}{q(x)} dx$ between pairs of original inputs and their adversarial form, which used to increase the model's robustness and gently minimize the prediction inconsistency. To ensure a fair comparison, we employed the Projected Gradient Descent (PGD) white-box attack [66] in both cases. We used the author's provided code to run on our datasets.

Table 7.4: Adversarial F-1 scores per label for the In-the-wild HCR dataset were obtained using the Zoo attack.

Method / Label	Proposed	Adar	RobustSense	Undefended
Bathroom	**0.65**	0.33	0.34	0.39
Exercising	**0.62**	0.33	0.49	0.29
Jogging	**0.65**	0.33	0.51	0.28
Lying down	0.36	**0.37**	0.21	0.28
Phone in Bag	**0.39**	0.38	0.35	0.29
Phone in Hand	**0.38**	0.33	0.09	0.30
Phone in Pocket	0.48	**0.62**	0.46	0.22
Phone on Table	0.60	**0.72**	0.62	0.22
Running	**0.63**	0.33	0.32	0.27
Sitting	0.53	0.60	**0.61**	0.24
Sleeping	0.62	0.65	**0.68**	0.28
Stairs	**0.65**	0.33	0.37	0.24
Standing	0.37	0.33	0.22	**0.38**
Talking on Phone	**0.69**	0.33	0.27	0.28
Typing	0.39	0.34	**0.50**	0.36
Walking	0.38	0.48	**0.50**	0.33

7.6 Results & Discussion

1) Overall Results: Figures (7.4) and (7.5) illustrate our overall findings. In Figure (7.4), we display the adversarial test scores (Adv F1, Adv BA), whereas in Figure (7.5), we use a computed value representing the amount of reduction when switching from clean to adversarial inputs (e.g. Drop in F1 = F1 - Adv F1). Generally, it can be observed that HSJ, which uses less information than the Zoo attack, clearly produced a higher impact and, specifically, a higher drop in performance (poor Adv F1 & Adv BA scores). The HSJ attack is significantly more effective than the Zoo attack because fewer queries are required to reach convergence. We obviously see that both sophisticated adversarial evasion attacks (Zoo and HSJ) can significantly impair the accuracy of undefended HCR models, resulting in a performance drop of up to 60(7.3) and (7.4), we present a sample of adversarial

scores reported per label by the Zoo attack on both scripted and in-the-wild HCR datasets. By achieving higher prediction scores for the original context labels, it is evident that our proposed method outperformed baseline methods in most cases.

2) Robustness trade-off: In Table (7.5), we evaluated varying the epsilon parameter for the duality-based network on the scripted dataset, controlling the robustness guarantees, where ε controls the size of l_1 norm ball to be robust against adversarial examples. Larger ε values reduce the amount of damage that could be caused by adversarial examples by sacrificing a bit on the performance on clean inputs.

3) Visualizing generated examples: In order to come up with a measure for imperceptibility and to evaluate the quality of the produced adversarial example utilized, we consider the two sets of clean vs. adversarial examples as two clusters, and we utilize a cluster evaluation score to test how similar these inputs are to each other. Specifically, we utilized the Silhouettescore Score $= \frac{b_i - a_i}{\max(b_i, a_i)}$, where b_i is the average distance that is the shortest between a point and every other point in any other cluster, whereas a_i is the mean distance that is between i and all data points that are from the same cluster. This score takes into consideration both compactness and separation. The visualization is illustrated in Figure (7.6) obtained by projecting feature embeddings into a two-dimensional space using the T-distributed Stochastic Neighbor Embedding (TSNE). In all cases, the generated adversarial examples in our proposed method are further away from the clean inputs, which means in case of an actual attack, there will be more likely to be rejected and classified as data from bad actors. Bad actors are users who carelessly (or maliciously) provide incorrect ground truth labels that do not reflect

actual events during data collection.

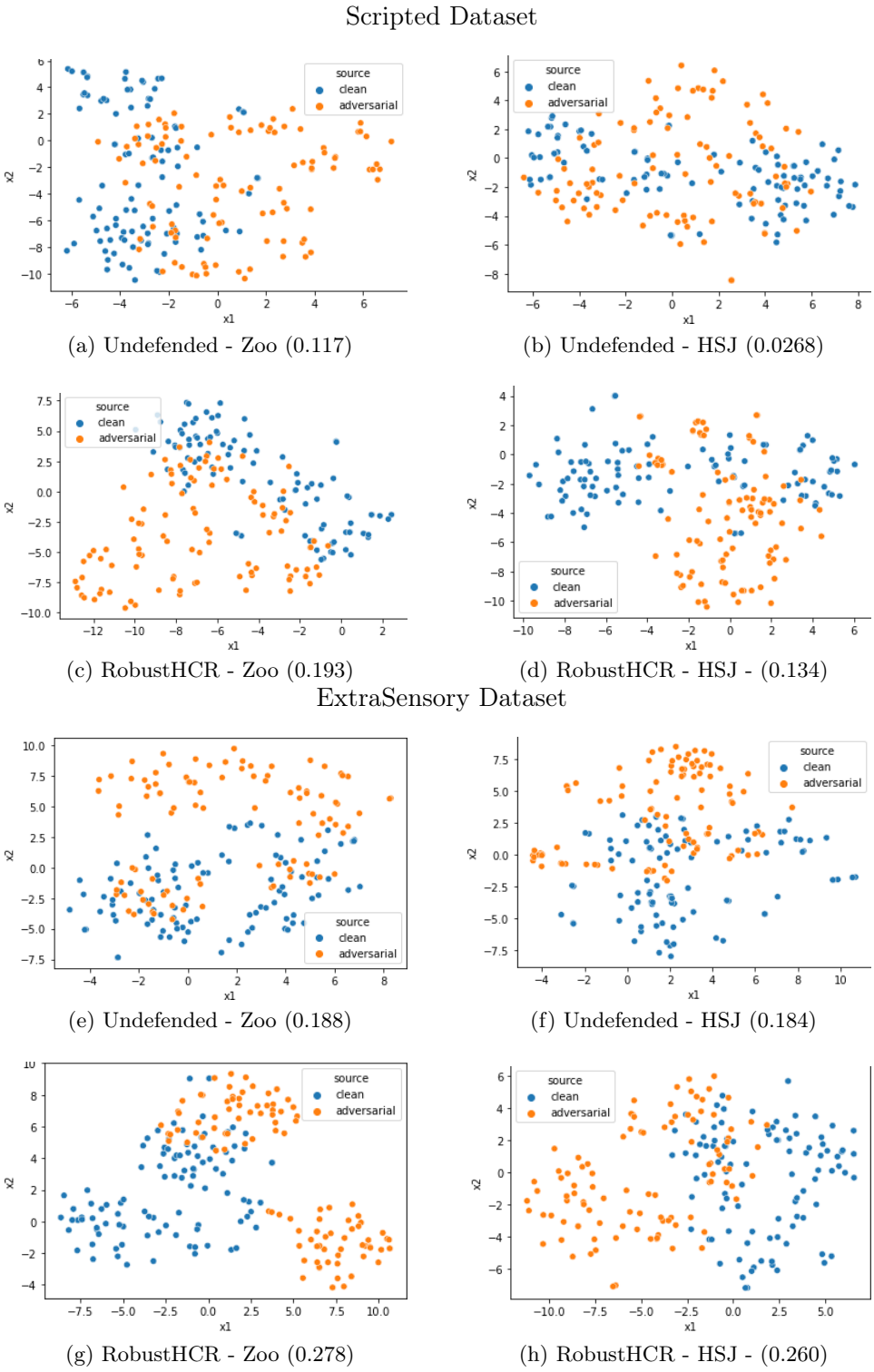
Our findings showed that even in a black-box environment without access to model parameters, smartphone HCRs are susceptible to evasion assaults. Through our empirical analyses of three smartphone HCR datasets, we also demonstrate the usefulness of our suggested defense technique in guarding against the evasion attack threat.

