# Computational Support for Longitudinal Well-Being

Xuhai "Orson" Xu

Doctor of Philosophy

University of Washington

2023

Reading Committee:

Anind K. Dey, Chair

Jennifer Mankoff

Tim Althoff

Andrew Campbell

Program Authorized to Offer Degree:
Information School

University of Washington

**Abstract**

Computational Support for Longitudinal Well-Being

Xuhai "Orson" Xu

Chair of the Supervisory Committee:
Anind K. Dey
Dean and Professor, Information School

As artificial-intelligent-powered devices have become more embedded in our lives, they offer an unprecedented ability to passively sense our daily behavior at a high resolution. These everyday devices are already equipped with machine learning techniques to monitor our basic health behaviors, such as physical activity and heart rate, and provide suggestions accordingly. However, they are still far from understanding our high-level, longitudinal behaviors, such as mental well-being. Early research about longitudinal behavior modeling and intervention is still facing a set of deployability challenges before being ready for real-world deployment. For behavior modeling, these challenges include interpretability (revealing human-readable insights about behavior), personalization (adapting models to every individual), and generalizability (ensuring models work robustly on new users and contexts). Furthermore, the results and insights of behavior models need to be connected with intervention techniques to influence users' behavior and improve their well-being. With mental well-being as the main application, my research is targeted at these deployability challenges by (1) collecting and releasing the first multi-year passive sensing datasets, (2) developing new behavior modeling techniques that are interpretable, personalized, and generalizable, and (3) designing and deploying a novel intervention technique based on behavior models' insights to improve user well-being. Combining these efforts, I propose the vision of "computational longitudinal well-being", where interactive systems based on everyday devices can precisely and robustly understand, model, and influence long-term human behavior for better health and well-being.

## TABLE OF CONTENTS

## LIST OF FIGURES

## LIST OF TABLES

## LIST OF SYMBOLS AND ABBREVIATIONS

| | |
|---|---|
| **HCI** | human-computer interaction |
| **AI** | artificial intelligence |
| **ML** | machine learning |
| **DL** | deep learning |
| **JIT** | just-in-time |
| **JITAI** | just-in-time adaptive intervention |
| **RKHS** | reproducing kernel Hilbert space |
| **Ubicomp** | ubiquitous computing |
| **IoT** | Internet of Thing |
| **AR** | augmented reality |
| **VR** | virtual reality |
| **XR** | extended reality |
| **HMD** | Head-Mounted Display |
| **ARM** | association rule mining |
| **MDD** | Major Depression Disorder |
| **CV** | Computer Vision |
| **NLP** | Natural Language Processing |

# Chapter 1

# INTRODUCTION

The topic of longitudinal health and well-being has been receiving growing interest from various stakeholders within the industry, academia, and regulatory entities [25, 154]. Especially in the past two decades, the rapid advance of technology has not only greatly facilitated our daily living, but also introduced more physical, mental, and social pressure, leading to more health issues [73]. The COVID-19 pandemic has triggered even more concerns about healthcare worldwide [138]. Take mental health as an example. In the U.S., it is estimated that more than 20% of adults would experience at least one mental disorder in their life time, which is equivalent to over 50 million Americans [12]. Major Depression Disorder (MDD), also known as depression, is one of the most common mental health challenges. It is often accompanied by low self-esteem [27], loss of interest in activities, anxiety [66], low energy, and pain [194]. A recent survey estimated that 8.4% of all U.S. adults and 21.9% of teenagers had at least one MDD episode over the past year [8]. Among 12.0% of young adults, it was reported that the depressive episodes resulted in severe impairment [8].

As artificial-intelligent(AI)-powered devices are increasingly embedded in our lives, they are becoming close companions in daily routines, offering an unprecedented ability to passively sense our behavior at a high resolution, and leading to a unique opportunity to model and impact our behavior to support our goals. Existing everyday devices (*e.g.*, smartphones and wearables) have already utilized AI/ML to monitor basic health behaviors, such as physical activity and heart rate. In the past decade, some early work has demonstrated the capability of moving from monitoring basic behaviors (such as step count or heart rate) to high-level, longitudinal behaviors, such as detecting physical health issues [19, 127, 213], monitoring mental health states [156, 199, 212], measuring job performance [122, 128], tracking education outcomes [200, 226], and tracing social justice [163]. Researchers have used a variety of approaches to address their research questions. Early research starts by doing

statistical analysis, such as correlation analysis (*e.g.*, [47, 156, 199]). Recently, researchers have begun to explore building various machine learning (ML) or deep learning (DL) models (*e.g.*, [7,19]) Meanwhile, other than using passive sensing data to predict behaviors, another thread of research starts to explore leveraging such data to build better just-in-time (JIT) intervention techniques [22, 129, 150].

However, existing ML/DL techniques for longitudinal behavior modeling and intervention are still far from achieving robust real-life deployability (as shown in Fig. 1.1). There are several important challenges:

**Interpretability**. Meaningful and effective support of end-user goals requires interpretability to achieve a transparent and trustful system. However, most longitudinal behavior research has focused on achieving higher detection or prediction accuracy using a black box model, but not on interpretability, which is especially important for mental-health-related behavior modeling problems for both patients (*e.g.*, for reflection) and clinicians (*e.g.*, for informativeness and accountability). This leads to the first research question (**RQ1**): *How to obtain insights and better understand users' behavior besides accurate modeling?*

**Personalization**. Each individual has their own unique behavior pattern, ability, and preference. However, most of the existing behavior models follow a one-size-fits-all approach, without addressing the individual difference during the modeling. Personalization

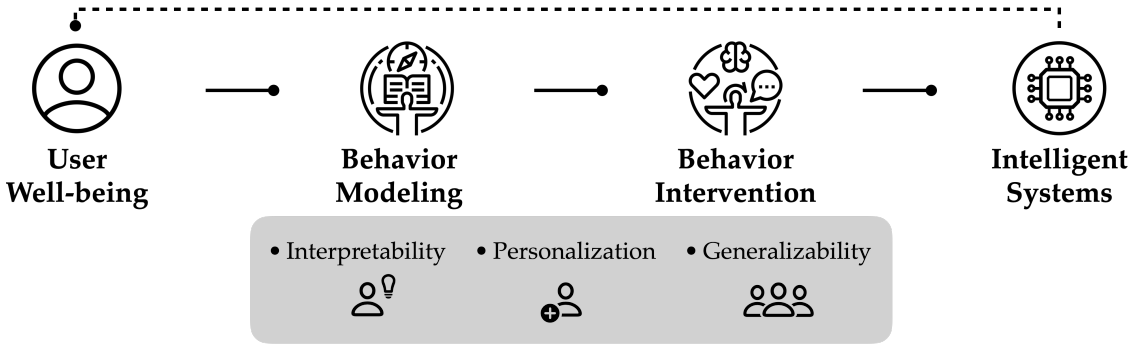



Figure 1.1: The main contributions of the thesis on both behavior modeling and intervention, centered around interpretability, personalization, and generalizability.

is important not only for more accurate models, but also for a more detailed understanding of each individual. This leads to the second research question (**RQ2**): *How to model individuals in light of high variability?*

**Generalizability**. Addressing the challenges of interpretability and personalization in a single dataset is the prerequisite for building deployable and generalizable longitudinal behavior modeling techniques. Once a model is trained on a collected dataset, it needs to be applied to new users under new contexts, which may have different distributions from the training data. However, there is no prior work aiming to address the challenge of cross-dataset generalization, which is an important step toward real-life deployment. This leads to the third question (**RQ3**): *How to generalize a model to other populations/contexts?*

**Intervention**. Addressing these deployability challenges can lead to better behavior models. However, in order to deploy these models, the results of these models need to be embedded into an intervention to better support users' well-being goals in real-life deployment. This further leads to the fourth question (**RQ4**): *How to leverage the insights of behavior models to develop better intervention techniques?*

With mental health as the main application, my research leverages passive sensing data from smartphones and wearables to address these research questions. I have been involved in multiple studies as the technical lead and study coordinator. We have collected and open-sourced the first multi-year passive sensing datasets containing over 700 users across four years. By collaborating with other institutions, our team can access datasets with over one thousand person-years. The datasets also contain users' self-reports on a wide range of well-established surveys, covering various physical, mental, and social well-being conditions [217].

Using the datasets, I have implemented a new behavior modeling algorithm to optimize both modeling accuracy and **interpretability** (RQ1). It utilizes the co-occurrence of multiple sensors' values and leverages Association Rule Mining (ARM) to generate interpretable behavior rules and provide better contextual behavior features [209]. I applied the algorithm to depression prediction and obtained an improvement of around 10% in accuracy and F1 score compared to baseline methods.

To address the challenge of individual uniqueness, I have created a collaborative-filtering-

based framework that effectively uses the study population's large volume of behavioral feature data and limited label data to enable **personalization** (RQ2). The framework leverages individual users' behavioral relevance (both similarities and differences) to generate classifications [210]. To obtain a detailed interpretation of users' behavior, the framework further leverages ARM to enable personalized interpretation. Our evaluation shows that this algorithm further improves depression prediction accuracy by 3-5%.

To address the gap of cross-dataset **generalizability** evaluation (RQ3), I have developed the first open-source benchmark platform GLOBEM – short for Generalization of LOngitudinal BEhavior Modeling – to support researchers in using, developing, and evaluating different longitudinal behavior modeling methods' cross-dataset generalizability [212]. I have also designed a new generalizable modeling algorithm, *Reorder*, which creates a new task of solving a temporal reordering puzzle in addition to the main depression prediction task, forcing the model to learn the continuity of behavior trajectories and achieve better generalization. This algorithm outperforms all previous models by 4-10% on cross-dataset generalization tasks.

To better bring the insights from behavior models to users, I further designed and deployed a theory-driven **intervention** technique called *TypeOut* (RQ4). It combines self-affirmation theory and typing-based JIT design to improve users' well-being [218]. All of my previous behavior models reveal a strong association between depression and smartphone overuse. Therefore, as an initial effort, I applied the intervention technique to reduce smartphone overuse. Our 10-week deployment study results with 54 users indicate that *TypeOut* could effectively reduce smartphone usage and frequency by over 25%.

The contributions of my thesis can be summarized from four perspectives:

1. **Algorithmic Contribution**: I have developed interpretable, personalized, and generalizable machine learning algorithms to address these key deployability challenges (RQ1-3). With mental health as the main application, the algorithms have become the new state-of-the-art several times.
2. **Design Contribution**: I have designed the first intervention technique that combines self-affirmation theory and JIT design to influence user behavior and improve their lon-

gitudinal mental well-being (RQ4). Based on the behavior models, I applied it to smartphone overuse reduction, and our in-the-wild deployment study revealed its effectiveness.

3. **Open-source Contribution**: We have collected and open-sourced the first multi-year passive sensing datasets that cover over 700 users across four years. I further developed and released the first open-source benchmark platform, which enables other researchers to develop, evaluate, and compare different behavior modeling algorithms in a systematic and reproducible cross-dataset setting.
4. **Empirical Contribution**: My algorithms can generate human-readable behavior rules at both the population level and the individual level. Many of our empirical findings of depression-related behavior are supported by psychology and psychiatry literature, with some findings suggesting new research questions for behavioral science researchers.

This thesis mainly focuses on mental health as the application domain. My other research also covers more aspects of human well-being (the gray part in Fig. 1.1 left), such as physical health [213] and education outcomes [227], as well as other ubiquitous devices (the gray part in Fig. 1.1 right), such as Internet of Thing (IoT) [214] and Virtual/Augment Reality (VR/AR) head-mounted displays (HMDs) [172, 215]. Combining these efforts, I have made progress towards my vision of **"computational longitudinal well-being"** by creating the next-generation intelligent system based on everyday devices to understand, model, and influence user behaviors for better health and well-being.

The following Ch. 2 will describe the background and related work. Ch. 3 will introduce our dataset that my research is based on. My research of interpretability, personalization, and generalizability will be introduced from Ch. 4 to Ch. 6. Then, Ch 7 will cover the novel JIT intervention design and deployment study. Finally, Ch. 8 concludes the thesis.

# Chapter 2
# BACKGROUND

This chapter covers the background from several perspectives: (1) health and passive sensing, (2) interpretable ML, (3) personalized ML, (4) generalizable ML, and (5) JIT intervention techniques based on passive sensing.

## *2.1 Health & Passive Sensing*

There is a growing realization that daily routine behaviors, such as movement trajectories, sleep patterns, social activities, and physical activities, can be continuously, passively, and longitudinally tracked by sensors embedded in mobile phones and wearable devices. Researchers have demonstrated the feasibility of using mobile sensing to capture and model longitudinal daily behavior in many settings [74,103,199,217], especially in health and well-being domains [156, 195, 201, 210, 213]. To give a few examples, Wang *et al.* [199] identified the correlation relationship between college students' mental health and data collected from smartphones. Abdullah *et al.* [7] used phone usage patterns (*e.g.*, call logs and application usage history) to predict sleep time, duration, and deprivation. Bae *et al.* [19] used multiple streams such as GPS location, phone usage patterns, and Bluetooth signals from mobile phones to detect drinking episodes of young adults.

Depression is one of the major mental health concerns worldwide. Detecting depression at an early stage can help mitigate or prevent its negative consequences. Successes in the last decade of using mobile sensing for depression-related research have made this topic increasingly popular. Earlier work focused on understanding the statistical relationship between depressive symptoms and features extracted from mobile sensing data [24, 87, 156]. For instance, Katikalapudi *et al.* [87] found a significant positive correlation between Internet usage and depression scores. Saeb *et al.* [156] identified a significant correlation between depression scores and location features (location variance, location entropy, and circadian

movement), and also phone usage features (usage duration and frequency). Ben *et al.* [24] observed a significant correlation between changes in depression scores and sleep duration, speech duration, and mobility. Recently, researchers have leveraged the results of correlation analysis to build ML models for depression diagnosis and detection. For example, Farhan *et al.* [54] used location features to detect biweekly depression, and their best model achieved an F1 score of 0.82 on a dataset with 79 college students over eight months. Wahle *et al.* [195] trained models on multiple data streams, including location, physical activity, phone usage, and WiFi scans, and achieved an accuracy of 61.5% for depression prediction on a dataset with 36 participants over ten weeks. Wang *et al.* [201] hand-crafted several cross-sensor features from mobile and wearable data. Their best model achieved 81.5% recall and 69.1% precision on a dataset collected from 68 college students over two nine-week terms. Chikersal *et al.* [35] developed a feature extraction, selection, and ML pipeline to detect both depression and change of depression. Their best model achieved an accuracy of 85.7% and 85.4% for the two tasks, respectively.

However, existing algorithms did not address the challenges of interpretability, personalization, and generalizability. My research aims to tackle these challenges to achieve robust and deployable longitudinal behavior modeling techniques.