7.7 Limitations and Future Work

In this work, we assumed that an adversary could tamper with the sensor data before it was input to the HCR model, which is presumed to be located on a server in the cloud. One of the exciting problems we may want to explore in the future is the feasibility of these attacks running entirely on a smartphone device. We also relied on a tabular view of the raw sensor data by extracting handcrafted features, which were passed to a ReLU-based neural network. Additionally, future work will expand our threat model to include the possibility of poisoning during data collection in the wild and investigate corresponding defense strategies.

7.8 Chapter Summary

HCR models deployed in the wild are vulnerable to adversarial threats such as data poisoning, which targets the training stage, and evasion attacks that are perpetuated during model inference. In this study, we investigate adversarial evasion attacks and defenses for smartphone-based HCR systems. We formally describe and demonstrate the vulnerability of HCR systems to adversarial examples gener-



(a) Undefended - Zoo (0.117)

(b) Undefended - HSJ (0.0268)

(c) RobustHCR - Zoo (0.193)

(d) RobustHCR - HSJ - (0.134)

(e) Undefended - Zoo (0.188)

(f) Undefended - HSJ (0.184)

(g) RobustHCR - Zoo (0.278)

(h) RobustHCR - HSJ - (0.260)

Figure 7.6: t-SNE plots for clean vs. adversarial examples, with computed silhouette scores.

ated in a black-box environment under two assumptions: the availability of class confidence scores and the availability of only class decisions. The empirical find-

ings demonstrate that adversarial attacks can degrade the performance of deep models operating in HCR systems by up to 60

Table 7.5: Varying the epsilon parameter for the duality-based network tested on the scripted dataset, controlling the robustness guarantees, where ε controls the size of l_1 norm ball to be robust against adversarial examples. Larger ε values reduce the amount of damage that could be caused by adversarial examples by sacrificing a bit of the performance on clean inputs.

Dataset	ε	F1-score	Adv F1-score
Scripted	0.01	0.915	0.515
	0.05	0.953	0.697
	0.10	0.953	0.793
	0.15	0.942	0.841
	0.20	0.939	0.855
	0.25	0.939	0.878
	0.30	0.918	0.886
	0.35	0.905	0.877
	0.40	0.860	0.845
In-the-wild	0.01	0.758	0.478
	0.05	0.732	0.629
	0.10	0.716	0.697
	0.15	0.698	0.686
	0.20	0.682	0.675
	0.25	0.661	0.657
	0.30	0.664	0.650
	0.35	0.654	0.646
	0.40	0.627	0.627

Part V

Dissertation Findings

Chapter 8

Discussion and Findings

This chapter provides a summary of the research conducted for this dissertation and an outline for future work, and discusses our significant findings. As stated earlier in this dissertation, the overarching goal is to design practical solutions to the challenges, as mentioned earlier, using various *representation learning* techniques, a collection of techniques that enable automated discovery of the representations required for weakly-supervised machine learning tasks from raw data. A strong representation will help various downstream tasks of interest, such as classification.

8.1 Accomplished Research Work

Human Context Recognition under inexact supervision *DeepContext* has two significant innovations. First, *DeepContext* employs a joint-learning fusion strategy that utilizes both domain-specific handcrafted features and features that are autonomously generated by a Convolutional Neural Network (CNN). Second, *DeepContext* addresses the problem of coarse-grained la-

bels by discovering and giving higher importance to the most salient regions of the sensor data. These regions are expected to correspond to a higher predictive value for specific contexts. This allows our model to overcome potentially noisy inputs, which is achieved by *DeepContext's* parametrized compatibility-based attention mechanism [22].

Leveraging Coincident Context Data Gathering Study Scripted datasets have accurate context labels, but user behaviors are not realistic. In-the-wild datasets have realistic user behaviors but often have wrong or missing labels. We proposed two methods motivated by this fact that work in two settings: 1) Inaccurate supervision using *PUCL*: Positive Unlabeled Context Learning [26] and 2) Incomplete supervision using *Triple-DARE*: Triplet-based Domain Adaptation for Lab-to-field Human Context Recognition. *PUCL* uses a transductive positive unlabeled learning methodology to transfer knowledge from the highly-accurate labels of the scripted dataset to the less accurate, more sparse but yet more realistic in-the-wild dataset. *Triple-DARE* utilized a transductive transfer learning method with triplet loss to adapt neural networks in various domains to mitigate the covariate shift problem.

Adversarial-Robust Human Context Recognition *Evasion attacks* are mainly concerned with manipulated data intended to deceive pre-trained classifiers into misclassifications. We demonstrate model inference evasion attacks, including adversarially calibrated input perturbations to fool classifiers. Black-box evasion attacks only require arbitrary model queries, unlike white-box methods. *RobustHCR* is a novel duality-based network defense framework for black box evasion threats. *RobustHCR* reliably predicts whether its in-

put is under attack, minimizing adversarial attacks. *RobustHCR*'s rigorous evaluation of scripted and in-the-wild smartphone HCR datasets shows that it can significantly improve the HCR model's robustness and protection from evasion attacks while maintaining acceptable performance on clean inputs.

8.2 Example of an Application Use Case

Mobile-sensed data-based HCRs can be utilized for several use cases. Smartphone-based HCRs may be used to continuously track and monitor the health of soldiers or veterans in order to detect Traumatic Brain Injuries (TBI) or infectious diseases. (e.g., Covid-19). By monitoring smartphone biomarkers for health, abnormal user behavior, physiological indicators, activities, and context visit patterns can be identified. Additionally, in one of the most challenging contexts to detect, *Stairs - Going Up* and *Stairs - Going Down*, we can significantly outperform the other state-of-the-art methods with the help of our proposed method *DeepContext*. Detecting whether the subject is avoiding using stairs might provide valuable insights about their mobility levels, which could facilitate the identification of potential ailments [92].

8.3 Findings for Robust Feature Extraction

What distinguishes a typical classification task using sensory data from other data types (e.g., images or text) is the necessary preprocessing steps on its inputs before feeding them to the machine learning model. Sensory data is initially segmented using sliding windows to generate training instances, which are then input to the

feature generator model that extracts feature vectors utilized for context prediction later in the pipeline.

The size of the sliding window plays a critical role in acquiring a helpful representation. The choice of small windows (e.g., 2 or 3 seconds) might produce significantly higher prediction performance for predicting ambulation activities using sensory data. However, from an application perspective, accurate predictions for larger segments are far more helpful in context predictions, where we find significant improvements from using our proposed method, *DeepContext*. Also, having a better-performing method under larger sliding windows implies the ability to learn valuable features under significantly larger background noise. The user-provided ground-truth labeling becomes more coarse-grained and less accurately associated with the entire training example. The advantage of our proposed method comes from its attention mechanism to learn context-specific salient features and more effectively suppress background noise occurring in the sensor data. Additionally, we speculate that the significant improvements we get in our feature extraction method are due to the utilization of both deeplearning-based generated features and domain-specific handcrafted features.

8.4 Findings for Transferability of In-lab Models to the Real world

The contexts and patterns of visits in scripted datasets are not representative of the real world. It is essential that HCR models are accurate on datasets collected in the wild, which are more representative of actual deployment scenarios. However,

HCR models perform less well when trained directly on more realistic datasets collected in the wild. For example, using an HCR model trained directly on an in-the-wild dataset, Vaizman achieved 71.7% accuracy [23]. This represents a 19.5% decrease in the accuracy of state-of-the-art HCR models on scripted versus in-the-wild datasets, highlighting the difficulty of achieving robust, high HCR performance on in-the-wild datasets. Diversity of Causes (DoC) and labeling issues are specific challenges posed by in-the-wild data sets. Approaches that train a robust HCR model on a scripted dataset and then transfer it to an in-the-wild dataset face the additional difficulty of a covariate shift between the scripted and in-the-wild datasets.

When attempting to use models trained on scripted data to improve performance on an unscripted dataset with similar context labels, we encounter a data shift problem known as covariate shift, where the distribution of features differs between training and test scenarios. Specifically, the covariate shift problem results from significant differences in the distribution of features extracted from scripted versus in-the-wild datasets [35, 36, 37]. More generally, because real-world applications must deal with some dataset shifts, addressing the covariate shift problem is essential for successfully deploying machine learning models in the wild [35].

In-the-wild HCR data-gathering studies depend on self-reported labels. As users' lives become hectic, they may stop providing labels, or worse, they may provide incorrect labels [32]. Consequently, most collected sensor data are unlabeled, necessitating the development of unsupervised HCR models capable of utilizing unlabeled data and minimizing the effects of mislabeled data. We found that applying positive unlabeled classifiers that leverage high-fidelity scripted datasets

can reduce the impact of potential erroneous labeling found in in-the-wild datasets.

We significantly improved using DA methods to adapt pre-trained HCR models for making predictions under DoC data. Specifically, using a triplet-loss-based DA method help the model in finding a joint embedding space for not only reducing the global discrepancy between the two dataset feature distribution but also improving intra-class compactness and inter-class separability, which eventually improves context recognition on the target dataset.

8.5 Findings for Robust Representations under Adversarial Attacks

In the wild, HCR models are susceptible to adversarial attacks such as data poisoning, which targets the training phase or those that occur during model inference. Using adversarial threats as a robustness metric is also essential. Concerning the viability of these threats, it is essential to identify the types of attacks an adversary can conduct against a smartphone HCR classifier based on two threat models. A score-based threat model assumes access to class confidence scores, whereas a label-based model only assumes access to the predicted label. An adversary can submit arbitrary inputs to a pre-trained HCR classifier and observe its output (e.g., class confidence scores or only class labels). During data collection in the wild, an adversary may transmit sensor inputs and self-reported annotations. Evasion attacks can be used to create poisonous examples, specifically inputs that resemble valid inputs but have misleading labels. While our proposed methods can be helpful for both types of attacks, we only focused on examining evasion

attacks and defenses in this dissertation.

Even in a black-box environment without access to model parameters, smartphone HCRs were vulnerable to evasion attacks. Our empirical analyses of three smartphone HCR datasets demonstrate the efficacy of our proposed defense against the threat of evasion attacks.

8.6 Limitations of the Proposed Approaches

While our work is proposed for mobile-sensed HCRs, we evaluated our approaches on smartphone sensory data that has multiple modalities. The generation of sensor features from inputs with varying segment sizes is one area we did not investigate; instead, we fixed the segment sizes in one experiment.

Also, when we performed the transferability of scripted to in-the-wild models, we assumed the same sensors and number of extracted features were constant across the two datasets as well as the context labeling vector.

Additionally, for our adversarial robustness work, we assumed that an adversary could tamper with sensor data before it was input into the HCR model, which is presumed to reside on a cloud-based server. The current design of RobustHCR assumes that there is no limit on the number of queries that can be executed, and we have ignored the time required to generate adversarial examples.

8.7 Future Work

As part of future work, in order to reduce the reliance on human-annotated data, we will use Dual-GANs to generate synthetic samples designed for mobile-sensing data with DoC. To get over the restrictions of labeled datasets, we propose research methods to advance state-of-the-art representation learning techniques for mobile sensing data using contrastive-learning-based self-supervised learning techniques. We aim to design approaches to extract beneficial features from *unlabeled* context mobile sensing data.

The possibility of adversarial attacks functioning solely on a smartphone device is one of the interesting problems we may want to investigate in the future. Our threat model will be expanded in the future to incorporate the possibility of poisoning during data collection in the wild, and appropriate defense strategies will be investigated.

Human Context Generation Our method aims to adapt a generative model that has been trained on the source domain to produce synthetic samples in the target domain using few or no labeled samples from the target domain. We propose to use Dual-GANs with Elastic Weight Consolidation (EWC), a regularization technique designed to mitigate catastrophic forgetting in neural networks. This extends our previous work Triple-DARE, but we plan to apply the domain adaptation technique on deep generative models instead of context classification models.

Self-supervised Contrastive Learning In order to get over the restrictions of labeled datasets, we propose research methods to advance state-of-the-art

representation learning techniques for mobile sensing data using contrastive-learning-based SSL techniques. Our envisioned approach is designed to extract highly useful features from *unlabeled* context mobile sensing data. Our envisioned approach can also work with a provided supplementary labeled dataset to learn task-relevant features and suppress the noise related to DoC through contrastive learning. As a motivational use case, this approach could enable visual representation tools with rich sensory representations by leveraging self-supervised learning, which can be critical for analyzing sensory data on a large scale.

Chapter 9

Conclusion

Context awareness is crucial in enabling ubiquitous computing systems with optimal usability. Due to its ubiquity, smartphones will become more capable of providing contextual information about their users, known as *Human Context Recognition* (HCR), as technology becomes more integrated with the human experience. However, collecting smartphone sensor data in the wild often results in naturally occurring variations in the data. When leveraging models trained on scripted data, we encounter a data shift problem known as covariate shifts, where the distribution of features differs across training and test scenarios. Deploying such HCR models in the wild requires ensuring successful transferability from the lab to real-world applications by learning robust representations, which are hindered by several challenges residing in both inputs and labels due to the generation process of mobile-sensing data in addition to how annotations are acquired that is usually based on self-reporting.

This dissertation proposes designing practical solutions for filling research gaps in the mobile sensing domain, fusing various *representation learning techniques* in a

weakly-supervised-learning setting under *covariate shifts*. This endeavor attempts to integrate HCRs that utilize mobile-sensed data into the analytical and decision-making pipeline for a variety of motivational app use cases, including military deployment, mobile health, and assisted living. Specifically, 1) we propose state-of-the-art methods for sensor-based feature extractions with *DeepContext* that addresses the problem of coarse-grained labels by discovering and giving higher importance to the most salient regions of the sensor data using a parametrized compatibility-based attention mechanism. 2) We demonstrate that it is helpful to leverage scripted datasets with accurate context labels to improve context recognition in real-world applications through a coincident context data-gathering study in which the same contexts were collected. We proposed two methods motivated by this fact that work in two settings: a) Inaccurate supervision using *PUCL*: Positive Unlabeled Context Learning and b) Incomplete supervision using *Triple-DARE*: Triplet-based Domain Adaptation for Lab-to-field Human Context Recognition. *PUCL* uses a transductive positive unlabeled learning methodology to transfer knowledge from the highly-accurate labels of the scripted dataset to the less accurate, more sparse, yet more realistic in-the-wild dataset. *Triple-DARE* utilized a transductive transfer learning method with triplet loss to adapt neural networks in various domains to mitigate the covariate shift problem.

Finally, 3) We identify and propose defenses for potential adversarial threats to mobile-sensing in-the-wild gathering studies, which are *Evasion attacks*. We conclude our dissertation with *RobustHCR*, a defensive approach using robust optimizations by evaluating its performance against adversarial threats, increasing the model's robustness overall.

Bibliography

[1] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. "Towards a Better Understanding of Context and Context-Awareness." In: *Handheld and Ubiquitous Computing*. Vol. 1707. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 304–307 (cit. on p. 3).

[2] Mark Weiser. "The computer for the 21st century." In: *ACM SIGMOBILE mobile computing and communications review* 3.3 (1999), pp. 3–11 (cit. on p. 3).