### *2.2 Interpretable ML*

With the increasing prevalence of advanced black-box models making more critical predictions and decisions, the interpretability and transparency of AI systems have attracted more and more attention from various stakeholders [67, 70, 146, 189]. For instance, the European Union General Data Protection Regulation (GDPR) commission established the legal right to obtain explanations [189], and the Defense Advanced Research Projects Agency (DARPA) also formulated the XAI program to enable effective understanding and management of AI systems [70]. Addressing the broad vision of making AI more understandable for humans involves multidisciplinary research efforts. HCI researchers have focused on user trust [83, 148] and understanding [110, 111] of machine generated explanations. Psychology researchers have approached XAI from a more fundamental perspective and studied how people generate, communicate, and understand explanations [180, 222]. ML researchers, on

the other hand, have developed advanced algorithms for transparent models (*e.g.*, decision trees, Bayesian models [32, 108]) or used post-hoc explanation techniques (*e.g.*, feature importance, visual explanation [118, 164, 169]) to generate explanations. However, there is very limited previous work in the longitudinal behavior modeling domain to optimize both model accuracy and interpretability at the same time. Among various post-hoc techniques, rule-based understanding is suitable for longitudinal behavior modeling, because human behavior can be naturally summarized as a set of rules: if users do X, then they may do Y. Association Rule Mining (ARM) is a data mining method that can find frequent patterns, correlations, associations, or causal structures from datasets. It has been used in a range of domains, from helping to discover sales correlations in transaction datasets [141] to identifying disease correlations in medical datasets [13]. In Ch. 4, I will introduce my research on leveraging ARM for an interpretable rule-based behavior modeling algorithm.

### *2.3 Personalized ML*

Besides the problem of interpretability, another important challenge is personalization. There are two major types of personalized machine learning algorithms: sample-specific methods and similarity-based methods [193]. I review both classical and modern examples of the two methods. Sample-specific methods are model-based methods that leverage training and testing samples' features to enable personalization when inducing a model. Classic examples include lazy decision trees (LDT) [62] and lazy Bayesian rules (LBR) [232], where part of the training occurs when the testing sample is collected. Building separate models on individual data [29] is another example, where the individual identifier is used to select the model (*i.e.*, the model trained on this individual's data) for prediction. There have been more recent advances in ML community [160, 219]. For instance, Schulam *et al.* [160] proposed a hierarchical model with multi-resolution latent variables where they first train population-level parameters offline and then train individual-specific parameters through an online process. Lengerich *et al.* [107] proposed a personalized logistic regression model (PLR) to include different parameters for every sample, with the parameter matrix having a low-rank property as the constraint on the parameters' degree of freedom. However, these sample-specific methods usually require a large number of ground-truth labels for each in-

dividual. In the area of passive sensing, the collection of labels is expensive, and many data sets have only one label for each individual. In contrast, similarity-based methods use a distance measure and combine training samples in some fashion for the prediction. Traditional methods such as K-Nearest Neighbour (KNN) [88] and locally weighted regression (LWR) [17] are examples of this approach. There have been more recent advances in the behavior modeling area with this type of method. Lane *et al.* [104] and Abdullah [6] leveraged personal informatics, mobility behavior, and raw sensor data to construct three similarity matrices. They used the three matrices to initialize parameters when training three boost models for activity recognition. Then, they leveraged majority voting of the three models to determine the final predictions. Lopez *et al.* [115] used spectral clustering to group individuals into several groups based on their behavioral profiles and then applied multi-task learning for subjective pain estimation. However, these methods still require a number of labels to enable effective model training. Moreover, they do not explore the aspect of interpretability. In Ch. 5, I will introduce a new similarity-based framework that addresses both personalized behavior detection and personalized interpretation by the combination of collaborative filtering and ARM.

### *2.4 Generalizable ML*

Building a model that can generalize across multiple domains or datasets has been a challenging problem in the ML community. Such a task corresponds to an ideal real-life deployment setup: directly applying a trained model to new users, without the need to access any data from the new users. In the past few years, ML researchers have proposed a wide range of algorithms for domain generalization. Most of them belong to one of the following three categories [197, 233]: 1) Data manipulation, which enhances the data by augmentation or generation techniques to assist the model training (*e.g.*, [44, 228]); 2) Representation learning, which aims to learn desirable feature representations that can generalize across domains (*e.g.*, [15, 55, 63]); 3) Learning strategy, which focuses on exploiting the training procedure to promote a model's generalizability (*e.g.*, [109, 188, 220]). Almost any applied ML area would encounter the cross-dataset domain generalization challenge for real-life deployment, such as object recognition [20, 30, 86, 184], NLP [134, 230], affective computing [37, 84], secu-

rity [81, 229], and intelligent interaction [211, 214, 216]. Researchers have developed cross-dataset benchmark platforms such as DomainBed [69], DeepDG [197], and WILDS [98] to facilitate related studies in the ML community. Recently, researchers have started to merge multiple passive sensing datasets for behavior model training [10]. Mishra *et al.* [130] evaluated three models' generalizability in the physiological stress detection domain using four datasets collected from different wearable devices in a passive manner. However, their models aimed at short-term detection (over a few seconds), so the ground truth labels were frequent and rich. Moreover, the datasets they employed were all collected in a lab setting. There is no prior work on cross-dataset generalizability evaluation in the longitudinal behavior modeling field with data collected in the wild. In Ch. 6, I will introduce how I address this issue by developing a systematic evaluation platform, together with a new algorithm achieving better generalizability.

### *2.5 Just-in-Time Intervention*

Building interpretable, personalized, and generalizable behavior models is important. But it is the first step toward next-generation intelligent systems. Based on these models, a system should also bring intelligent intervention to users to change their behavior and improve their well-being. In recent years, the advances of passive sensing have led to the rapid growth of just-in-time intervention (JITI) and just-in-time adaptive intervention (JITAI) in the HCI and ubiquitous computing community, using AI/ML techniques to facilitate the design of intervention techniques [9, 22, 129, 142, 144, 150]. In various kinds of JITI (*e.g.*, persuasion, education, incentivisation [125]), the Fogg Behavior Model points out that a successful triggering process is necessary for them to be effective [58]. However, in most existing technology, users are often the passive receivers of intervention (*e.g.*, [89,94,99,120,150]). An intervention system usually delivers just-in-time persuasive, educative, or incentive content to users that are based on some theories (*e.g.*, dual process [60], goal setting [114]). But there is a lack of mechanism to encourage users to effectively engage with the intervention, which can cause the failure of the triggering process and result in the decline of compliance and adoption rate, and diminish or even wipe out the effectiveness of the intervention [139, 158, 221]. In Ch. 7, I will introduce a new theory-driven intervention design to achieve the

balance between engagement and effectiveness. Based on the interpretable insights from depression prediction models, I applied the design to smartphone overuse reduction as a starting point, since it is closely related to depression.

# Chapter 3

# DATASET

After years of data collection studies and collaboration, my teams have access to datasets with over one thousand student-years of behavior and self-report data from multiple institutions, including University of Washington, Carnegie Mellon University, and Dartmouth College. In this chapter, I will introduce one of the major datasets from University of Washington that I have been deeply involved in. The datasets from other institutions have a similar structure.

## *3.1 Study Procedure*

Our team recruited undergraduates via emails, flyers, and social posts from 2018 to 2021 [163]. After the first year, previous-year students were invited to join again. The study was conducted during Spring quarters (10 weeks) each year, so the impact of seasonal effects was controlled. Participants received up to $245 in compensation based on their compliance level each year. The study went through an IRB review and approval. Fig. 3.1 presents the overview of the data collection process and Tab. 3.1 summarizes the stufy information

The four datasets (DS1 to DS4) have 155, 218, 137, and 195 participants (705 person-years overall, and 497 unique people). We intentionally oversampled minoritized groups to make our datasets more representative. Our datasets have a high representation of females (58.9%), immigrants (24.2%), first-generations (38.2%), and people with disability (9.1%), and have a wide coverage of races, with Asian (53.9%) and White (31.9%) being dominant (Hispanic/Latino 7.4%, Black/African American 3.3%).

## *3.2 Survey Data*

We collected survey data at multiple stages of the study. We delivered extensive surveys before the start and at the end of the study (pre/post surveys) and weekly Ecological

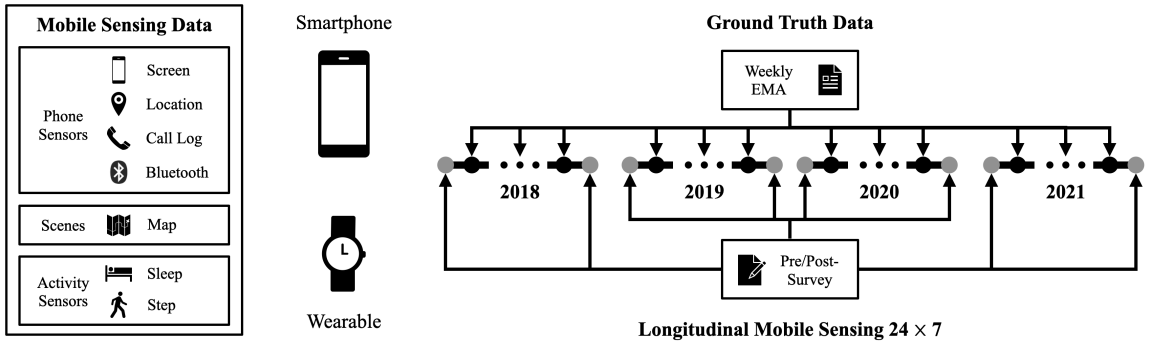



Figure 3.1: Overview of Longitudinal Passive Sensing Data Collection Studies. Each year's study lasted a 10-week academic quarter.

Table 3.1: Basic Study Information and Participant Demographics of Four Datasets.

| | **Year1 - DS1** | **Year2 - DS2** | **Year3 - DS3** | **Year4 - DS4** |
|---|---|---|---|---|
| **Participants** | • Total: 155<br>• Gender: F 107, M 48<br>• Generation: Im 34, 1stG 53, 2ndG 11, 3rdG 57<br>• Disability: 5<br>• Race: A 82, B 5, H 9, N 4, PI 3, W 50, A&PI 2 | - Total: 218<br>• Gender: F 111, M 107<br>• Generation: Im 54, 1stG 75, 2ndG 18, 3rdG 63, NA 8<br>• Disability: 21<br>• Race: A 102, B 6, H 10, N 2, PI 1, W 70, A&B 1, A&W 16, H&W 2, B&W 2, A&H&W 1, B&H&W 1, H&N&W 1, NA 3<br>• Overlap: 23 in Year1 | - Total: 137<br>• Gender: F 75, M 61, NB 1<br>• Generation: Im 35, 1stG 52, 2ndG 8, 3rdG 40, NA 2<br>• Disability: 22<br>• Race: A 74, B 3, H 8, PI 3, W 40, A&W 6, B&H&W 1, NA 2<br>• Overlap: 19 in Year1&2, 4/47 in Year1/2 | - Total: 195<br>• Gender: F 122, M 67, NB 6<br>• Generation: Im 48, 1stG 89, 2ndG 13, 3rdG 42, NA 3<br>• Disability: 16<br>• Race: A 104, B 4, H 18, N 1, PI 2, W 48, A&W 13, H&W 2, NA 3<br>• Overlap: 19 in Year1&2&3, 4 in Year1&2, 4 in Year1&3, 47 in Year2&3, 2/19/20 in Year1/2/3 |
| **Survey** | • Pre/post: UCLA, SocialFit, 2-Way SSS, PSS, ERQ, BRS, CHIPS, STAI, CES-D, BDI2, MAAS, BFI10, Brief-COPE, GQ, FSPWB, EDS, CEDH, B-YAACQ<br>• Weekly EMA: PHQ-4, PSS-4, PANAS | | | |
| **Depression** | • Weekly: Depression & Affect (45.5%)<br>• End-term: BDI-II (35.4%) | • Weekly: PHQ-4 (52.1%)<br>• End-term: BDI-II (42.9%) | • Weekly: PHQ-4 (46.9%)<br>• End-term: BDI-II (40.7%) | • Weekly: PHQ-4 (45.0%)<br>• End-term: BDI-II (40.2%) |
| **Sensor** | • Smartphone: Location, Phone Usage, Call, Bluetooth<br>• Wearable: Physical Activity, Sleep | | | |

*Participants with less than 2 weekly EMAs or less than a 25% of their sensor data (*i.e.*, missing rate > 75%) were excluded from the dataset. In the depression row, the percent indicates the portion of participants having at least mild depressive symptoms based on the corresponding questionnaires. Gender acronym - F: Female, M: Male, NB: Non-binary. Generation acronym - Im: Immigrant (born in another country), 1stG: First generation (parents immigrated to the US), 2ndG: Second generation (grandparents immigrated to the US), 3rdG: Third generation (great grandparents or further back immigrated to the US), NA: Prefer not to respond. Racial acronym - A: Asian, B: Black or African American, H: Hispanic or Latino, N: American Indian/Alaska Native, PI: Pacific Islander, W: White, NA: Did not report. & is used when participants reported more than one races.

Momentary Assessment (EMA) surveys during the study to collect in-the-moment self-report data. All surveys consisted of well-established and validated questionnaires to ensure

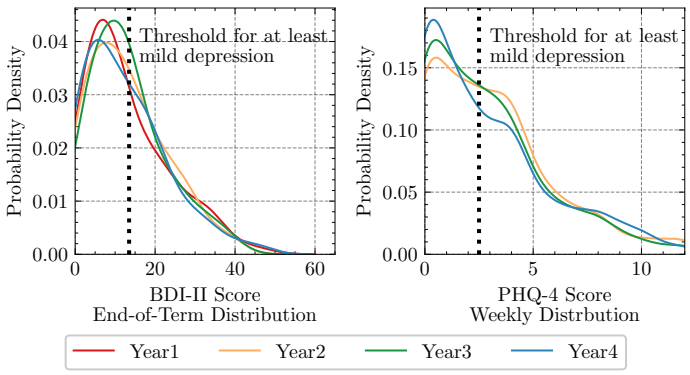



Figure 3.2: The Distribution of Label Scores for End-of-Term (BDI-II) and Weekly Depression Scales (PHQ-4).

data quality.

Our pre/post surveys included questionnaires to cover various aspects of life, including:

- Personality. BFI-10, The Big-Five Inventory-10 [151].
- Physical health. CHIPS, Cohen-Hoberman Inventory of Physical Symptoms [39],
- Mental well-being. BDI-II, Beck Depression Inventory-II [23]; ERQ, Emotion Regulation Questionnaire [68]; UCLA, Short-form UCLA Loneliness Scale [155]; BRS, Brief Reslilience Scale [170]; RRQ, Rumination-Reflection Questionnaire [185]; Brief-COPE, Brief Coping Orientation to Problems Experienced Inventory [33]; FSPWB, Flourishing Scale Psychological Well-Being Scale [46].
- Social well-being. Sense of Social and Academic Fit Scale [196]; GQ, Gratitude Questionnaire [123]; EDS, Everyday Discrimination Scale [4, 204]; MED, Major Experiences of Discrimination [4]; CEDH, Chronic Work Discrimination and Harassment [26, 204].