[3] Albrecht Schmidt. *Ubiquitous computing-computing in context*. Lancaster University (United Kingdom), 2003 (cit. on p. 3).

[4] P. Rashidi and D.J. Cook. "Keeping the Resident in the Loop: Adapting the Smart Home to the User." In: *IEEE Trans.Sys., Man, and Cybernetics - Part A: Systems and Humans* 39.5 (Sept. 2009), pp. 949–959 (cit. on p. 3).

[5] Parisa Rashidi and Alex Mihailidis. "A Survey on Ambient-Assisted Living Tools for Older Adults." In: *IEEE Journal of Biomedical and Health Informatics* 17.3 (May 2013), pp. 579–590 (cit. on p. 3).

[6] Mashfiqui Rabbi, Predrag Klasnja, Maureen Walton, Susan Murphy, Meredith Philyaw-Kotov, Jinseok Lee, Anthony Mansour, Laura Dent, Xiaolei Wang, Rebecca Cunningham, Erin Bonar, and Inbal Nahum-Shani. "SARA: a mobile app to engage users in health data collection." en. In: *Proc. ACM UbiComp '17*. Maui, Hawaii, 2017, pp. 781–789 (cit. on p. 3).

[7] Alfredo J. Perez, Miguel A. Labrador, and Sean J. Barbeau. "G-Sense: a scalable architecture for global sensing and monitoring." In: *IEEE Network* 24 (2010) (cit. on p. 3).

[8] Bruno M.C. Silva, Joel J.P.C. Rodrigues, Isabel de la Torre Díez, Miguel López-Coronado, and Kashif Saleem. "Mobile-health: A review of current state in 2015." en. In: *J. Biomed. Inf.* 56 (Aug. 2015), pp. 265–272 (cit. on p. 3).

[9] Youngki Lee, Junehwa Song, Chulhong Min, Chanyou Hwang, Jaeung Lee, Inseok Hwang, Younghyun Ju, Chungkuk Yoo, Miri Moon, and Uichin Lee. "SocioPhone: everyday face-to-face interaction monitoring platform using multi-phone sensor fusion." en. In: *Proc. ACM MobiSys '13*. Taipei, Taiwan: ACM Press, 2013, p. 375 (cit. on p. 3).

[10] Mashfiqui Rabbi Xiaochao Yang Hong Lu Giuseppe Cardone Shahid Ali Afsaneh Doryab Ethan Berke Andrew Campbell Tanzeem Choudhury. Mu Lin Nicholas D. Lane. "BeWell+: Multi-dimensional Wellbeing Monitoring with Community-guided User Feedback and Energy Optimization." In: *Wireless Health 2012* (Oct. 2012) (cit. on pp. 3, 4).

[11] Kelly Servick. *Mind the phone.* 2015 (cit. on p. 4).

[12] Mathias Basner, Kenneth M Fomberstein, Farid M Razavi, Siobhan Banks, Jeffrey H William, Roger R Rosa, and David F Dinges. "American time use survey: sleep time and its relationship to waking activities." In: *Sleep* 30.9 (2007), pp. 1085–1095 (cit. on p. 4).

[13] David E Bloom, Elizabeth Cafiero, Eva Jané-Llopis, Shafika Abrahams-Gessel, Lakshmi Reddy Bloom, Sana Fathima, Andrea B Feigl, Tom Gaziano, Ali Hamandi, Mona Mowafi, et al. *The global economic burden of noncommunicable diseases.* Tech. rep. Program on the Global Demography of Aging, 2012 (cit. on p. 4).

[14] Yonatan Vaizman, Katherine Ellis, and Gert Lanckriet. "Recognizing Detailed Human Context in the Wild from Smartphones and Smartwatches." en. In: *IEEE Pervasive Computing* 16.4 (Oct. 2017), pp. 62–74 (cit. on pp. 4, 7, 11, 12, 16, 32, 35, 44, 51, 83, 84, 105, 122, 124).

[15] Andrew Perrin. *Mobile Technology and Home Broadband 2021.* 2021 (cit. on p. 4).

[16] Published by S. O'Dea and Feb 17. *Smartphone users 2026.* 2022 (cit. on p. 4).

[17] DARPA. *DARPA WASH BAA.* (accessed: 07.06.2020). URL: `https://beta.sam.gov/opp/cfb9742c60d055931003e6386d98c044/view` (cit. on pp. 5, 27).

[18] Pierluigi Casale, Oriol Pujol, and Petia Radeva. "Personalization and user verification in wearable systems using biometric walking patterns." In: *Personal and Ubiquitous Computing* 16.5 (2012), pp. 563–580 (cit. on p. 7).

[19] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. "A public domain dataset for human activity recognition using smartphones." In: *Esann*. Vol. 3. 2013, p. 3 (cit. on p. 7).

[20] Daniela Micucci, Marco Mobilio, and Paolo Napoletano. "Unimib shar: A dataset for human activity recognition using acceleration data from smartphones." In: *Applied Sciences* 7.10 (2017), p. 1101 (cit. on p. 7).

[21] Henrik Blunck, Sourav Bhattacharya, Allan Stisen, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Mads Møller Jensen, and Tobias Sonne. "ACTIVITY RECOGNITION ON SMART DEVICES: Dealing with Diversity in the Wild." In: *GetMobile: Mobile Comp. and Comm.* 20.1 (July 2016), 34–38 (cit. on p. 7).

[22] A. Alajaji, W. Gerych, K. Chandrasekaran, L. Buquicchio, E. Agu, and E. Rundensteiner. "Deep-Context: Parameterized Compatibility-Based Attention CNN for Human Context Recognition." In: *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*. 2020, pp. 53–60 (cit. on pp. 10, 20, 32, 34, 69, 83, 93, 136).

[23] Yonatan Vaizman, Nadir Weibel, and Gert Lanckriet. "Context recognition in-the-wild: Unified model for multi-modal sensors and multi-label classification." In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1.4 (2018), pp. 1–22 (cit. on pp. 10, 34, 53, 83, 127, 139).

[24] Youngjae Chang, Akhil Mathur, Anton Isopoussu, Junehwa Song, and Fahim Kawsar. "A Systematic Study of Unsupervised Domain Adaptation for Robust Human-Activity Recognition." In: *ACM IMWUT* 4 (Mar. 2020), pp. 1–30 (cit. on pp. 11, 36–39, 76, 77, 106).

[25] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Prentow, Mikkel Kjærgaard, Anind Dey, Tobias Sonne, and Mads Jensen. "Smart Devices are Different: Assessing and Mitigating Mobile Sensing Heterogeneities for Activity Recognition." In: *Proc. Sensys.* Nov. 2015, pp. 127–140 (cit. on pp. 12, 39, 106).

[26] Abdulaziz Alajaji, Walter Gerych, Luke Buquicchio, Kavin Chandrasekaran, Hamid Mansoor, E. Agu, and Elke A. Rundensteiner. "Smartphone Health Biomarkers: Positive Unlabeled Learning of In-the-Wild Contexts." In: *IEEE Pervasive Computing* 20 (2021), pp. 50–61 (cit. on pp. 12, 16, 17, 20, 32–35, 64, 65, 105, 110, 136).

[27] Oscar D. Lara and Miguel A. Labrador. "A Survey on Human Activity Recognition using Wearable Sensors." In: *IEEE Communications Surveys & Tutorials* 15.3 (2013), pp. 1192–1209 (cit. on pp. 12, 16, 34, 44).

[28] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. "Deep Learning for Sensor-based Activity Recognition: A Survey." In: *Pattern Recognition Letters* 119 (Mar. 2019). arXiv: 1707.03502, pp. 3–11 (cit. on pp. 12, 16, 34).