Our EMA surveys focused on capturing participants' recent sense of their mental health, including PHQ-4, Patient Health Questionnaire 4 [2, 101]; PSS-4, Perceived Stress Scale 4 [1, 40]; and PANAS, Positive and Negative Affect Schedule [3, 202].

Since I have focused on depression prediction as the example task, BDI-II (post) and PHQ-4 (EMA) were employed as the ground truth. Both are screening tools for further

inquiry of clinical depression diagnosis. From Ch. 4 to Ch. 6, I will mainly focus on a binary classification problem to distinguish whether participants' scores indicated at least mild depressive symptoms through the scales (*i.e.*, PHQ-4 > 2, BDI-II > 13)[1]. The average number of depression labels is 11.6±2.6 per person. Fig. 3.2 summarizes the distribution of survey scores across four datasets. The percentage of reports with at least mild depression is 39.8±2.7% for BDI-II and 47.4±2.8% for PHQ-4.

## *3.3 Sensor Data*

We developed a mobile app using the AWARE Framework [57] that continuously collects location, phone usage (screen status), Bluetooth scans, and call logs. The app is compatible with both the iOS and Android platforms. Participants installed the app on smartphones and left it running in the background. In addition, we provided Fitbits to each participant to collect their physical activities and sleep behaviors. The mobile app and wearable passively collected sensor data 24×7 during the study. The average number of days per person per year is 77.5±8.9 among the four datasets.

### *3.3.1 Feature Extraction*

We utilized RAPIDS [192], an open-source platform that provides a Reproducible Analysis Pipeline for Data Streams. It supported feature extraction from data collected via multiple mobile and wearable devices with various time windows.

**Data Type: Location.** We incorporated all features in RAPIDS-Location, which included location variance, location entropy, travel distance, *etc.* In addition, we also added more features (duration of staying) for specific points of interest, including places for living, study, exercise, and relaxation.

**Data Type: Phone Usage.** We included all features in RAPIDS-Screen that covered the statistics of unlocking episodes (count, sum, mean, std, max, min). We further contextualized these features at different locations (home and study places) to capture fine-grained

[1]DS1 did not have PHQ-4, we trained a simple rule-based model using two sub-questions of PANAS to estimate PHQ-4 score.

phone usage behaviors.

**Data Type: Bluetooth.** We used all features from RAPIDS-Bluetooth, including the number of scans of participants' own devices and others' devices, as well as the unique count of these devices.

**Data Type: Call.** We employed features from RAPIDS-Call that covered the statistics of incoming/outgoing calls' duration (count, sum, mean, std, max, min, entropy), and the count of missed calls.

**Data Type: Physical Activity.** We utilized physical activity features from RAPIDS-Fitbit-Steps. They included both high-level features (number of steps, duration of being active) and low-level features about the statistics of active or sedentary episodes (mean, std, max, min).

**Data Type: Sleep.** We leveraged sleep-related features from RAPIDS-Fitbit-Sleep, including high-level summary features (total duration of being asleep or in bed), and low-level features about the statistics (count, mean, max, min) of episodes of being asleep, restless, and awake during the sleep.

**Feature Time Range.** Research has found that people tend to have distinctive behavior patterns during different times of the day [36], or accumulate their behavior routines through a period of days [29]. Thus we incorporated different time ranges during feature extraction, including four epochs of a day (split at 6 am, 12 pm, 6 pm, and 12 am), the whole day, and the past one/two weeks. It is worth noting that all features are calculated every day for each user, forming a long daily feature vector.

#### *3.3.2 Feature Post-processing*

After feature extraction, we further conducted a few post-processing steps to provide a comprehensive feature set: 1) Feature normalization: We added all features' normalized version based on each individual's distribution: subtracting the median and scaling with the 5-95 quantile range on each individual; 2) Feature discretization: A few modeling algorithms may benefit from using categorical levels instead of raw feature values (*e.g.*, [209]). Thus, we also added all features' 3-level discretized versions (split by the one/two/three third

percentile within each individual's data).

Missing data is inevitable due to various reasons, such as a low battery, data transfer loss, and sensor permission withdrawal. For example, the average missing rate for location features is 14.5±4.0%. Other than special notice, we omitted missing values during analysis and used a median-based imputation when necessary.

For datasets collected from other institutions, we also followed a similar feature extraction and processing procedure.

Using these datasets, I then introduce the algorithms developed to address behavior modeling challenges.

# Chapter 4
# INTERPRETABLE BEHAVIOR MODELING

With the extensive multi-year dataset, I start with answering the first research question (**RQ1**): *How to obtain insights and better understand users' behavior besides accurate modeling?*. Most of the previous depression prediction work either focused on a single sensor channel such as location (*e.g.*, [29,54,156]), or combined multiple sensor channels but treated each sensor channel as a separate feature, which misses the opportunity to capture co-occurrence relationships between sensors (*e.g.*, [177,195]). Such co-occurrence relationships might boost the performance of machine learning model, and more importantly, they can provide better interpretable insights for understanding people's behavior from wearable and mobile sensors.

In this chapter, I developed a new approach to capturing these co-occurrence relationships across sensor channels. Such co-occurrence relationships can provide us with a deeper interpretation of users' behaviors and how they are related to depression. I then proposed a new feature extraction and model training pipeline to produce more powerful models. I will first introduce the methods in Ch. 4.1. Evaluation results are summarized in Ch. 4.2. Finally, the key findings are highlighted in Ch. 4.3.

## *4.1 Methods*

The key idea of the approach is to: **identify behavior rules that capture important differences between classes of users.** Fig. 4.1 visualizes the overall framework for the method.

### *4.1.1 Brief Introduction of ARM*

Before I introduce my new algorithm, here is a brief recap of ARM: ARM is a data mining method that can find frequent patterns, correlations, associations, or potential causal struc-

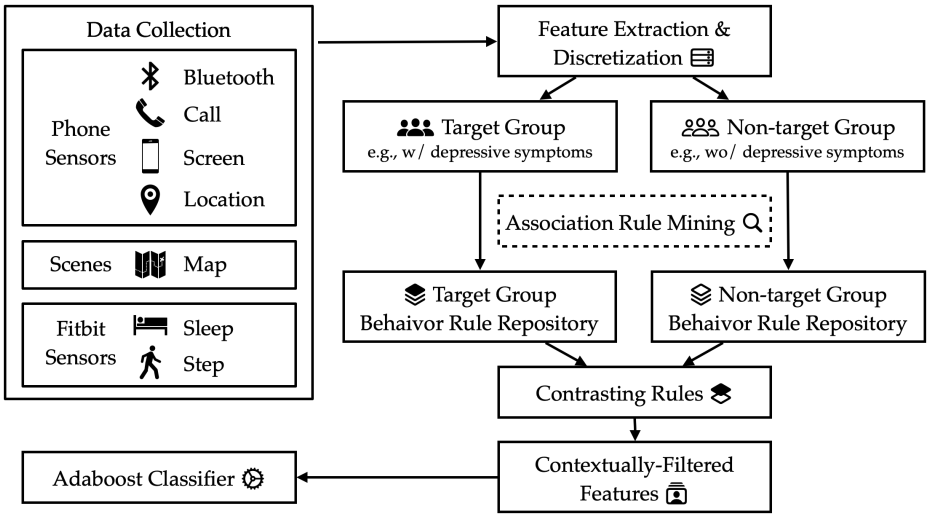



Figure 4.1: The Framework of Modeling with Interpretability. After feature extraction, I first separate two groups of users with/without depressive symptoms. Then I mine out frequent behavior rules from each group and identify effective rules that capture behavior differences between the two groups. These effective rules are then used to extract contextually-filtered features, which can lead to powerful depression prediction models. Note that the model is trained/tested on a different user group to avoid overfitting.

tures from datasets. ARM outputs frequent co-occurrence patterns expressed as association rules [11]. Each association rule is in the form of $[X \rightarrow Y]$, where $X$ and $Y$ are co-occurring sets of context features, and the association rule indicates that the context features in $Y$ are likely to occur whenever the features in $X$ are observed. For rule $[X \rightarrow Y]$, two parameters are defined [11]:

- **Support**: Support represents the fraction of times the context set $\{X, Y\}$ occurs in the dataset, *i.e.*, the joint probability $sup = P(X, Y)$.
- **Confidence**: Confidence represents the proportion of times $Y$ co-occurs whenever $X$ occurs, *i.e.*, the conditional probability $conf = P(Y|X)$.

A support and confidence threshold, namely $sup_{min}$ and $conf_{min}$, is used to set the minimum support and confidence allowable for each discovered rule.

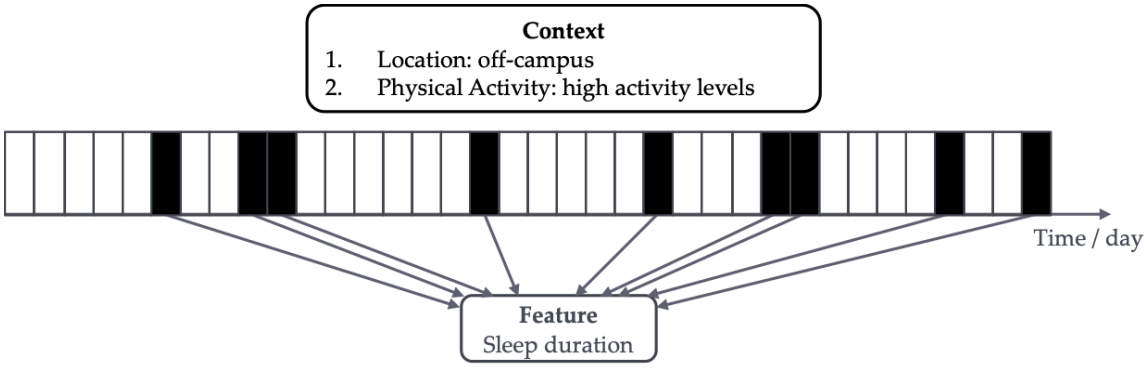



Figure 4.2: The Concept of Contextually Filtered Features. Once a rule is obtained [Contexts → Feature], the contexts are used to filter a subset of time windows to calculate a cleaner aggregation of the target features.

In our case, ARM is particularly suitable since human behavior can be naturally summarized as a set of rules. Using our datasets, I have extracted unimodal features as described in Ch. 3. I use ARM to generate rules using behavior features on a per-class basis (*e.g.*, once for the depression user group, and once for the non-depression user group). I then use a novel metric to select the top rules that can capture the behavior differences between classes. Based on the top rules, I design a new automated approach to obtain contextually-filtered features (see Fig. 4.2).

The specific procedure of the new algorithm works as follows.

### *4.1.2 Step 1: Rule Mining in Two Classes Separately*

I first split the dataset into two groups according to the classification label, namely *grp1* and *grp2*. (*e.g.*, depression group *vs.* non-depression group). I perform ARM on them separately to generate a large rule set in each group. ARM naturally fits the problem since I obtain multiple features from various sensors at the same time. In a rule $[X \rightarrow Y]$, both $X$ and $Y$ would contain behavioral features. An example rule could be: [$X$: {*Staying at home, Low activity level*} → $Y$: {*Being asleep*}] during the night. Next, I devise a novel approach to select the best rules from the two rule sets.

#### *4.1.3 Step 2: Rule Selection Using a Novel Metric*

My method emphasizes the characteristics of and differences between the two classification groups. The groups can be different by (1) sharing similar contexts but with different behavior in those contexts (see Sec. 4.1.3.1), or (2) having contexts that are common in one class and uncommon in the other class (see Sec. 4.1.3.2). To capture the difference, I use two complementary perspectives: one looks at rules that are the same between two groups but with different *sup* and *conf* values, while the other looks at rules that are unique to only one group.

##### *4.1.3.1 Common Rules in Two Groups*

I present a set of metrics for characterizing a rule's usefulness for identifying an individual's group membership.

**Contextual Specificity** Rules that are too general are unlikely to discriminate between the two groups. Thus, I filter rules that are not very specific. For example, a rule that captures a behavior pattern such as changing locations every weekday morning (from home to work) is less specific than a rule that contains more features, *e.g.*, changing location with a low level of physical activity but a high number of co-locations with other people (the number of Bluetooth encounters). I formalize this in terms of the number of features in $X$ for a rule $[X \rightarrow Y]$.

$$CtxSpec = |X|$$

**Confidence Difference** The confidence of a rule, *i.e.*, the conditional probability, indicates how probable $Y$ is to occur given the context feature set $X$. For the same $X$, the difference in confidence directly reflects the different probabilities of $Y$ between the two groups. For example, when in working spaces such as offices or libraries (same $X$), people with depressive symptoms may have more difficulties with concentration [16], thus spending more time interacting with their phones [201]; this could appear in the analysis as higher confidence for the behavior rule $R_i$ $[X: \{Stay\ at\ working\ spaces\} \rightarrow Y: \{Phone\ interaction\ time\}]$ for people with depressive symptoms.

The bigger the difference in confidence, the greater the discrepancy in the expression of the rule between the two groups. I formalize this as

$$ConfDiff = |\Delta conf|$$

**Condition Discrepancy** The probability of the context set $X$, *i.e.*, $P(X)$, closely interrelates with the confidence of the rule. The interesting rules are those that have different context probabilities across groups. Continuing the previous example, people with depressive symptoms may spend less time in working or social spaces [16], leading to a different $P(X)$ for those with and without depressive symptoms for rule $R_i$. Note that $P(X) = \frac{P(X,Y)}{P(Y|X)} = \frac{sup}{conf}$. Thus I formalize this as

$$CondDisc = |\Delta P(X)| = |\Delta \frac{P(X,Y)}{P(Y|X)}| = |\Delta \frac{sup}{conf}|$$

**Direction Difference** Only the rules with *CondDisc* and *ConfDiff* in the same direction show a clear distinction. In other words, they are the rules that have both higher $P(X)$ and higher $P(Y|X)$ in one group than the other. I formalize this as

$$DirDiff = \begin{cases} 1 & \text{if sign}(\Delta conf \cdot \Delta \frac{sup}{conf}) \text{ is positive} \\ 0 & \text{otherwise} \end{cases}$$

Based on these characteristics, I combine the four characteristics into a metric $M$ using Equation 4.1. The intuition comes from a weighted addition of the logarithm value of the three characteristics. The logarithmic function is monotonically increasing; thus it will not change the relative order when ranking the rules based on the metric.

$$M = DirDiff \cdot CtxSpec^{w_1} \cdot ConfDiff^{w_2} \cdot CondDisc^{w_3} \tag{4.1}$$

*DirDiff* simply causes features to be dropped if a rule has reverse directions on the *ConfDiff* and *CondDisc* between two groups. The three weight values are used to adjust the relative importance of the remaining three characteristics. I rely on $M$ to rank the rules that are common in both groups and select the top-$n$ rules. I remove redundant rules by the definition: $Rule_1$ *covers* $Rule_2 \iff X_2 \subseteq X_1$ and $Y_2 \subseteq Y_1$. Alg. 1 (top) presents the procedure.