[29] Aaqib Saeed, Tanir Ozcelebi, and Johan Lukkien. "Synthesizing and Reconstructing Missing Sensory Modalities in Behavioral Context Recognition." In: *Sensors* 18 (Sept. 2018), p. 2967 (cit. on p. 12).

[30] Yonatan Vaizman, Katherine Ellis, Gert Lanckriet, and Nadir Weibel. "ExtraSensory App: Data Collection In-the-Wild with Rich User Interface to Self-Report Behavior." en. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. Montreal QC Canada: ACM, Apr. 2018, pp. 1–12 (cit. on p. 12).

[31] Zhi-Hua Zhou. "A brief introduction to weakly supervised learning." In: *National Science Review* 5.1 (2018), pp. 44–53 (cit. on p. 13).

[32] H. Mansoor, W. Gerych, L. Buquicchio, K. Chandrasekaran, E. Agu, and E. Rundensteiner. "DELFI: Mislabelled Human Context Detection Using Multi-Feature Similarity Linking." In: *2019 IEEE VDS*. 2019, pp. 11–19 (cit. on pp. 13, 23, 139).

[33] Zhi-Hua Zhou. "A brief introduction to weakly supervised learning." en. In: *National Science Review* 5.1 (Jan. 2018), pp. 44–53 (cit. on p. 14).

[34] Zhi-Hua Zhou. "A brief introduction to weakly supervised learning." en. In: *National Science Review* 5.1 (Jan. 2018), pp. 44–53 (cit. on pp. 14, 28, 30, 44).

[35] "A unifying view on dataset shift in classification." In: 45.1 (2012), pp. 521–530 (cit. on pp. 14, 25, 76, 139).

[36] Wouter M. Kouw. "An introduction to domain adaptation and transfer learning." In: *ArXiv* abs/1812.11806 (2018) (cit. on pp. 14, 36, 76, 139).

[37] Annamalai Natarajan, Gustavo Angarita, Edward Gaiser, Robert Malison, Deepak Ganesan, and Benjamin M. Marlin. "Domain Adaptation Methods for Improving Lab-to-Field Generalization

of Cocaine Detection Using Wearable ECG." In: *Proc. Ubicomp*. Heidelberg, Germany: ACM, 2016, 875–885 (cit. on pp. 14, 75–77, 80, 139).

[38] Nicolas Papernot, Fartash Faghri, Nicholas Carlini, Ian Goodfellow, Reuben Feinman, Alexey Kurakin, Cihang Xie, Yash Sharma, Tom Brown, Aurko Roy, et al. "Technical report on the cleverhans v2. 1.0 adversarial examples library." In: *arXiv preprint arXiv:1610.00768* (2016) (cit. on pp. 14, 15).

[39] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." In: *arXiv preprint arXiv:1412.6572* (2014) (cit. on p. 14).

[40] Robert Geirhos, J. Jacobsen, Claudio Michaelis, R. Zemel, Wieland Brendel, M. Bethge, and Felix Wichmann. "Shortcut Learning in Deep Neural Networks." In: *Nature Machine Intelligence* (2020) (cit. on pp. 15, 16, 26, 108).

[41] Yingzhe He, Guozhu Meng, Kai Chen, Xingbo Hu, and Jinwen He. "Towards security threats of deep learning systems: A survey." In: *IEEE Transactions on Software Engineering* (2020) (cit. on pp. 15, 16, 40, 107).

[42] AKM Iqtidar Newaz, Nur Imtiazul Haque, Amit Kumar Sikder, Mohammad Ashiqur Rahman, and A Selcuk Uluagac. "Adversarial attacks to machine learning-based smart healthcare systems." In: *Proc. Globacom*. IEEE. 2020, pp. 1–6 (cit. on p. 15).

[43] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. "Deepfool: a simple and accurate method to fool deep neural networks." In: *Proc. IEEE CVPR*. 2016, pp. 2574–2582 (cit. on pp. 15, 39, 106).

[44] Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. "Adversarial attacks on deep-learning models in natural language processing: A survey." In: *ACM Trans. Intelligent Systems and Technology (TIST)* 11.3 (2020), pp. 1–41 (cit. on pp. 15, 39, 106).

[45] Nicholas Carlini and David Wagner. "Audio adversarial examples: Targeted attacks on speech-to-text." In: *Security and Privacy Workshops (SPW)*. IEEE. 2018, pp. 1–7 (cit. on pp. 15, 39, 106).

[46] Samuel Harford, Fazle Karim, and Houshang Darabi. "Adversarial attacks on multivariate time series." In: *arXiv preprint arXiv:2004.00410* (2020) (cit. on pp. 15, 39, 106).

[47] Fazle Karim, Somshubra Majumdar, and Houshang Darabi. "Adversarial attacks on time series." In: *Trans. pattern analysis and machine intelligence* (2020) (cit. on pp. 15, 39, 106).

[48] Izaskun Oregi, Javier Del Ser, Aritz Perez, and Jose A Lozano. "Adversarial sample crafting for time series classification with elastic similarity measures." In: *International Symposium on Intelligent and Distributed Computing*. Springer. 2018, pp. 26–39 (cit. on pp. 15, 39, 106).

[49] Cezara Benegui and Radu Tudor Ionescu. "Adversarial Attacks on Deep Learning Systems for User Identification based on Motion Sensors." In: *Proc. NIPS*. Springer. 2020, pp. 752–761 (cit. on pp. 15, 39, 106).

[50] Ramesh Kumar Sah and Hassan Ghasemzadeh. "Adversarial Transferability in Wearable Sensor Systems." In: *arXiv:2003.07982 [cs, eess, stat]* (July 2021). arXiv: 2003.07982 (cit. on pp. 15–17, 39, 106, 110, 113, 117).

[51] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli. "Towards poisoning of deep learning algorithms with back-gradient optimization." In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. 2017, pp. 27–38 (cit. on p. 15).

[52] Chen Wang, Jian Chen, Yang Yang, Xiaoqiang Ma, and Jiangchuan Liu. "Poisoning attacks and countermeasures in intelligent networks: Status quo and prospects." en. In: *Digital Communications and Networks* (July 2021) (cit. on p. 15).

[53] Marco Melis, Ambra Demontis, Battista Biggio, Gavin Brown, Giorgio Fumera, and Fabio Roli. "Is deep learning safe for robot vision? adversarial examples against the icub humanoid." In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2017, pp. 751–759 (cit. on pp. 16, 40, 107).

[54] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. "Evasion attacks against machine learning at test time." In: *Joint European conference on machine learning and knowledge discovery in databases*. Springer. 2013, pp. 387–402 (cit. on pp. 16, 17, 113).

[55] Battista Biggio and Fabio Roli. "Wild patterns: Ten years after the rise of adversarial machine learning." en. In: *Pattern Recognition* 84 (Dec. 2018), pp. 317–331 (cit. on pp. 16, 17, 113, 114).

[56] Im Y Jung. "A review of privacy-preserving human and human activity recognition." In: *Int'l Journal on Smart Sensing & Intelligent Systems* 13.1 (2020) (cit. on p. 16).

[57] Hamid Mansoor, Walter Gerych, Luke Buquicchio, Kavin Chandrasekaran, Emmanuel Agu, and Elke Rundensteiner. "DELFI: Mislabelled Human Context Detection Using Multi-Feature Similarity Linking." In: *2019 IEEE Visualization in Data Science (VDS)*. Vancouver, BC, Canada: IEEE, Oct. 2019, pp. 11–19 (cit. on p. 17).

[58] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, et al. "Collecting complex activity datasets in highly rich networked sensor environments." In: *Proc. INSS 2010*. IEEE. 2010, pp. 233–240 (cit. on p. 22).

[59] Yonatan Vaizman, Katherine Ellis, and Gert Lanckriet. "Recognizing detailed human context in the wild from smartphones and smartwatches." In: *IEEE Pervasive Computing* 16.4 (2017), pp. 62–74 (cit. on pp. 22, 39, 40, 63, 71, 106).

[60] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate." In: *arXiv:1409.0473 [cs, stat]* (May 2016). arXiv: 1409.0473 (cit. on pp. 23, 48).

[61] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention." In: *arXiv:1502.03044 [cs]* (Apr. 2016). arXiv: 1502.03044 (cit. on pp. 23, 24).

[62] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. "Learning Deep Features for Discriminative Localization." In: *arXiv:1512.04150 [cs]* (Dec. 2015). arXiv: 1512.04150 (cit. on p. 23).

[63] Hidetoshi Shimodaira. "Improving predictive inference under covariate shift by weighting the log-likelihood function." In: *Journal of statistical planning and inference* 90.2 (2000), pp. 227–244 (cit. on p. 25).

[64] Alexander Robey, Hamed Hassani, and George J Pappas. "Model-based robust deep learning: Generalizing to natural, out-of-distribution data." In: *arXiv preprint arXiv:2005.10247* (2020) (cit. on p. 26).

[65] Christina Heinze-Deml and Nicolai Meinshausen. "Conditional variance penalties and domain shift robustness." In: *arXiv preprint arXiv:1710.11469* (2017) (cit. on p. 26).

[66] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. "Towards deep learning models resistant to adversarial attacks." In: *arXiv preprint arXiv:1706.06083* (2017) (cit. on pp. 26, 128).

[67] Nathan Drenkow, Numair Sani, Ilya Shpitser, and Mathias Unberath. "A systematic review of robustness in deep learning for computer vision: Mind the gap?" In: *arXiv preprint arXiv:2112.00639* (2021) (cit. on p. 26).

[68] Gayatri Sravanthi Kuntla, Xin Tian, and Zhigang Li. "Security and privacy in machine learning: A survey." In: *Issues in Information Systems* 22.3 (2021) (cit. on p. 26).

[69] Nic Ford, Justin Gilmer, Nicolas Carlini, and Dogus Cubuk. "Adversarial examples are a natural consequence of test error in noise." In: *arXiv preprint arXiv:1901.10513* (2019) (cit. on p. 26).

[70] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. "Adversarial examples are not bugs, they are features." In: *Advances in neural information processing systems* 32 (2019) (cit. on p. 26).

[71] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. "Intriguing properties of neural networks." In: *arXiv preprint arXiv:1312.6199* (2013) (cit. on pp. 26, 105).

[72] Khansa Rasheed, Adnan Qayyum, Mohammed Ghaly, Ala Al-Fuqaha, Adeel Razi, and Junaid Qadir. "Explainable, trustworthy, and ethical machine learning for healthcare: A survey." In: *Computers in Biology and Medicine* (2022), p. 106043 (cit. on p. 26).

[73] Bo Li, Peng Qi, Bo Liu, Shuai Di, Jingen Liu, Jiquan Pei, Jinfeng Yi, and Bowen Zhou. "Trustworthy ai: From principles to practices." In: *ACM Computing Surveys* 55.9 (2023), pp. 1–46 (cit. on p. 26).

[74] Brendon G Anderson, Tanmay Gautam, and Somayeh Sojoudi. "An Overview and Prospective Outlook on Robust Training and Certification of Machine Learning Models." In: *arXiv preprint arXiv:2208.07464* (2022) (cit. on p. 26).

[75] Anind K Dey, Katarzyna Wac, Denzil Ferreira, Kevin Tassini, Jin-Hyuk Hong, and Julian Ramos. "Getting closer: an empirical investigation of the proximity of user to their smart phones." In: *Proceedings of the 13th international conference on Ubiquitous computing*. 2011, pp. 163–172 (cit. on p. 32).

[76] Raghu Kiran Ganti, Soundararajan Srinivasan, and Aca Gacic. "Multisensor fusion in smartphones for lifestyle monitoring." In: *2010 International Conference on Body Sensor Networks*. IEEE. 2010, pp. 36–43 (cit. on p. 32).

[77] Adil Mehmood Khan, Ali Tufail, Asad Masood Khattak, and Teemu H Laine. "Activity recognition on smartphones via sensor-fusion and KDA-based SVMs." In: *International Journal of Distributed Sensor Networks* 10.5 (2014), p. 503291 (cit. on p. 32).

[78] Yujie Dong, Jenna Scisco, Mike Wilson, Eric Muth, and Adam Hoover. "Detecting periods of eating during free-living by tracking wrist motion." In: *IEEE journal of biomedical and health informatics* 18.4 (2013), pp. 1253–1260 (cit. on p. 32).

[79] Jianbo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiaoli Li, and Shonali Krishnaswamy. "Deep convolutional neural networks on multichannel time series for human activity recognition." In: *IJCAI*. Vol. 15. Buenos Aires, Argentina. 2015, pp. 3995–4001 (cit. on p. 33).

[80] Nils Y Hammerla, Shane Halloran, and Thomas Plötz. "Deep, Convolutional, and Recurrent Models for Human Activity Recognition Using Wearables." In: *IJCAI*. arXiv: 1604.08880. Apr. 2016 (cit. on p. 33).

[81] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. "Deep learning for sensor-based activity recognition: A survey." en. In: *Pattern Recognition Letters* 119 (Mar. 2019), pp. 3–11 (cit. on p. 33).

[82] Charissa Ann Ronao and Sung-Bae Cho. "Human activity recognition with smartphone sensors using deep learning neural networks." en. In: *Expert Systems with Applications* 59 (Oct. 2016), pp. 235–244 (cit. on pp. 33, 110).

[83] Zehao Sun, Shaojie Tang, He Huang, Zhenyu Zhu, Hansong Guo, Yu-e Sun, and Liusheng Huang. "SOS: Real-time and accurate physical assault detection using smartphone." In: *Peer-to-Peer Networking and Applications* 10.2 (2017), pp. 395–410 (cit. on pp. 33, 110).

[84] Jorge-L. Reyes-Ortiz, Luca Oneto, Albert Samà, Xavier Parra, and Davide Anguita. "Transition-Aware Human Activity Recognition Using Smartphones." en. In: *Neurocomputing* 171 (Jan. 2016), pp. 754–767 (cit. on pp. 34, 50, 51, 84).

[85] Wanmin Wu, Sanjoy Dasgupta, Ernesto E Ramirez, Carlyn Peterson, and Gregory J Norman. "Classification accuracies of physical activities using smartphone motion sensors." In: *Journal of medical Internet research* 14.5 (2012), e130 (cit. on p. 34).

[86] Seyed Amir Hoseini-Tabatabaei, Alexander Gluhak, and Rahim Tafazolli. "A survey on smartphone-based systems for opportunistic user context recognition." en. In: *ACM Computing Surveys* 45.3 (June 2013), pp. 1–51 (cit. on pp. 34, 44).

[87] Jianbo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiaoli Li, and Shonali Krishnaswamy. "Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition." In: *IJCAI*. 2015 (cit. on p. 34).

[88] Valentin Radu, Catherine Tong, Sourav Bhattacharya, Nicholas D. Lane, Cecilia Mascolo, Mahesh K. Marina, and Fahim Kawsar. "Multimodal Deep Learning for Activity and Context Recognition." en. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1.4 (Jan. 2018), pp. 1–27 (cit. on p. 34).

[89] Nils Y. Hammerla, Shane Halloran, and Thomas Ploetz. "Deep, Convolutional, and Recurrent Models for Human Activity Recognition using Wearables." In: *arXiv:1604.08880 [cs, stat]* (Apr. 2016). arXiv: 1604.08880 (cit. on p. 34).