**Data:** *grp1*, *grp2*, mining thresholds $sup_{min}$ and $conf_{min}$

$R_1$ = ARM(*grp1*, $sup_{min}$, $conf_{min}$);

$R_2$ = ARM(*grp2*, $sup_{min}$, $conf_{min}$);

---

**// Select Common Rules, see Section 4.1.3.1**

$R_{common}$ = $(R_1 \cap R_2)$;

**for** *each rule r in* $R_{common}$ **do**

  $CtxSpec = |X|$;

  $ConfDiff = |\Delta conf|$;

  $CondDisc = |\Delta \frac{sup}{conf}|$ ;

  $DirDiff = \text{sign}(\Delta conf \cdot \Delta \frac{sup}{conf}) > 0$;

  $M_{common}[r] = DirDiff \cdot CtxSpec^{w_1} \cdot ConfDiff^{w_2} \cdot CondDisc^{w_3}$

**end**

$sort(M_{common})$ ; // Sort by score

$T_{common} = M_{common}[0]...M_{common}[n-1]$ ; // Select top $n$ rules

$T_{common} = T_{common} \setminus \{r \in T_{common} \mid \exists\, r^* \in T_{common}, r^* \neq r, X_r \subseteq X_{r^*} Y_r \subseteq Y_{r^*}\}$ ; // Remove redundancy

---

**// Select Unique Rules, see Section 4.1.3.2**

$R_{unique}$ = $(R_1 \cup R_2) \setminus (R_1 \cap R_2)$;

$sup_{dis}$ = $99^{th}$ percentile($|\Delta sup|$ in $R_{common}$);

$conf_{dis}$ = $99^{th}$ percentile($|\Delta conf|$ in $R_{common}$);

$R_{unique} = R_{unique} \setminus \{r \in R_{unique} \mid r_{sup} < sup_{min} + sup_{dis}\}$ ; // Remove rules close to threshold

$R_{unique} = R_{unique} \setminus \{r \in R_{unique} \mid r_{conf} < conf_{min} + conf_{dis}\}$;

**for** *each rule r in* $R_{unique}$ **do**

  $CtxSpec = |X|$;

  $ConfDiff = conf$ ;

  $CondDisc = \frac{sup}{conf}$ ;

  $M_{unique}[r] = CtxSpec^{w_1} \cdot ConfDiff^{w_2} \cdot CondDisc^{w_3}$

**end**

$sort(M_{unique})$;

$T_{unique} = M_{unique}[0]...M_{unique}[n-1]$;

$T_{unique} = T_{unique} \setminus \{r \in T_{unique} \mid \exists\, r^* \in T_{unique}, r^* \neq r, X_r \subseteq X_{r^*} Y_r \subseteq Y_{r^*}\}$;

---

**// Merge The Rules**

$T_{top} = T_{common} \cup T_{unique}$;

---

**Algorithm 1:** ARM for Best Rules Selection.

#### *4.1.3.2 Unique Rules in One Group*

In addition to common rules that are mined from both groups, the two groups can also have unique rules discovered in one group but not the other. These rules reflect the differences in behavior patterns and contexts between the two groups. Selecting the best unique rules in each group can also help identify the distinctions between the two groups. *ConfDiff*, *CondDesc* and *DirDiff* are undefined when a rule is present in only one group. I solve this by setting $\frac{sup}{conf}$ and *conf* to zero when a rule is not present, thus simplifying the calculation of *ConfDiff* and *CondDesc* as shown in Equation 4.2. *DirDiff* is not used since it always equals 1. I then employ the same metric $M$ as Equation 4.1 and select the top-$n$ rules.

$$\begin{aligned} ConfDiff &= conf \\ CondDisc &= \frac{sup}{conf} \end{aligned} \tag{4.2}$$

However, note that the approach above could treat a rule as unique to one group if the threshold values filtered the rule out in the other group. For instance, take a rule $r$ that occurs in both groups with $sup_1 = 0.101$ and $sup_2 = 0.099$, which are very close. The rule would not appear in *grp2* when the threshold $sup_{min} = 0.100$ is applied, while it would appear in *grp1*.

Thus I need to filter out these rules whose *sup* and *conf* are close to the threshold. I set the minimal distance to be 99th percentiles of the $|\Delta sup|$ and $|\Delta conf|$, as shown in lines 15-18 of Alg. 1. Similar to the common rules, I rank the resulting unique rules using metric $M$, and remove any redundant rules.

Overall, I obtain $T_{common}$ from Sec. 4.1.3.1 and $T_{unique}$ from Sec. 4.1.3.2. The final set of top rules is calculated on line 28 as $T_{top} = T_{common} \cup T_{unique}$. Next, I describe a new approach to extract contextually filtered features from the top rule set.

### *4.1.4 Step 3: Contextually Filtered Feature Creation*

Once a top rule set $T_{top}$ has been selected using the algorithms, they can be used to generate contextually filtered features. These features in turn can be fed into a machine learning

**Data:** person set $P$, top rule set $T_{top}$

**for** *each person* $p \in P$ **do**
  Let $E_p$ be all epochs involving person $p$;
  Let $E_{p.f}$ be all features in $E_p$, start empty;
  **for** *each rule* $r \in T_{top}$ **do**
    $E_r$ = all epochs $e \in E_p$ where $X$ is fulfilled;
    **for** *each unimodal feature* $y \in Y$ **do**
      $r_{mean(y)} = mean(y, E_r)$ ;
      $r_{std(y)} = std(y, E_r)$;
      $E_{p.f} = E_{p.f} \cup \{r_{mean(y)}, r_{std(y)}\}$;
    **end**
  **end**
**end**

**Algorithm 2:** Contextually Filtered Features Extraction from Top Rules.

model to train a classifier.

For each rule $[X \rightarrow Y]$, I use $X$ as the "selector" (or filter) to select the days over which to aggregate (the days that fulfill the context feature sets, *i.e.*, the elements of $[X]$). For each element of $[Y]$, I calculate the mean and standard deviation using data from all of the filtered days. Fig. 4.2 visualizes an example of this process. Consider as an example, the rule $[X:$ {*Being at sport spaces, High activity level* $\} \rightarrow Y:$ {*Long phone call duration*}]. I select all time periods (*i.e.*, epochs), $E_p$ for person $p$ that fulfill the context {*Being at sports spaces, High activity level* }. Then, I calculate the average and standard deviation of the features in $Y$ only for the selected epochs. Thus in this example, the new contextually filtered features for the person $p$ are the mean and standard deviation of *Duration of outgoing call*, for all epochs in which the person spent a long time in sports spaces and had a high level of physical activity.

Alg. 2 presents the feature extraction procedure. Note that one rule can have multiple features in set $Y$; thus the number of features generated can be greater than the number of rules. There can also be duplicate features $y$ in $Y$ for different rules. However, as the context ($X$) of these rules are different, the contextually filtered feature calculated from the same $y$ in different rules is expected to have different mean and standard deviation values.

The final feature set for each person can be used for training classifiers to identify group membership of a person (*e.g.*, depression *vs.* non-depression group).

## *4.2 Evaluation*

I apply this approach to depression prediction among undergraduate students. I first introduce the implementation of rule mining, rule selection, and contextually filtered feature extraction in Ch. 4.2.1. I then highlight the results in Ch. 4.2.2.

### *4.2.1 Implementation*

#### *4.2.1.1 Data Set Preparation*

To avoid overfitting, I took one year of dataset (as described in Ch. 3) and randomly divided the dataset on a per-person basis into two subsets. I created a RuleGenerateSet of 35% of the people for extracting rules, and a TrainTestSet of the rest of the people to train and test the machine learning models. I applied the rule mining and selecting algorithm on the RuleGenerateSet and calculated the contextually filtered features on the TrainTestSet.

#### *4.2.1.2 Feature Selection*

I employed mutual information [149] to perform feature selection. I started with the whole feature set, repeated the calculation, and iteratively selected the intersection of the top 50 features until the number of features converged [100]. I went through the process on four epochs for weekdays (night 12am-6am, morning 6am-12pm, afternoon 12pm-6pm, evening 6pm-12am) and the same four epochs for weekends, resulting in 8 epochs in total. This process ended up with 23 top features on average among the 8 epochs (Min = 18, Max = 29, 181 in total). These 181 features were denoted as the *unimodal feature set*, which I used to train baseline classifiers since this approach is similar to the common practices used in previous literature related to depression prediction [195]. I considered the top features in each epoch group to identify the daily-epoch features to be used for rule mining. Specifically, a daily-epoch feature would be selected as long as either the mean or standard deviation of the feature is in the top feature set. I obtained an average of 20 (Min=15, Max=29) top

daily-epoch features in each epoch.

#### *4.2.1.3 Rule Mining and Selecting*

ARM is typically applied on symbolic or categorical data; thus I used the discretized top daily-epoch features, and fed them into my pipeline. Following Alg. 1, I employed the tools provided by [59] to mine rules 16 separate times: Once in each of the 8 epochs for each class of users. $sup_{min}$ and $conf_{min}$ are set in each group separately (0.07-0.19) to control the number of generated rules, leading to approximately 16,000 rules (Min = 4,500, Max = 26,000) among the groups. I used grid search for Equation 4.1, ranging from 0.0 to 2.0 with 0.5 as the interval, to set the best weights $(w_1, w_2, w_3)$ which were $(1.0, 1.5, 0.5)$. I used the F1 score in the RuleGenerateSet as the metric for selection, resulting in an average of 13 rules (Min = 6, Max = 19 rules) per epoch, 105 in total.

#### *4.2.1.4 Contextually-Filtered Feature Extraction*

After I obtained the rules, I turned to the TrainTestSet and used Alg. 2 to extract an average of 17 contextually filtered features (Min = 8, Max = 23) per epoch, 137 in total. Note that one rule $X \rightarrow Y$ can have multiple features $y$ in $Y$; thus the number of contextually filtered features generated can be greater than the number of rules. I aggregated each $y$ in $Y$, for each individual, using mean and standard deviation, over daily-epochs that matched $X$. This step added 274 (137×2) additional features to the unimodal features.

#### *4.2.1.5 Model Training*

I tested two feature sets: 1) Contextually Filtered Features: only the features extracted based on rules (vector length 274); 2) Hybrid Features: both the contextually filtered features and the unimodal features (vector length 455, 274+181). I employed AdaBoost [61] with decision-tree-based component classifiers during the training. Leave-one-user-out cross-validation was employed to avoid over-fitting, since previous work has consistently found that this method is approximately unbiased and has small variance [182, 231].

I compared the models with four baselines: 1) Majority: the classifier simply predicts the

major label in the dataset (*i.e.*, *no depressive symptoms*); 2) Best Single Feature: prediction is made based on the value of the single feature that best distinguishes the classes; 3) Class Association Rules: labels are embedded into the input during association rule mining and the generated rules are used for classification [113, 223]; 4) Unimodal Features: the model is trained on the unimodal features before rule mining (vector length 181, a common practice in previous work [195]).

### *4.2.2 Results*

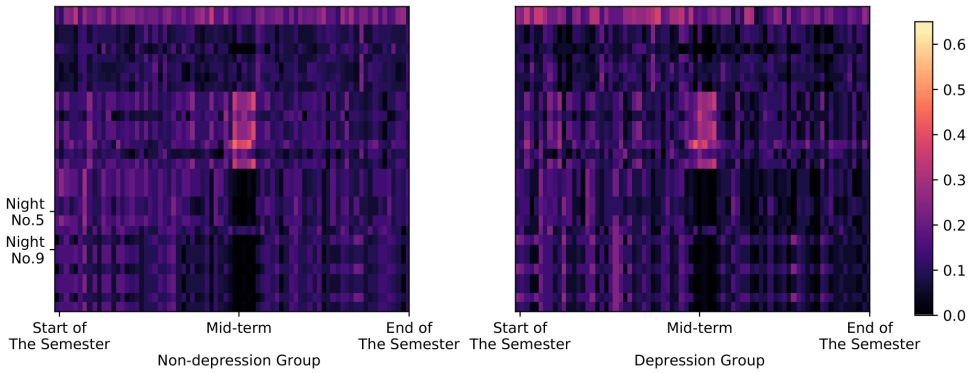
I first show that the top rules can capture the behavior differences between the student group with depressive symptoms and the student group without depressive symptoms. Then, I demonstrate that the best classifier trained on the contextually filtered features can achieve an average of 9.7% performance increase over the baseline model trained on unimodal features.

#### *4.2.2.1 Capturing Behavior Patterns with Rules*

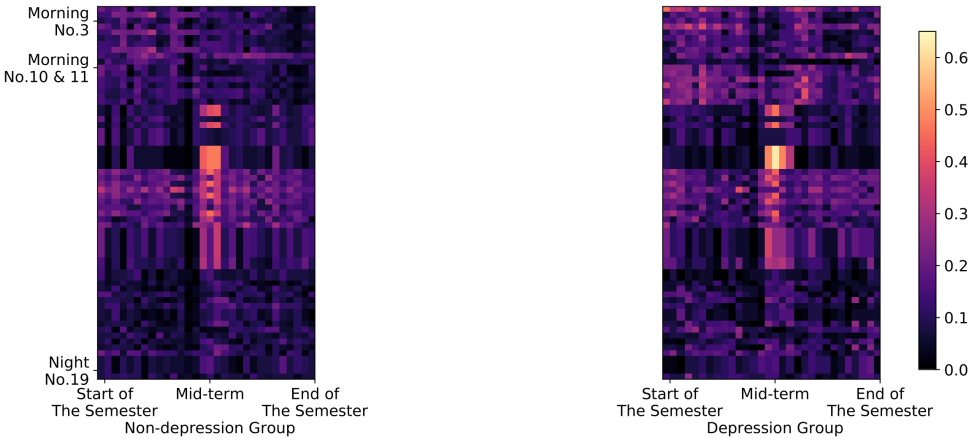
Fig. 4.3 visualizes heatmaps that represent how many students' behavior was captured by each rule throughout the study period. From both heatmaps of weekdays and weekends, abnormal color patterns are observed during the middle of the study. The academic calendar of the university showed that this period was when midterm examinations took place, followed by a spring break. Students tended to have a stressful period to prepare for examinations, and then have a brief relaxing period. As a result, during the midterm and break period, some rules became less frequent than the other times (represented by dark areas in the middle-top of the heatmap). This is positive evidence that contextually filtered features capture routine behavior, unlike their unimodal counterparts.

I further investigated the rules that capture different behavior patterns between students with depressive symptoms and students without depressive symptoms. I used a paired t-test on every rule to identify the rules that were significantly different between the two groups.

Tab. 4.1 show two examples. Weekday night rule No.5 indicates that students are likely to have good sleep quality when they are on campus and have low co-location (*i.e.*, the




(a) Weekday Rules (Left: Non-depression Group, Right: Depression Group)




(b) Weekend Rules (Left: Non-depression Group, Right: Depression Group)

Figure 4.3: Heatmaps of prevalence of the top rules among students with and without depressive symptoms, for weekends and weekdays. X axis is day of semester, Y axis shows rules aligned from morning to night epochs. Color indicates the proportion of students in a class that fulfill a particular rule. The brighter the color, the larger proportion of students having the pattern. The abnormal vertical color patterns in the middle of both figures correspond to the mid-term examines and the break period, indicating that the rules can capture people's routine behavior. Rule names on the left indicate some example rules that are significantly different between the two classes of participants.