[90] Francisco Ordóñez and Daniel Roggen. "Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition." en. In: *Sensors* 16.1 (Jan. 2016), p. 115 (cit. on p. 34).

[91] Seyed Amir Hoseini-Tabatabaei, Alexander Gluhak, and Rahim Tafazolli. "A survey on smartphone-based systems for opportunistic user context recognition." en. In: *ACM Computing Surveys* 45.3 (June 2013), pp. 1–51 (cit. on p. 34).

[92] Oscar D. Lara and Miguel A. Labrador. "A Survey on Human Activity Recognition using Wearable Sensors." In: *IEEE Communications Surveys & Tutorials* 15.3 (2013), pp. 1192–1209 (cit. on pp. 34, 58, 137).

[93] Chi Ian Tang, Ignacio Perez-Pozuelo, Dimitris Spathis, Soren Brage, Nick Wareham, and Cecilia Mascolo. "SelfHAR: Improving Human Activity Recognition through Self-Training with Unlabeled Data." In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5.1 (Mar. 2021) (cit. on p. 35).

[94] Aaqib Saeed, Tanir Ozcelebi, Stojan Trajanovski, and Johan Lukkien. "Learning behavioral context recognition with multi-stream temporal convolutional networks." en. In: *arXiv:1808.08766 [cs, stat]* (Aug. 2018). arXiv: 1808.08766 (cit. on pp. 35, 44).

[95] Aaqib Saeed, Tanir Ozcelebi, and Johan Lukkien. "Multi-task Self-Supervised Learning for Human Activity Detection." In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3.2 (June 2019). arXiv: 1907.11879, pp. 1–30 (cit. on p. 35).

[96] Bulat Khaertdinov, Esam Ghaleb, and Stylianos Asteriadis. "Contrastive Self-supervised Learning for Sensor-based Human Activity Recognition." In: *2021 IEEE International Joint Conference on Biometrics (IJCB)*. ISSN: 2474-9699. Aug. 2021, pp. 1–8 (cit. on p. 35).

[97] Rebecca Adaimi and Edison Thomaz. "Leveraging active learning and conditional mutual information to minimize data annotation in human activity recognition." In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3.3 (2019), pp. 1–23 (cit. on p. 35).

[98] Harish Haresamudram, Irfan Essa, and Thomas Plötz. "Contrastive Predictive Coding for Human Activity Recognition." en. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5.2 (June 2021), pp. 1–26 (cit. on p. 35).

[99] Fantine Mordelet and J-P Vert. "A bagging SVM to learn from positive and unlabeled examples." In: *Pattern Recognition Letters* 37 (2014), pp. 201–209 (cit. on p. 35).

[100] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. "Learning Transferable Features with Deep Adaptation Networks." In: *ArXiv* abs/1502.02791 (2015) (cit. on pp. 36, 38, 77, 79, 86, 93).

[101] Baochen Sun and Kate Saenko. "Deep CORAL: Correlation Alignment for Deep Domain Adaptation." In: *ECCV Workshops*. 2016 (cit. on pp. 36, 38, 77, 79, 93).

[102] Yaroslav Ganin, E. Ustinova, Hana Ajakan, Pascal Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. "Domain-Adversarial Training of Neural Networks." In: *J. Mach. Learn. Res.* 17 (2016), 59:1–59:35 (cit. on pp. 36, 93).

[103] Md Abdullah Al Hafiz Khan, Nirmalya Roy, and Archan Misra. "Scaling Human Activity Recognition via Deep Learning-based Domain Adaptation." en. In: *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. Athens: IEEE, Mar. 2018, pp. 1–9 (cit. on pp. 36–38, 77, 79, 80, 93).

[104] Yiqiang Chen, Jindong Wang, Meiyu Huang, and Han Yu. "Cross-position activity recognition with stratified transfer learning." en. In: *Pervasive and Mobile Computing* 57 (July 2019), pp. 1–13 (cit. on pp. 36–38).

[105] Andrea Rosales Sanabria and Juan Ye. "Unsupervised domain adaptation for activity recognition across heterogeneous datasets." en. In: *Pervasive and Mobile Computing* 64 (Apr. 2020), p. 101147 (cit. on pp. 36–38).

[106] Garrett Wilson, Janardhan Rao Doppa, and Diane J. Cook. "Multi-Source Deep Domain Adaptation with Weak Supervision for Time-Series Sensor Data." en. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Virtual Event CA USA: ACM, Aug. 2020, pp. 1768–1778 (cit. on pp. 37, 38).

[107] Weijian Deng, Liang Zheng, and Jianbin Jiao. "Domain Alignment with Triplets." In: (Dec. 2018) (cit. on pp. 37, 76, 77, 89).

[108] Florian Schroff, D. Kalenichenko, and James Philbin. "FaceNet: A unified embedding for face recognition and clustering." In: *IEEE CVPR* (2015), pp. 815–823 (cit. on pp. 37, 77, 88–90).

[109] Annamalai Natarajan, Gustavo Angarita, Edward Gaiser, Robert Malison, Deepak Ganesan, and Benjamin M. Marlin. "Domain adaptation methods for improving lab-to-field generalization of cocaine detection using wearable ECG." en. In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. Heidelberg Germany: ACM, Sept. 2016, pp. 875–885 (cit. on p. 38).

[110] Benoît Frénay and Michel Verleysen. "Classification in the presence of label noise: a survey." In: *IEEE transactions on neural networks and learning systems* 25.5 (2013), pp. 845–869 (cit. on pp. 39, 63).

[111] Mattia Zeni, Wanyi Zhang, Enrico Bignotti, Andrea Passerini, and Fausto Giunchiglia. "Fixing Mislabeling by Human Annotators Leveraging Conflict Resolution and Prior Knowledge." In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3.1 (Mar. 2019) (cit. on pp. 39, 63).

[112] Eric Wong and Zico Kolter. "Provable defenses against adversarial examples via the convex outer adversarial polytope." In: *Proc. ICML*. PMLR. 2018, pp. 5286–5295 (cit. on pp. 39, 106, 109, 111, 112, 116, 117, 120).

[113] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. "Adversarial examples: Attacks and defenses for deep learning." In: *IEEE Trans. neural networks and learning systems* 30.9 (2019), pp. 2805–2824 (cit. on pp. 40, 107).

[114] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models." In: *Proc. ACM workshop on artificial intelligence and security*. 2017, pp. 15–26 (cit. on pp. 40, 107, 110–112, 116, 117).

[115] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. "Hopskipjumpattack: A query-efficient decision-based attack." In: *2020 ieee symposium on security and privacy (sp)*. IEEE. 2020, pp. 1277–1294 (cit. on pp. 40, 107, 110–112, 116–118).

[116] Yonatan Vaizman, Nadir Weibel, and Gert R. G. Lanckriet. "Context Recognition In-the-Wild: Unified Model for Multi-Modal Sensors and Multi-Label Classification." In: *IMWUT* 1 (2017), 168:1–168:22 (cit. on pp. 45, 52, 55).

[117] Saumya Jetley, Nicholas A. Lord, Namhoon Lee, and Philip H. S. Torr. "Learn To Pay Attention." In: *arXiv:1804.02391 [cs]* (Apr. 2018). arXiv: 1804.02391 (cit. on pp. 45, 47–49, 58, 83, 94).

[118] Frédéric Li, Kimiaki Shirahama, Muhammad Nisar, Lukas Köping, and Marcin Grzegorzek. "Comparison of Feature Learning Methods for Human Activity Recognition Using Wearable Sensors." en. In: *Sensors* 18.3 (Feb. 2018), p. 679 (cit. on p. 46).

[119] Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, and Tarek F. Abdelzaher. "DeepSense: A Unified Deep Learning Framework for Time-Series Mobile Sensing Data Processing." In: *WWW*. 2016 (cit. on pp. 46, 48, 55, 59).

[120] Kun Wang, Jun He, and Lei Zhang. "Attention-based Convolutional Neural Network for Weakly Labeled Human Activities Recognition with Wearable Sensors." In: *IEEE Sensors Journal* 19.17 (Sept. 2019). arXiv: 1903.10909, pp. 7598–7604 (cit. on pp. 48, 52, 59).

[121] Valentin Radu, Catherine Tong, Sourav Bhattacharya, Nicholas D. Lane, Cecilia Mascolo, Mahesh K. Marina, and Fahim Kawsar. "Multimodal Deep Learning for Activity and Context Recognition." en. In: *ACM J. Interactive, Mobile, Wearable and Ubiq. Tech.* 1.4 (Jan. 2018), pp. 1–27 (cit. on pp. 50, 59).

[122] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M. Buhmann. "The Balanced Accuracy and Its Posterior Distribution." In: *Int'l Conf. on Pattern Recognition* (2010), pp. 3121–3124 (cit. on p. 52).

[123] Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, and Tarek F. Abdelzaher. "DeepSense: A Unified Deep Learning Framework for Time-Series Mobile Sensing Data Processing." In: *WWW*. 2016 (cit. on p. 53).

[124] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. "Automatic differentiation in PyTorch." In: (2017) (cit. on p. 53).

[125] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." In: *arXiv:1512.03385 [cs]* (Dec. 2015). arXiv: 1512.03385 (cit. on p. 58).

[126] M. Dehghani, A. Severyn, Sascha Rothe, and J. Kamps. "Learning to Learn from Weak Supervision by Full Supervision." In: *ArXiv* abs/1711.11383 (2017) (cit. on p. 64).

[127] Ming Zeng, Tong Yu, Xiao Wang, Le T Nguyen, Ole J Mengshoel, and Ian Lane. "Semi-supervised convolutional neural networks for human activity recognition." In: *2017 IEEE International Conference on Big Data (Big Data)*. IEEE. 2017, pp. 522–529 (cit. on p. 64).

[128] Kaixuan Chen, Lina Yao, Dalin Zhang, Xianzhi Wang, Xiaojun Chang, and Feiping Nie. "A semisupervised recurrent convolutional attention model for human activity recognition." In: *IEEE transactions on neural networks and learning systems* 31.5 (2019), pp. 1747–1756 (cit. on p. 64).

[129] Ali Akbari and Roozbeh Jafari. "Transferring activity recognition models for new wearable sensors with deep generative domain adaptation." In: *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*. 2019, pp. 85–96 (cit. on p. 64).

[130] Valentin Radu and Maximilian Henne. "Vision2sensor: Knowledge transfer across sensing modalities for human activity recognition." In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3.3 (2019), pp. 1–21 (cit. on p. 64).

[131] Hirotaka Hachiya, Masashi Sugiyama, and Naonori Ueda. "Importance-weighted least-squares probabilistic classifier for covariate shift adaptation with application to human activity recognition." In: *Neurocomputing* 80 (2012). Special Issue on Machine Learning for Signal Processing 2010, pp. 93–101 (cit. on pp. 77, 80).

[132] Abdulaziz Alajaji, Walter Gerych, Luke Buquicchio, Kavin Chandrasekaran, Hamid Mansoor, E. Agu, and Elke A. Rundensteiner. "Smartphone Health Biomarkers: Positive Unlabeled Learning of In-the-Wild Contexts." In: *IEEE Pervasive Computing* 20 (2021), pp. 50–61 (cit. on pp. 77, 78, 80).

[133] Garrett Wilson, Janardhan Rao Doppa, and D. Cook. "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* (2020) (cit. on pp. 77, 80).

[134] Alexander Hermans, Lucas Beyer, and B. Leibe. "In Defense of the Triplet Loss for Person Re-Identification." In: *ArXiv* abs/1703.07737 (2017) (cit. on pp. 77, 88).

[135] Bulat Khaertdinov, Esam Ghaleb, and Stylianos Asteriadis. "Deep Triplet Networks with Attention for Sensor-based Human Activity Recognition." In: *2021 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE. 2021, pp. 1–10 (cit. on p. 77).

[136] Jonathon Byrd and Zachary Chase Lipton. "What is the Effect of Importance Weighting in Deep Learning?" In: *ICML*. 2019 (cit. on p. 80).

[137] Arthur Gretton, Bharath K. Sriperumbudur, D. Sejdinovic, Heiko Strathmann, Sivaraman Balakrishnan, Massimiliano Pontil, and Kenji Fukumizu. "Optimal kernel choice for large-scale two-sample tests." In: *NIPS*. 2012 (cit. on p. 86).

[138] A. Alajaji, W. Gerych, K. Chandrasekaran, L. Buquicchio, E. Agu, and E. Rundensteiner. "Deep-Context: Parameterized Compatibility-Based Attention CNN for Human Context Recognition." In: *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*. 2020, pp. 53–60 (cit. on p. 94).

[139] L. V. D. Maaten and Geoffrey E. Hinton. "Visualizing Data using t-SNE." In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605 (cit. on p. 101).

[140] Abdur R Shahid, Ahmed Imteaj, Peter Y Wu, Diane A Igoche, and Tauhidul Alam. "Label Flipping Data Poisoning Attack Against Wearable Human Activity Recognition System." In: *arXiv preprint arXiv:2208.08433* (2022) (cit. on p. 105).

[141] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. "Intriguing properties of neural networks." In: *arXiv preprint arXiv:1312.6199* (2013) (cit. on p. 106).

[142] Jianfei Yang, Han Zou, and Lihua Xie. "SecureSense: Defending Adversarial Attack for Secure Device-Free Human Activity Recognition." In: *IEEE Transactions on Mobile Computing* (2022) (cit. on pp. 107, 111, 112, 127).

[143] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. "Practical black-box attacks against machine learning." In: *Proc. ACM Asia conference on computer and communications security*. 2017, pp. 506–519 (cit. on pp. 110, 117).

[144] Siddhant Bhambri, Sumanyu Muku, Avinash Tulasi, and Arun Balaji Buduru. "A survey of black-box adversarial attacks on computer vision models." In: *arXiv preprint arXiv:1912.01667* (2019) (cit. on pp. 110, 112, 113).

[145] Samuel Henrique Silva and Peyman Najafirad. "Opportunities and challenges in deep learning adversarial robustness: A survey." In: *arXiv preprint arXiv:2007.00753* (2020) (cit. on pp. 111, 112).

[146] Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. "Improving the robustness of deep neural networks via stability training." In: *Proc. IEEE CVPR*. 2016, pp. 4480–4488 (cit. on p. 111).

[147] Ramesh Kumar Sah and Hassan Ghasemzadeh. "Adar: Adversarial activity recognition in wearables." In: *Proc. IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE. 2019, pp. 1–8 (cit. on pp. 111, 112, 127).

[148] Alexander Robey, Hamed Hassani, and George Pappas. "Model-Based Robust Deep Learning." In: (May 2020) (cit. on p. 116).

[149] Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. "Online convex optimization in the bandit setting: gradient descent without a gradient." In: *arXiv preprint cs/0408007* (2004) (cit. on p. 117).

[150] Yurii Nesterov and Vladimir Spokoiny. "Random gradient-free minimization of convex functions." In: *Foundations of Computational Mathematics* 17.2 (2017), pp. 527–566 (cit. on p. 117).

[151] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. "Black-box adversarial attacks with limited queries and information." In: *Proc. ICML*. PMLR. 2018, pp. 2137–2146 (cit. on p. 117).

[152] Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, et al. "Adversarial Robustness Toolbox v1. 0.0." In: *arXiv preprint arXiv:1807.01069* (2018) (cit. on p. 127).