Table 4.1: Examples of rules that capture behavior difference between students with and without depressive symptoms. We tested a rule's ability to differentiate between classes using a paired t-test; significance level is indicated in the *Rule* column. We selected rules for this table that show the strongest significant difference. *Type* is the method by which the rule was found. *Prop in Non-dep* and *Prop in Dep* are the proportion of students in a class that fulfill the rule, averaged over days in the study. Note that $M$ varies between different epochs (people can have different behavior pattern during the day) as well as their types (*i.e.*, common or unique).

| **Rule** | $X$ | $Y$ | **Type** | **Prop in Non-dep** | **Prop in Dep** | ***Ctx Spec*** | ***Conf Diff*** | ***Cond Disc*** | $M$ |
|---|---|---|---|---|---|---|---|---|---|
| Wkdy Night No.5*** | - [CampusMap] Percentage of time off-campus (low)<br>- [Bluetooth] Number of unique device of others (low) | [Sleep] Sleep efficiency (high) | Common | 11.4% | 8.5% | 2 | 0.137 | 0.094 | 0.031 |
| Wkend Night No.19* | - [Screen] Mean length of screen being unlock (high) | [Sleep] Mean length of being asleep (low) | Unique (Dep) | 7.8% | 10.3% | 1 | 0.387 | 0.374 | 0.147 |

* indicates $p < 0.05$, ** indicates $p < 0.01$, *** indicates $p < 0.001$

number of Bluetooth encounters) during weekday nights. This rule is in both groups, but appears significantly more in the non-depression group ($t_{75} = 3.99, p < 0.001$). Moreover, Weekend night rule No. 19 is only present in the depression group. This suggests the potential effect of phone usage on sleep quality for depressive students.

These results indicate that the rules extracted by my method have good interpretability and can help better understand students' life experiences related to depressive symptoms. The observations can be supported by relevant findings in psychology and clinical psychiatry that sleep disturbance and lack of focus are common symptoms of depression [16, 181, 186].

Table 4.2: Comparison of baseline machine learning classifiers and contextually filtered features. The models above the dashed line are baselines. Models based on unimodal features, contextually filtered features, and hybrid features, are trained using AdaBoost [61] with decision-tree-based component classifiers, with leave-one-out cross-validation. The number of estimator and the maximum depth of the decision-tree are hyper-parameters that can be tuned. We use grid search to select the best parameters for each model. Our best model with hybrid features has the number of estimator as 10 and the maximum depth as 3. A t-test on the test results between the hybrid features and unimodal raw features show that the new method significantly outperforms the standard method ($p < 0.01$).

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1 Score** |
|---|---|---|---|---|
| Majority | 0.579 | 0.579 | 1.000 | 0.734 |
| Best Single Feature | 0.704 | 0.725 | 0.755 | 0.740 |
| Class Association Rules [223] | 0.608 | 0.629 | 0.850 | 0.723 |
| Unimodal Features with AdaBoost [61] | 0.716 | 0.725 | 0.771 | 0.747 |
| Contextually Filtered Features with AdaBoost [61] | 0.807 | 0.765 | **0.886** | 0.821 |
| Hybrid Features with AdaBoost [61] | **0.818** | **0.843** | 0.843 | **0.843** |

#### *4.2.2.2 Achieving Better Modeling Performance*

Other than achieving interpretation of daily behavior, my method is able to achieve better depression prediction performance. Tab. 4.2 summarizes the model results with four metrics: accuracy, precision, recall, and F1 score. The model trained on the hybrid features has the best performance, followed by the model trained on contextually filtered features. My best model has an accuracy of 0.818 and an F1 score of 0.843. It outperforms the baseline model using the unimodal features, by an average of 9.7% absolute increase, indicating the effectiveness of the method. These baselines provide strong evidence that my model is either better than previous work [29, 54, 195], or comparable to the state-of-the-art [156, 201].

We further examined the effect of each characteristic, using an ablation study on the three weights. We set one of the weights to zero in each trial and redo the rule selection,

Table 4.3: Results of the ablation study. One of the three weight values is set to zero in each trial, which can lead to different rule sets, and new models are trained based on these rules. The other weights $(w_1, w_2, w_3)$ are set to $(1.0, 1.5, 0.5)$.

| Classification | Ablated Metric | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|---|
| Depression Detection with Contextually Filtered Features | *ConfDiff* - $w_2 = 0$ | 0.761 | 0.804 | 0.788 | 0.796 |
| | *CtxSpec* - $w_1 = 0$ | 0.761 | 0.863 | 0.759 | 0.807 |
| | *CondDisc* - $w_3 = 0$ | 0.784 | 0.808 | 0.824 | 0.816 |
| | No Metric Ablated | 0.807 | 0.765 | 0.886 | 0.821 |

feature extraction, and modeling training. Tab. 4.3 summarizes the results. Removing *CtxSpec* ($w_1 = 0$) and *ConfDiff* ($w_2 = 0$) lead to similar results, with both models having a drop in accuracy of 4.5%. The model without *CtxSpec* has a slightly higher F1 score than without *ConfDiff*. Removing *CondDisc* ($w_3 = 0$) has the least impact on the results, with two percentage-points drop in accuracy. These results are consistent with the relative order of weight values. Confidence Difference is the most essential part in the metric $M$, and the Condition Discrepancy is the least important part.

### *4.3 Summary*

In this chapter, I presented a new method based on association rule mining for generating contextually filtered features in an automated way, which performed better than standard feature selection approaches for depression prediction. I showed that the best rules selected by my method were highly interpretable and captured students' routine behaviors, and behavior pattern differences between students with and without depressive symptoms. I demonstrated that my best model outperformed a standard model by an average of 9.7% across various metrics.

# Chapter 5

# PERSONALIZED BEHAVIOR MODELING

Most of the existing algorithms and models, including the one I introduced in Ch. 4, follow a one-size-fits-all (*i.e.*, population modeling) approach that looks for common behaviors amongst all users. Such a method disregards the fact that individuals can behave very differently, resulting in reduced model performance. Moreover, the interpretation results in Ch. 4 is only population-level behavior understanding. This leads to our second research question (**RQ2**): *How to model individuals in light of high variability?*.

In this chapter, I present a new method to address the problems of personalization. With depression prediction as the main application, the method uses a collaborative-filtering-based concept and leverage a unique subset of users with relevant behavior trajectories for personalized prediction and interpretation. The details of the method are introduced in Ch. 5.1. I then summarize the evaluation results in Ch. 5.2. Finally, I highlight the key conclusion in Ch. 5.3.

## *5.1 Methods*

The key idea of the approach is to: **leveraging users whose behavior is relevant to the target user to achieve personalized prediction and interpretation.** Fig. 5.1 visualizes the overall framework.

### *5.1.1 Personalized Classification*

I first introduce the concept of user behavioral profile and its relationship with collaborative filtering (Sec. 5.1.1.1). Then, I leverage the correlation matrix from user behavioral profiles to impute missing data (Sec. 5.1.1.2). I then propose a measurement of the behavior relevance (square of the correlation) using the imputed behavioral profiles (Sec. 5.1.1.3) and use the metric to select features with good performance in the training set (Sec. 5.1.1.4). When

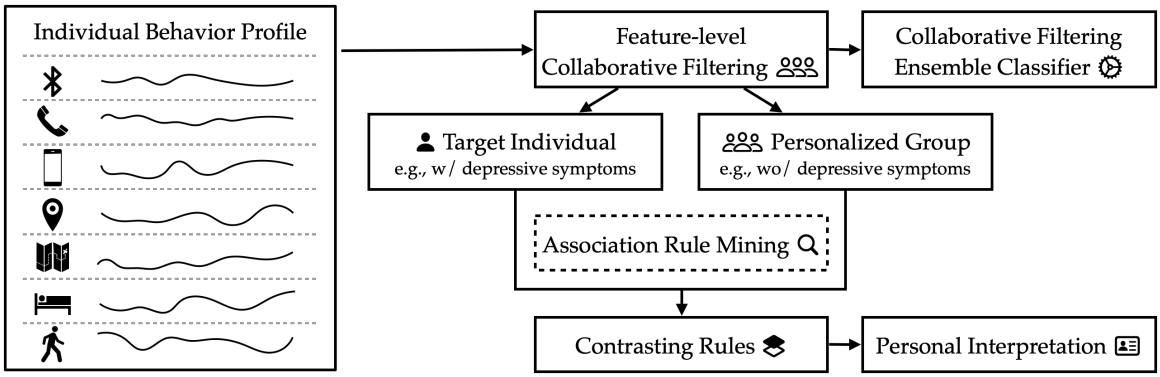



Figure 5.1: The Framework of Personalized Modeling. For each target individual, I calculate their behavior relevance scores against everyone in the training user pool and select a subset of users with high scores. These users' labels are first used to predict the target user's label. And their behavior data are further leveraged to generate personalized interpretation following the pipeline in Ch. 4.

a new testing user is added to the analysis, I employ the selected features to generate intermediate classification outputs. Finally, I use majority voting to compile the intermediate results into the final classification output (Sec. 5.1.1.5).

#### *5.1.1.1 Collaborative Filtering and Behavior Relevance Metric*

My method is inspired by the idea of memory-based collaborative filtering [88]. Borrowing the idea of user-level collaborative filtering, I propose the concept of a *user-behavior profile* to depict a group of users' behaviors. Each user-behavior profile, represented by a matrix (see Fig. 5.2), focuses on one particular feature. The user-behavior profile can be viewed as a user-item matrix from traditional collaborative filtering. The method looks at each feature independently. Therefore, I also include users' target labels (*i.e.*, ground truth labels) in each behavior profile (the bold frame marked with $L$ in Fig. 5.2), which can be regarded as a column of "special items" in the profile matrix. A new user's "special item", marked by $X$, is the element that needs to be predicted (missing values are marked as ?).

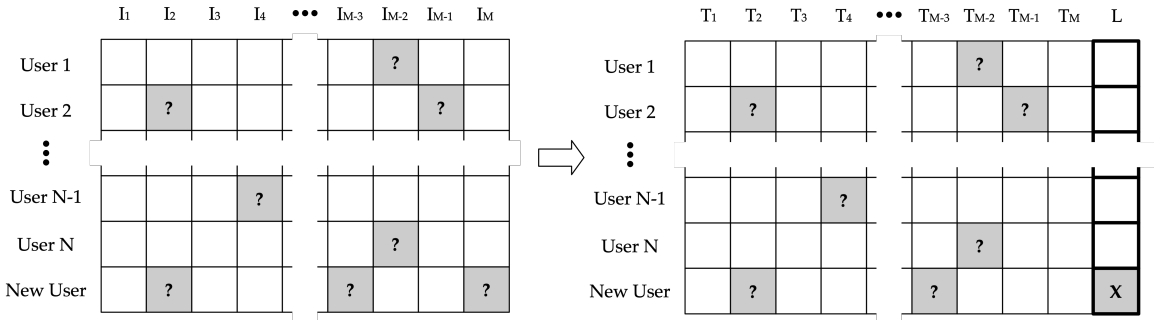



Figure 5.2: The similarity between our method and collaborative filtering. The left part shows a **user-item** matrix that is commonly used in recommendation systems, where the "?" marks indicate preference scores to be predicted. In contrast, the right part shows the common data format in mobile sensing. The matrix is the **behavior profile** of one particular feature (each user has a time series data from $T_1$ to $T_M$), plus a column of labels (Column $L$). where "?" marks indicate the missing value and "X" mark indicate the target label to be predicted, *e.g.*, whether the new user is depressed or not.

#### *5.1.1.2 Data Imputation*

After constructing the user behavior profile for each feature, I impute the missing data in the behavior profiles. Following Ch. 4, I employed the normalized version of each user's features. Then, I use the weighted average of other users' normalized data as the imputed value, where the weights are the correlation of the longitudinal feature value between this user and other users. The intuition is to leverage the data from people with similar behavior (for the feature that the data represents) to impute the missing value. Alg. 3 lists the detailed steps for data imputation.

#### *5.1.1.3 Behavior Relevance Metric*

From the imputed data, I propose a *behavior relevance metric*. Prior work focuses on people with similar behaviors (*e.g.*, [6, 104]). However, when predicting the final outcome (*e.g.*, having depressive symptoms or not), the scope can be expanded. People whose behaviors are strongly related to a target user can be divided into two types. People who behave

similarly (with a strong positive correlation) are the ones whose behavior patterns change in a similar way as the target user, *e.g.*, they all have more phone calls during the weekend. Moreover, there are also people who behave differently (with a strong negative correlation). Here, behavior patterns change in a way opposite to the target user, *e.g.*, the target user mostly stays at home and moves less on weekday evenings, while these people often hang out for socializing at that time. Both types of people are relevant to a target user. Their similarities and differences may be indicators for classification (*e.g.*, social behaviors are related to depressive symptoms); thus both should have representation. Therefore, I further define the *behavior relevance metric* as the square of the Pearson correlation coefficient.

**Data:**

$E$: the epoch set; $D$: the days in the dataset; $U$: users in the training set;

$F$: the overall feature set. $F_e$: the feature set of a particular epoch $e$ ($\subseteq E$), $F_e \subseteq F$;

$R_{E,F}$: the list of raw behavior profiles. Each matrix $R_{e,f}$ ($|U| \times |D|$) is the behavior profile of feature $f$ ($\subseteq F_e$) in epoch $e$ ($\subseteq E$). $R_{e,f}$'s rows and columns can be indexed by a number or a list. For example, $R_{e,f}[u, d]$ locates the feature value of user $u$ on day $d$. $R_{e,f}[u, D]$ locates the feature array of user $u$ in days $D$. The same can be applied to other matrices in algorithms;

$P_{E,F} = Copy(R_{E,F})$ ; // The placeholder of the imputed behavior profiles

**for** $f$ **in** $F$ **do**

  $e = GetEpoch(f, E)$;

  $Cor_U = PairwiseCor(R_{e,f}, R_{e,f})$ ; // Calculate the user-pairwise correlation matrix (|U|x|U|) with the missing value ignored

  **for** $u$ **in** $U$ **do**

    $D_m = FindDayMissing(R_{e,f}[u, D])$ ; // Get the days where u has missing data

    **for** $d$ **in** $D_m$ **do**

      $U_{nm} = FindUsersNotMissing(R_{e,f}[U, d])$ ; // Find users who have data on d

      $weight_u = Cor_U[u, U_{nm}]$ ; // Use correlation scores as weights

      $P_{e,f}[u, d] = WeightedAvg(R_{e,f}[U_{nm}, d], weight_u)$ ; // Impute the missing data

    **end**

  **end**

**end**

$Return(P_{E,F})$ ; // Return the imputed behavior profiles

**Algorithm 3:** Data Imputation.

**Data:**

$E$, $F$, $F_e$, $D$, $U$ same as Algorithm 3;

$L$: the label list in the training set. $|L| = |U|$. The list can be indexed by a user $u$ to get the label $L[u]$, or by a list of users $U$ to get the label list $L[U]$. The same can be applied to other lists/arrays;

$P_{E,F}$: the list of imputed behavior profiles. Each matrix $P_{e,f}$ ($|U| \times |D|$) is the imputed behavior profile of feature $f$ ($\subseteq F_e$) in epoch $e$ ($\in E$);

$RankingScore_F = EmptyArrayWithSize(F); Threshold_F = EmptyArrayWithSize(F)$ ;
**for** $u_{vd}$ **in** $U$ **do**
    $U_{tr} = U \setminus \{u_{vd}\}$ ; // Evaluate across each training user to ensure stability
    $accs = EmptyArrayWithSize(F)$;
    **for** $f$ **in** $F$ **do**
        $e = GetEpoch(f, E)$;
        $Cor_{U_{tr}} = PairwiseCor(P_{e,f}[U_{tr}], P_{e,f}[U_{tr}])$ ; // Calculate user-user correlation matrix
        $Rel_{U_{tr}} = Cor_{U_{tr}} \odot Cor_{U_{tr}}$ ; // Correlation square as the relevance matrix
        $weights = Rel_{U_{tr}} - diag(Rel_{U_{tr}})$ ; // Relavance metrics against others
        $labelscores = WeightedAvg(L[U_{tr}], weights)$ ; // Label scores calculated from others
        $th1 = Avg(labelscores[\{u; u \in U_{tr}, L[u] = T\}]); th2 = Avg(labelscores[\{u; u \in U_{tr}, L[u] = F\}])$;
        $th = Avg(th1, th2)$ ; // Splitting threshold
        $UpdateAvg(Threshold_F[f], th)$;
        $accs[f] = AccuracyByThreshold(labelscores, th, L[U_{tr}])$;
    **end**
    $Filter(accs, th = 0.5)$ ; // Remove features that perform poorly on the training set
    **for** $f$ **in** $F$ **do**
        **if** $Rank(f, accs) \in TopRank(accs)$ ; // Assign ranking scores
        **then** $RankingScore_F[f] += Rank(f, accs)$ ; **else** $RankingScore_F[f] += 0$ ;
    **end**
**end**
$SF = SelectTopFeatures(RankingScore_F)$;
$TH = Threshold_F[SF]$;
$Return(SF, TH)$ ; // Return the selected features and their corresponding thresholds

**Algorithm 4:** Feature Selection

#### *5.1.1.4 Feature Selection*

I group raw sensor data into 10 epochs (morning, afternoon, evening, night, and the whole day ×weekday and weekends) that capture behavior at different times. This further increases the number of behavior features by order of magnitude. Therefore, selecting the most helpful

features for distinguishing target labels is needed.

Using the behavior relevance metric, I conduct a feature selection process on the training set. For each feature, an inner leave-one-user-out loop is used to find the most important and stable features. Specifically, I take one user within the training set as the "validating user" each time (the rest are "training users"), and compute *label score*s for all training users, by calculating the weighted-average of the label value (False is -1 and True is 1), with the relevance scores (against other training users) as weights. Then, I calculate the average of these scores among users with False labels and another average among users with True labels. I use the mean of the two average scores as the splitting threshold. To see how well this feature works, its threshold is compared against each training user's label score to get a tentative label. Having the labels and the ground truth, the average accuracy for this feature from training users can be obtained. I filter out features whose validation accuracy is below 0.5 and assign a ranking score for the top ten percentile among the remaining features according to the accuracy value (score $n$ for $n^{th}$ best feature) and zero for other features. I repeat this across each "validating user" and get a series of ranking scores for each feature. I then sum the score and pick half features with the lowest scores (*i.e.*, top five percentile features) as the best features. Alg. 4 lists the detailed steps for feature selection.

#### *5.1.1.5 Majority Voting*

When a testing user is added to the analysis (with already collected data), I first calculate their relevance scores (against the users in the training set) for only the selected features. Then, for each selected feature, I filter out the training users whose relevance score is among the bottom-quartile (*i.e.*, bottom 25 percentile, a conservative threshold [85]) as their behavior is not relevant to the new user and can introduce noise. This leads to a unique, personalized training set for each new user. For each selected feature, I calculate a weighted-average label score for the new user, and obtain an intermediate classification output using the splitting threshold calculated from users in the training set. In other words, for each selected feature, using data from the remaining users for that feature, a classification result just for that feature is produced. Finally, I use a majority voting approach to aggregate

**Data:**

$E$, $D$, $U$, $L$ same as Algorithm 3;

$SF$: the selected feature set from Algorithm 4; $SF_e$: the selected feature set of a particular epoch $e$ ($\subseteq E$);

$TH$: the threshold lists corresponding to the selected features from Algorithm 4;

$P^U_{E,SF}$: the list of behavior profiles of training users $U$. Each matrix $P^U_{e,f}$ ($|U| \times |D|$) is the behavior profile of feature $f$ ($\subseteq SF_e$) in epoch $e$ ($\in E$);

$u_t$: a testing user; $P^{u_t}_{E,SF}$: the list of behavior profiles of the testing user $t$. Each matrix $P^t_{e,f}$ ($1 \times |D|$) is the behavior profile of feature $f$ ($\subseteq SF_e$) in epoch $e$ ($\in E$).

$Results = EmptyArrayWithSize(SF)$;
**for** $f$ **in** $SF$ **do**
    $e = GetEpoch(f, E)$;
    $Cor_{u_t} = PairwiseCor(P^{u_t}_{e,f}, P^U_{e,f})$ ;  // Calculate the correlation matrix (1x|U|) between the testing users and users in the training set
    $Rel_{u_t} = Cor_{u_t} \odot Cor_{u_t}$ ;  // Correlation square as the relevance scores
    $U_t = FilterUsers(Rel_{u_t})$ ;  // Remove users whose similary is among bottom-quartile
    $weight_t = Rel_{u_t}[u_t, U_t]$;
    $score = WeightedAvg(L[U_t], weight_t)$;
    **if** $score > TH[f]$ **then** $Results[f] = TRUE$ **else** $Results[f] = FALSE$
**end**
$FinalResult = MajorityVoting(Results)$ ;  // Majority voting across epochs
$Return(FinalResult)$;

**Algorithm 5:** Memory-based Classification

these features' intermediate outputs, as shown in Alg. 5.

### *5.1.2 Personalized Interpretation*

Beyond classification, I further propose a method that combines the relevance metric with ARM to provide personalized interpretation. In Ch. 4, I applied ARM on the whole participant group to generate popular behavior rules among the population. In contrast, here I propose to focus on a single user's data for personalized interpretation. The interpretation focuses on generating personalized behavior rules that can capture the behavior differences between target users and other users, provide more insights into their life experiences, and suggest potential ways to support behavior changes to achieve a desired goal.

As the interpretation is focused on behavior distinctions, I identify a small subset of

users whose behaviors are very different from a target user (Sec. 5.1.2.1). Then, I leverage ARM to mine frequent behavior rules separately (Sec. 5.1.2.2). Finally, I identify the rules that can provide the most meaningful information (Sec. 5.1.2.3).

#### *5.1.2.1 Identifying Informative User Groups with Negative Correlated Behavior*

Different users have different degrees of similarities when compared to a target user. In order to generate personalized interpretation, I need to first identify a group of users that are the most informative. Users in the group have different labels than the target user, and very different behavior on the selected features. For instance, if the target user is a student with depressive symptoms, then the identified group would be the subset of the users who do not have depressive symptoms and whose behavior features are strongly relevant to the target user, but in a negative direction, *i.e.*, strong negative correlation (among the top-quartile for a given behavioral feature). It is worth noting that this process is conducted on each target user and each selected feature individually. Therefore, the identified groups are personalized to every user.

#### *5.1.2.2 Behavior Rule Mining*

I use one target user $t$ and one selected feature $f_s$, together with the identified user group $g$ as examples when introducing the next two steps. Given the target user $t$, I create an identified user group $g$ in terms of the selected feature $f_s$. I then employ ARM on the discretized features two times to mine frequent behavior rules, once using the target user $t$'s data and again using the identified group $g$'s data. The output rules of ARM are in the form of $X \rightarrow Y$ with support $P(X)$ and confidence $P(Y|X)$, where both $X$ and $Y$ are a set of discretized features at certain levels.

Each selected feature $f_s$ belongs to a particular epoch. The rule mining is performed on the whole feature set within the epoch, which outputs a large number of behavior rules. An informative rule should suggest meaningful behavior changes to influence the final outcome. To identify these rules, I only focus on the rules whose $Y$ includes the selected feature $f_s$, because $f_s$ affects the classification result during the majority voting procedure. I dynami-

cally adjust ARM support and confidence thresholds to ensure the number of behavior rules from the target user and the identified group is no less than ten thousand.

#### *5.1.2.3 Behavior Rule Pairing*

With the rules from the target user side $t$ and the identified group side $g$, two sides need to be comparable. I propose a rule pairing approach to align the rules from the two sides.

If two rules (one from each side) have exactly the same antecedent and a similar consequent, they can be aligned. Specifically, two rules will be paired if they have identical $X$ (the same features at the same discretized value levels) and the same features in $Y$ (but not necessarily the same level). For example, if the target user has a rule $R_t$ as $X_t : \{f_x(low)\} \rightarrow Y_t : \{f_s(medium)\}$, a rule to be paired on the identified group $R_g$ needs to have $X_g$ exactly same as $X_t$, *i.e.*, $X_g : \{f_x(low)\}$. Meanwhile, its $Y_g$ needs to have the same features $f_s$, but not necessarily at the same level, *i.e.*, $Y_g : \{f_s[at\ any\ level]\}$.

On each side, it is possible that among the selected rules, there might be multiple rules having identical $X$ and same features in $Y$, but at different feature value levels. Continuing the example, on the identified group side, one rule $R_{g1}$ is $X_g : \{f_x(low)\} \rightarrow Y_g : \{f_s(medium)\}$, while another rule $R_{g2}$ is $X_g : \{f_x(low)\} \rightarrow Y_g : \{f_s(high)\}$. This will lead to multiple pairs of rules, *i.e.*, $R_t$ can be paired with both $R_{g1}$ and $R_{g2}$. However, $R_{g1}$ and $R_{g2}$ appear with different frequencies in the dataset, as represented by their confidence values (their support values are the same because of the same $X_g$). Including both pairs will introduce additional noise. Therefore, for each $X$, I only retain the rule with the highest confidence and discard other rules, as this rule is the most representative and indicates the most common behavior when $X$ appears. Then, I pair these representative rules following the description above and discard the unpaired rules. Once the rules are paired, I select the top three rule pairs that have the most significant confidence gap between the two sides for interpretation. This step finds the largest behavior differences between the target user and the identified group. I repeat the process for each selected feature to obtain a set of personalized behavior rules for the target user.

The pipeline is described in Alg. 6. The whole pipeline is conducted on each target user

independently. Thus the interpretation provided by the final selected rules is personalized.

### *5.2 Evaluation*

I used one year of dataset from Ch. 3 to evaluate the performance of the algorithm and compared it against baseline methods. I then demonstrated how my method can generate

**Data:**

$E$, $F$, $F_e$, $D$, $U$, $L$ same as Algorithm 3; $SF$: the selected feature set from Algorithm 4;

$u_t$: a target user with $L[u_t] = target$; $U_{ntar}$: non-target users with $L[U_{ntar}] \neq target$, *e.g.*, with vs. without depressive symptoms;

$P_{E,F}$: the list of behavior profiles. Each matrix $P_{e,f}$ ($|U| \times |D|$) is the imputed behavior profile of feature $f$ ($\subseteq F_e$) in epoch $e$ ($\in E$).

$PersonalizedRules = EmptyList()$;
**for** $f$ **in** $SF$ **do**
    $e = GetEpoch(f, E)$;
    // Get the identified group whose behavior are most negatively correlated
    $P_{u_t} = P_{e,f}[u_t, D]; P_{U_{ntar}} = P_{e,f}[U_{ntar}, D]$;
    $Cor_{u_t} = PairwiseCor(P_{u_t}, P_{U_{ntar}})$ ; // Calculate correlation scores between each target user and non-target users
    $Rel_{u_t} = Cor_{u_t} \odot Cor_{u_t}$;
    $filter(Rel_{u_t}, Sign(Cor_{u_t}) < 0)$ ; // Focus on users with negative correlation
    $U_{idt} = GetTopUsers(Rel_{u_t})$ ; // Get the top quartile users as the identified group
    // Mine behavior rules seperately, focusing on rules with $f$ in Y
    $P^{l}_{u_t} = \{P_{e,f}[u_t, D]; f \in F_e\}; P^{l}_{U_{idt}} = \{P_{e,f}[U_{idt}, D]; f \in F_e\}$ ; // Get full behavior set
    $BehaviorRules_{u_t} = AssociationRuleMining(P^{l}_{u_t}, f)$;
    $BehaviorRules_{U_{idt}} = AssociationRuleMining(P^{l}_{U_{idt}}, f)$;
    // Focus on the rules with the highest confidence under each context, i.e., X
    $BehaviorRules_{u_t} = UniqueContext(BehaviorRules_{u_t})$;
    $BehaviorRules_{U_{idt}} = UniqueContext(BehaviorRules_{U_{idt}})$;
    $PairedRules = Pair(BehaviorRules_{u_t}, BehaviorRules_{U_{idt}})$ ; // Same context X
    // Select the top three rules with largest gap on rule confidence
    $TopRules = GetTopRules(PairedRules)$;
    $Append(PersonalizedRules, TopRules)$;
**end**
$Return(PersonalizedRules)$;

**Algorithm 6:** Personalized Interpretation

personalized interpretation and inspected examples of personalized behavior rules.

### *5.2.1 Implementation*

As there was no need for hyperparameter tuning, the implementation was straightforward from Alg. 3 to Alg. 6.

I compared the method with a few closely related baseline methods on the same dataset. I employed leave-one-user-out cross-validation to avoid over-fitting.

1. Majority, a baseline that simply classifies all samples as the major class in the dataset.
2. Single Best Threshold, a simple threshold-based method that uses the best single mean & variance aggregated feature as the splitting threshold.
3. K-Nearest Neighbour (KNN), a typical similarity-based method that uses similar neighbours for classification [88]. I adopted Euclidean distance of all features as the similarity measurement. $K$ was set as 5.
4. Lazy Bayesian Rules (LBR), a classical sample-specific personalized algorithm that builds a naive Bayesian classifier for each test sample when it appears [232].
5. Long short-term memory (LSTM), a neural network commonly used for time series data [157]. I used a small two-layer bidirectional LSTM, both with 16 hidden units and each users' feature across the whole study period as one data point.
6. Personalized Logistic Regression (PLR), a recent state-of-the-art sample-specific personalized algorithm [107]. It assigns specific parameters for each sample, with low-rank representation and external covariates as the approaches to limit the parameters' degree of freedom. Based on grid search, I set the rank as 10 and the number of neighbors as 3.
7. Multi-sensor Classifier (MSC), a popular method for depression prediction that concatenates multiple sensors' average feature value and trains a classifier with off-the-shelf models. It closely replicates some previous work (*e.g.*, [54, 156, 195]). I used random forest, with the maximum depth and the tree numbers 5 and 30 based on tuning.
8. Contextually-Filtered Classifier (CFC), the method introduced in Ch. 4. Note that CFC is a population modeling approach. The hyperparameters of the final AdaBoost decision-

Table 5.1: Comparison of baseline and state-of-the-art machine learning classifiers and our new algorithm. T-tests on both the balanced accuracy and the F1 score between our method and the best baseline CFC show that our method significantly outperforms the baseline method ($p < 0.05$ in all cases).

| **Algorithm** | **Accuracy** | **Bal Accuracy** | **Precision** | **Recall** | **F1 Score** |
|---|---|---|---|---|---|
| Majority | 0.608 | 0.500 | 0.608 | 1.000 | 0.756 |
| Single | 0.639 | 0.679 | 0.853 | 0.492 | 0.624 |
| KNN [88] | 0.567 | 0.527 | 0.627 | 0.712 | 0.667 |
| LBR [232] | 0.629 | 0.615 | 0.702 | 0.678 | 0.690 |
| LSTM [157] | 0.557 | 0.462 | 0.589 | 0.898 | 0.711 |
| PLR [107] | 0.667 | 0.668 | 0.765 | 0.661 | 0.709 |
| MSC [54, 156, 195] | 0.716 | 0.700 | 0.725 | 0.771 | 0.747 |
| CFC [209] | 0.773 | 0.781 | 0.863 | 0.746 | 0.800 |
| **Our Algorithm** | **0.825** | **0.819** | **0.862** | **0.847** | **0.855** |

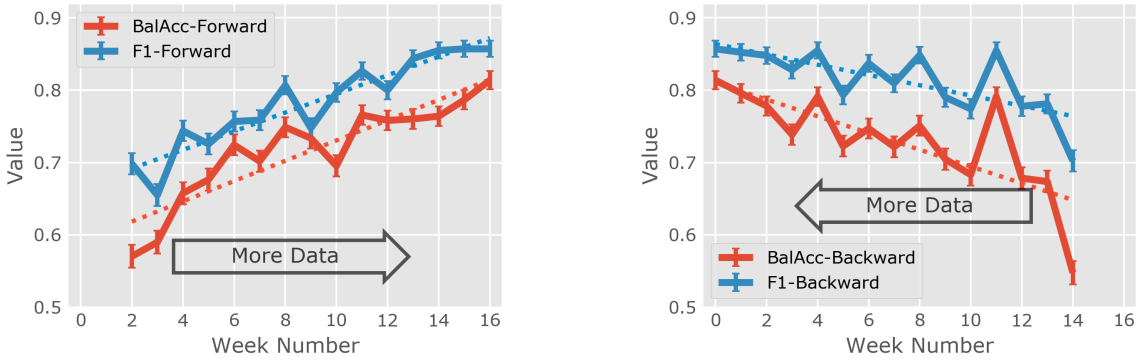
tree-based classifier, *i.e.*, maximum depth and the number of the estimator, were set as 5 and 20 based on tuning.

### *5.2.2 Results*

#### *5.2.2.1 Achieving Better Modeling Performance*

**Overall Performance.** The modeling results are summarized in Tab. 5.1 in terms of five metrics: accuracy (the overall success rate), balanced accuracy (taking the imbalance on labels into account), precision, recall, and F1 score. Compared to the best-performing baseline model (CFC), my method's results have improvement in most metrics, particularly the accuracy (5.1%), the recall (10.1%) and the F1 score (5.5%). I also investigated how much data is needed in order to obtain a satisfactory performance of my algorithm.

**How Many Days Does The Algorithm Need?** I evaluated the effect of the number




(a) Counting Forward

(b) Counting Backward

Figure 5.3: The results when using data from different numbers of weeks. Forward means using the data between the beginning of the data collection period (the beginning of the semester) and the particular week number, while backward means using the data between the particular week number till the end of the period (the end of the semester). Error bars indicate the standard error of the mean.

of days of data from two perspectives: forward and backward. Given a particular week number, forward means only using the data between the beginning of a study and the week number, while backward means only using the data between the week number through until the end of the period, *i.e.*, the time when students finished the final BDI-II survey. These two perspectives are complementary. The first perspective indicates how early we can use the collected data to predict the depression status at the end of the semester, while the second perspective indicates how depression can be reflected in the most recent behavior. Fig. 5.3 visualizes the results of both perspectives.

In general, both figures present an increasing trend on the two metrics as the number of days used for training increases. Some interesting phenomena are found: In the counting forward approach, it follows an overall trend that the more data we have, the better performance we can achieve. The performance of only using the data from the first several weeks to predict the depressive status at the end of the semester is not satisfactory. Moreover,

there is a small drop in balanced accuracy from week 8 to week 10, accompanied by a smaller drop in F1 score (see Fig. 5.3a). This was during the mid-term period and students might have been busy preparing for exams, indicating that such a break in their routine might affect the effectiveness of the relevance metric for depression detection. From the backward perspective, there is a peak in balanced accuracy using five weeks of data (end of study back to week 11). The F1 score also has a small peak, but it is less significant. This suggests that a one-month period could be a good time window for signaling depression status.

**How Many People Does The Algorithm Need?** I also evaluated the method in terms of the number of users required to establish a good training set. To determine this, I uniformly sample a certain number of users from the whole dataset and call this the training dataset. The remaining users comprise the testing dataset. The process is repeated one hundred times to obtain the mean and the standard error. Fig. 5.4 visualizes the results.

Not surprisingly, both the balanced accuracy and the F1 score increase monotonically as the number of users increases. The more users in the training set, the more likely a testing user can find users with similar or opposite behavior, leading to better results. Such an increase becomes slower when the number of users is above 60. Both metrics are close to a plateau when there are 60 participants.

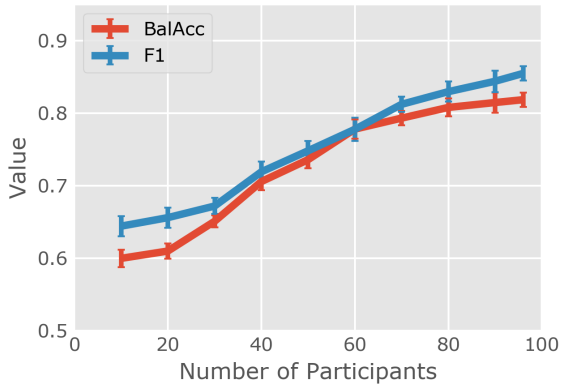



Figure 5.4: The results when using different numbers of users for training. Each data point is the mean of one hundred random samples from the dataset.

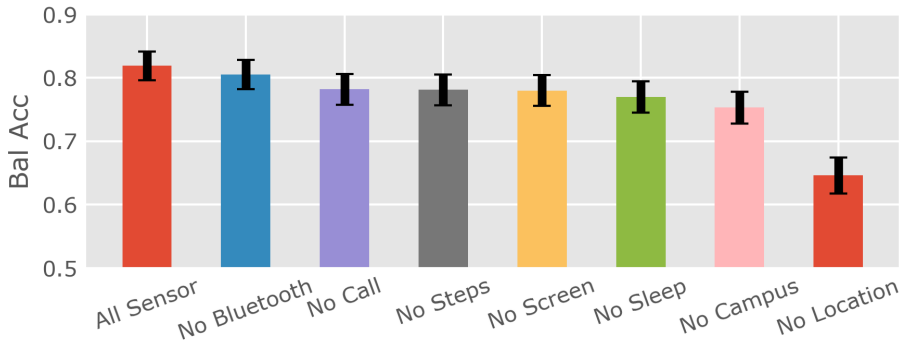



Figure 5.5: The balanced accuracy results of the feature ablation study. Each time one of the seven feature types is removed and the whole pipeline is applied on the remaining features.

**How Important is Each Feature Type?** In order to investigate the effect of each feature type, I further conducted a feature ablation study. For the seven feature types – phone screen, call, Bluetooth, location, campus, sleep, and step – I removed one of them and re-ran the whole pipeline using the remaining six feature types. Fig. 5.5 summarized the results.

The results show that the two mobility-related feature types (location and campus) were the most important, and removing them leads to the biggest drop in the balanced accuracy (17.3% and 6.6% absolute value, respectively). This is supported by the previous literature [54, 156]. In contrast, removing Bluetooth feature has the least effect, with 1.4% absolute value drop.

#### *5.2.2.2 Obtaining Individual-Level Interpretability*

Beyond achieving good classification results, I further show that the method is able to generate a personalized understanding of individual students, especially those with depressive symptoms.

The behavior rules generated from my method are tailored for every student with positive labels (*i.e.*, having depressive symptoms). I randomly picked two students with moderate

depression (P18 with post BDI-II score 24, and P72 with post BDI-II score 26) as anecdotal examples and investigated how the personalized rules capture behavior differences from their identified user groups (students without depressive symptoms) with different behaviors. Tab. 5.2 lists out a subset of these rules.

The majority of the rules between the two students are different. It is expected as these two students did not have identical behavior. Interestingly, close inspections of the rules reveal both the homogeneity and the heterogeneity at the same time. Some of their behaviors share commonalities that are supported by existing depression-related literature, while some other behaviors are quite different between the two.

The top four rows of Tab. 5.2 shows examples of the homogeneity of sleep pattern and phone usage behavior. P18's weekend morning rule No.4 indicates that P18 had more interrupted sleep (higher number of sleep bouts) compared to their identified group on weekend mornings when they were not at social spaces or dorm, and their total numbers of sleep bouts (including being asleep, restless, awake) were low. Similarly, P72's weekday all-day rule No.3 indicates that P72 had shorter sleep duration than the identified group when they had fewer incoming calls and were away from social spaces. Both rules indicate that the two students' sleep patterns were disturbed. Similar sleep patterns were consistent among other students. Among the 38 students who were labeled as being depressed, **92.1% of them (35 out of 38) had more than half of the rules about sleep showing a more disturbed sleep pattern than in their respective identified group.** Moreover, 73.7% (28 out of 38) had more than seventy percent of the rules showing such a trend.

Beyond sleep patterns, homogeneity was also observed in phone usage patterns. For P18's weekday evening rule No.10, both the user and the identified group had a high number of unlocks per minute when they mostly stayed somewhere far from dorms and social spaces. Their rules have a similar support value but P18's rule had a higher confidence value, which means that this rule was more common for P18 than for the identified group. This suggests that P18 was unlocking their phone more often. Likewise, P72's weekday evening rule No.2 shows that P72 spent a longer time than the identified group interacting with their phones when they remained sedentary at some place outside dorm. Both rules reflect that the two students had more active phone usage than their respective identified groups. These

Table 5.2: Examples of top paired rules that capture behavior differences between a target user with depression and their identified groups without depression. Two rules in a pair have identical $X$ and the same selected feature in $Y$ (shown in bold). Each item is displayed in a "[feature type]feature(discretized value)" manner. The bold feature highlights the difference between the target user and the identified group. The dashed lines group the type of selected features such as sleep-related and screen-related behavior.

| **PID** | **Rule** | $X$ | $Y_{tar}$ | $sup_{tar}$<br>$conf_{tar}$ | $Y_{oppo}$ | $sup_{oppo}$<br>$conf_{oppo}$ | **Property** |
|---|---|---|---|---|---|---|---|
| 18 | Wkend Morning No.4 | - [Campus] Pct. of time at social space or dom (low)<br>- [Sleep] Total bout nums during the sleep (low) | - [Sleep] **Num of bouts being asleep** (medium) | 0.200<br>0.667 | - [Sleep] **Num of bouts being asleep** (low) | 0.152<br>0.298 | More disturbed sleep pattern |
| 72 | Wkdy Allday No.3 | - [Campus] Pct. of time at dorm (low) | - [Sleep] **Duration of being asleep** (low)<br>- [Campus] Pct. of time at sports space (low) | 0.367<br>0.550 | - [Sleep] **Duration of being asleep** (medium)<br>- [Campus] Pct. of time at sports space (low) | 0.172<br>0.246 | More disturbed sleep pattern |
| 18 | Wkdy Evening No.10 | - [Location] Moving time Pct. (low)<br>- [Campus] Pct. of time at social space or dorm (low) | - [Screen] **Num of unlock per minute** (high) | 0.132<br>0.588 | Same as $Y_{tar}$ with lower *conf* | 0.137<br>0.413 | More phone interaction |
| 72 | Wkdy Evening No.2 | - [Location] Pct. of time at home (low)<br>- [Step] Avg duration of sedentary bouts (high) | - [Campus] Pct. of time at dorm space (low)<br>- [Screen] **Avg duration of interaction bouts** (high) | 0.171<br>0.302 | - [Campus] Pct. of time at dorm space (low)<br>- [Screen] **Avg duration of interaction bouts** (low) | 0.161<br>0.395 | More phone interaction |
| 18 | Wkend Allday No.23 | - [Bluetooth] Num of unique others' device (low)<br>- [Campus] Pct. of time at social space or dorm (low) | - [Campus] Pct. of time at greens space (low)<br>- [Campus] **Pct. of time at sport space** (low) | 0.267<br>1.000 | Same as $Y_{tar}$ with lower *conf* | 0.222<br>0.435 | More time at sport space |
| 72 | Wkend Allday No.18 | - [Location] Log of location variance (high)<br>- [Campus] Pct. of time at dorm (low) | - [Campus] **Pct. of time at sport space** (low) | 0.267<br>0.533 | Same as $Y_{tar}$ with higher *sup* and *conf* | 0.356<br>0.744 | Less time at sport space |
| 18 | Wkend Allday No.21 | - [Call] Num of outgoing calls (low) | - [Sleep] Total bout nums during the sleep (low)<br>- [Call] **Num of outgoing calls (medium)** | 0.133<br>0.364 | - [Sleep] Total bout nums during the sleep (low)<br>- [Call] **Num of outgoing calls (low)** | 0.300<br>0.720 | More call communication |
| 72 | Wkend Evening No.7 | - [Call] Num of outgoing calls (low)<br>- [Campus] Pct. of time at dorm (low) | - [Call] **Duration of outgoing calls (low)** | 0.166<br>0.625 | Same as $Y_{tar}$ with lower *sup* and *conf* | 0.158<br>0.441 | Less call communication |

patterns were also observed in other students: **73.7% of the users with depressive symptoms have the majority of the rules about phone usage showing the same trend.** These observations are consistent with the population-level interpretation in Ch. 4 and are supported by relevant findings in psychology and clinical psychiatry [16, 181, 186].

In contrast, some rules capture behavior heterogeneity between P18 and P72. Examples of mobility and communication behavior are shown in the last four rows in Tab. 5.2. P18's weekend all-day rule No.23 suggests that P18 spent a shorter time in sports spaces (for exercise) and green spaces (for relaxation) than the identified group when they were out of dorms and far from large groups of people (indicated by the Bluetooth feature). However, P72 had a rule with the opposite behavior: P72 spent more time in sports spaces than the identified group when they were out of the dorm and had a large location variance. A similar contrast was also observed in phone call behavior. P72 had a shorter duration of outgoing phone calls than that of the identified group (weekend all-day rule No.21), indicating less social communication. In contrast, P18 had more outgoing calls than the identified group under similar contexts. These distinguishing rules reflect individual behavior differences between P18 and P72.

As the interpretation method is designed to mine each individual's behavior rules independently, it can capture the behavior similarity and the difference among users at the same time. Examples in Tab. 5.2 support that my method can generate personalized rules for personalized interpretation.

### *5.3 Summary*

In this chapter, I presented a new method for personalized behavior classification. My method borrowed the idea from memory-based collaborative filtering and used the behavior correlation square to capture the behavior relevance among individuals. Moreover, it combined the relevance metric and association rule mining to obtain personalized behavior rules. The results showed that the method outperforms the state-of-the-art model on depression prediction by 5.1% on the accuracy and 5.5% on the F1 score, with statistical significance. Moreover, the method also generated highly interpretable rules that capture both homogeneity and heterogeneity in students' behavior related to depression.

# Chapter 6
# GENERALIZABLE BEHAVIOR MODELING

Addressing interpretability and personalization on a single dataset is important. However, it is only the prerequisite for building deployable and generalizable modeling techniques. To ensure that a behavior model can work for a larger group of users, its generalizability needs to be verified on multiple datasets from different populations. This leads to our third research question (**RQ3**): *How to generalize a model to other populations/contexts?*

In this chapter, I first point out the challenges of cross-dataset generalization (Ch. 6.1) and the lack of model generalizability in previous research (Ch. 6.2). Then, to answer RQ3, I introduce my new algorithm to improve model generalization by leveraging behavior continuity – one of the human behavior nature properties of daily routines (Ch. 6.3 and Ch. 6.4). Furthermore, I develop a new benchmark platform with 26 behavior models to accelerate the generalizability research for other researchers (Ch. 6.5). Finally, I summarize this chapter in Ch. 6.6.

## *6.1 Challenge of Cross-dataset Generalization*

To quantify the difference among datasets, I first conducted a "Name-The-Dataset" task on our four datasets [184]. For every dataset, I aggregated each feature matrix by calculating the mean along the time dimension to obtain a feature vector for each sample. I then performed an 80%/20% user split for the training/testing set, *i.e.*, no overlapping user's data is in both the training and testing sets simultaneously. I used a portion of the training data to train a small random forest model (10 decision trees, each with a maximum depth of 3) to classify which dataset a data belongs to (*i.e.*, four-class classification). I used SMOTE to mitigate the data imbalance as datasets have different sizes [34].

The left side of Fig. 6.1a indicates the model performance on the testing set. With only one user in the training set (around 0.2% of samples from the training set), the model is able

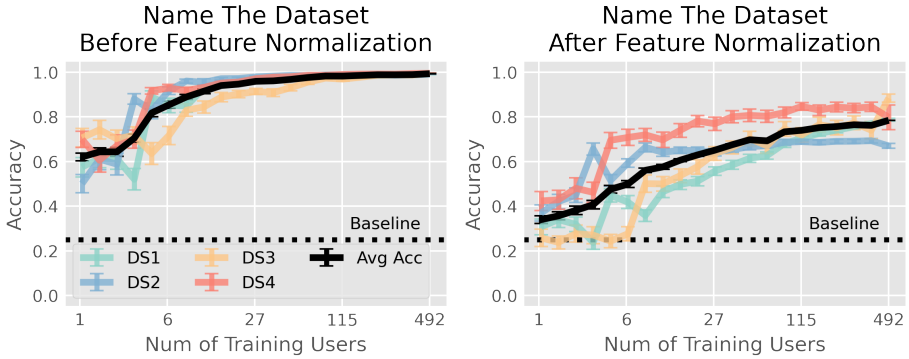
to achieve an average accuracy of 62.0%, compared to 25% as the baseline. With five users (1%) and 50 users (10%) from the training data, the accuracy reaches 81.7% and 96.8%, respectively. These results indicate that the behavior features from different datasets (*i.e.*, populations or years) have different distributions allowing them to be easily differentiated.

Feature normalization is one of the common techniques for mitigating feature value differences and aligning the data. Therefore, I also trained another model with normalized features (subtracting the median and dividing by the 5-95 quantile range). As shown in the right side of Fig. 6.1a, the model still achieves an accuracy of 34.0%, 49.9%, and 67.8% with 1, 5, and 50 users from the training set. The normalization does diminish the distribution shift, but the distinguishable differences between datasets still persist.

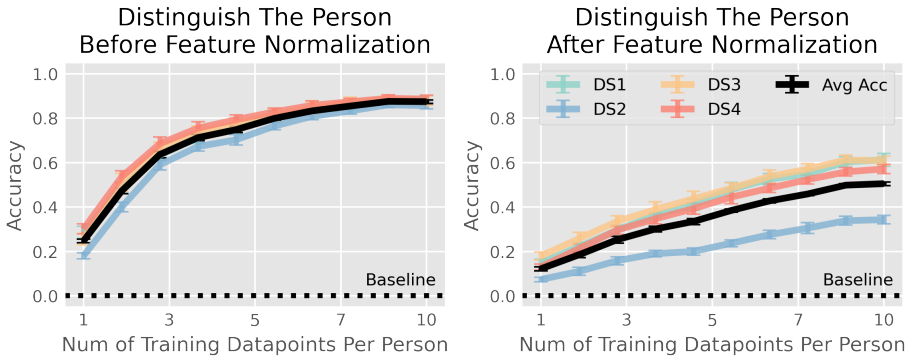
To quantify the differences among individuals, I further conducted a "Distinguish-The-Person" task, replacing the label from the dataset with the user ID. I performed an 80%/20% split on each user's data for the training/testing set. Similar to the "Name-The-Dataset" task, I then used a proportion of the training data to train another random forest model (maximum leaves=2k) to classify which user ID a data point belongs to, which is a challenging 534-class classification task (same as the total number of unique participants in the four datasets).

The two plots in Fig. 6.1b show the performance using the features before and after normalization. Using 1, 5, and 10 data points per user from the training set, the model achieves 24.7%, 74.8%, 87.4% using direct features, and 12.2%, 33.5%, 50.5% using normalized features, which are all significantly higher than the baseline accuracy 0.19% (1/534). I also tested the person-year classification (618 user-years), which showed similar results.

These analysis results clearly show that data from different datasets and individuals all have distinguishing distributions. In the "Distinguish-The-Person" task, the relative advantage over baseline is even larger than the "Name-The-Dataset" task, suggesting that the challenges of domain generalization in longitudinal human behavior modeling may come more from individual differences than dataset or population differences.




(a) "Name-The-Dataset" Task (n=10, max depth=3)




(b) "Distinguish-The-Person" Task (n=10, max leaf num = 2K)

Figure 6.1: Results of Domain Classification Tasks using Simple Random Forest. Four colored lines indicate the accuracy of the four datasets, and the black line indicates the overall accuracy. Error bar indicates the standard error. The same below.

### *6.2 Lack of Generalizability in Previous Research*

In order to inspect how generalizable previous depression prediction algorithms are, I re-implemented nine algorithms with the same or similar features, and evaluated these models on four datasets in Ch. 3. It is worth noting that the re-implementation may not be exactly the same as the prior work due to the lack of open-source code.

1. **Canzian *et al.*** [29]: used location trajectory features directly computed from the past

two-week time window to train a support vector machine (SVM) for depression prediction.

2. **Saeb *et al.*** [156]: used the combination of location and screen features and aggregated their daily average of the past two weeks to train a logistic regression model with elastic regularization.
3. **Farhan *et al.*** [54]: used location and physical activity features from the past two-week window to train an SVM model.
4. **Wahle *et al.*** [195]: used features from several sensors (activity, location, WiFi, screen, and call) over the past two weeks. They used both daily aggregations (*i.e.*, mean, sum, variance) and direct computation of the features of the two weeks to build SVM and Random Forest models. I left out the WiFi and call features to ensure its compatibility with our datasets.
5. **Lu *et al.*** [117]: used location, activity, and sleep features computed from the past two weeks and built multi-task learning models combining linear regression and logistic regression. To further deal with device platform differences, they built one model for iOS devices and one for Android devices.
6. **Wang *et al.*** [201]: used location, screen, activity, sleep, and audio features and aggregated their daily average and slope of the past two weeks (for the frequent prediction) or the whole study period (for the end-of-term prediction). They built a lasso-regularized logistic regression model for the prediction. I excluded audio features as they were not collected in all datasets.
7. **Xu *et al.*- Interpretable** [209]: the method in Ch. 4. It was originally developed and evaluated on only one dataset in Ch. 3.
8. **Xu *et al.*- Personalized** [210]: the method in Ch. 5. It was originally developed and evaluated on only one dataset in Ch. 3.
9. **Chikersal *et al.*** [35]: used a similar set of basic features as *Xu et al.- Interpretable* and calculated more aggregations (breakpoint and slope) across multiple time ranges (daily and biweekly). They first trained a nested randomized logistic regression for feature selection. Then, they trained separate gradient boosting and logistic regression models

using data from every sensor, and combined the prediction with another Adaboost model to generate the final prediction.

Some models focused on end-of-term detection [35,54,156,209,210], while others focused on frequent weekly detection [29,117,195,198]. Thus I evaluated these models on both tasks. To keep the process consistent with the prior work, the models' hyperparameters were tuned via grid search with the same range as mentioned in each prior work. The training and testing used data from the same dataset, using twenty-fold cross-validation at the user level (*i.e.*, leaving 5% of the users out in each fold). For each of the four datasets, I repeated this process and calculated balanced accuracy as the metric. Tab. 6.1 summarizes the results of model performance on each dataset.

For the end-of-term depression prediction, only three models, *Xu et al.- Interpretable*, *Xu et al.- Personalized*, and *Chikersal et al.*, achieve a balanced accuracy over 60%. However, these models' performance is significantly lower than the reported result in the prior work (average $\Delta = 15.9 \pm 10.7\%$). Similar findings are also observed in the repeated weekly depression prediction task. None of these models' performance reaches 60%. Again, there is a big gap between the reported results and the results in our datasets (average $\Delta = 22.6 \pm 8.5\%$). Such findings indicate that prior algorithms do not generalize well on our datasets.

## *6.3 Methods*

The previous two sections show the challenge of cross-dataset generalization. To address this challenge, I propose two new methods that leverage human behavior characteristics from two different perspectives: one using the similarity of behavior trajectories among different individuals (Section 6.3.1), and the other leveraging the continuity of behavior trajectories of each individual (Section 6.3.2).

### *6.3.1 Clustering Similar Behavior Trajectories to Train Independent Models*

This method stems from a simple intuition: Although participants are from different populations, certain periods of some users' behavior trajectories could be similar and clustered

Table 6.1: Balanced Accuracy of Predicting End-of-term or Weekly Depression Status. Models are evaluated by 5-fold cross-validation within each of the four datasets. The "Prior Work" columns show the performance contrast between the reported results in prior literature (evaluated on their own datasets) and the results on our datasets. $^{+}/^{++}$ means the literature only reported F1-score/ROC AUC, which was usually close to balanced accuracy.

| **Model** | **Task1: End-of-Term Depression Detection** | | | | | | **Task2: Weekly Depression Detection** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **DS1** | **DS2** | **DS3** | **DS4** | **Avg** | *Prior Work* | **DS1** | **DS2** | **DS3** | **DS4** | **Avg** | *Prior Work* |
| *Majority Baseline* | 0.500 | 0.500 | 0.500 | 0.500 | 0.500 | - | 0.500 | 0.500 | 0.500 | 0.500 | 0.500 | - |
| Canzian *et al.* [29] | 0.559 | 0.516 | 0.526 | 0.500 | 0.525 | - | 0.512 | 0.497 | 0.512 | 0.500 | 0.505 | *0.760* |
| Saeb *et al.* [156] | 0.539 | 0.508 | 0.562 | 0.480 | 0.522 | *0.791* | 0.496 | 0.549 | 0.508 | 0.506 | 0.515 | - |
| Farhan *et al.* [54] | 0.552 | 0.609 | 0.505 | 0.620 | 0.572 | *0.855* | 0.519 | 0.515 | 0.519 | 0.513 | 0.517 | - |
| Wahle *et al.* [195] | 0.526 | 0.527 | 0.562 | 0.583 | 0.550 | - | 0.514 | 0.530 | 0.519 | 0.503 | 0.516 | *0.616* |
| Lu *et al.* [117] | 0.574 | 0.558 | 0.403 | 0.634 | 0.542 | - | 0.531 | 0.499 | 0.482 | 0.534 | 0.511 | *0.770*$^{+}$ |
| Wang *et al.* [201] | 0.566 | 0.500 | 0.537 | 0.503 | 0.527 | - | 0.534 | 0.500 | 0.512 | 0.500 | 0.512 | *0.809*$^{++}$ |
| Xu *et al.*- Interpretable [209] | 0.722 | 0.623 | 0.815 | 0.706 | 0.716 | *0.806* | 0.533 | 0.576 | 0.677 | 0.553 | 0.585 | - |
| Xu *et al.*- Personalized [210] | 0.723 | 0.699 | 0.818 | 0.649 | 0.722 | *0.819* | 0.568 | 0.562 | 0.614 | 0.572 | 0.579 | - |
| Chikersal *et al.* [35] | 0.728 | 0.776 | 0.795 | 0.698 | 0.749 | *0.816* | 0.615 | 0.613 | 0.595 | 0.551 | 0.593 | - |

together. Data in the same cluster might have closer distributions and simplify the classification task within the cluster.

Specifically, I build an ensemble model based on unsupervised clustering. Using all training data, I first follow [71, 208] to train a deep clustering model with convolutional auto-encoders (DCEC) that assigns data points with cluster indices. For each cluster, I then use data in that cluster to train a small Siamese model [97]. In the inference stage, data is fed into the DCEC model to find the cluster index, and classified by the corresponding Siamese model.

### *6.3.2 Reordering Behavior Trajectory to Force The Model to Learn Behavior Continuity*

The challenge of domain generalization is mainly caused by the data distribution shift in heterogeneous domains. In our case, such a shift comes not only from dataset differences (*i.e.*, each subpopulation behavior pattern varies [43, 126]), but also from individual differences (*i.e.*, each person behaves uniquely [133, 173]). However, despite these differences, there still exists a range of similarities among individuals' behaviors. For example, people tend to have daily routines, which define the structure of and influence of almost every aspect of everyday behaviors [21]. Although individuals have unique routines, these patterns would lead to *continuous* or even repetitive behavior trajectories from day to day [80]. Such an observation motivates me to leverage behavior continuity and construct a self-supervised learning task to obtain generalizable feature representations.

Self-supervised learning is a recently popular learning paradigm that builds self-supervised tasks (*i.e.*, pretext tasks) from unlabeled data. The pretext tasks are often not directly related to the main prediction task. It has been applied to domain generalization problems in CV tasks (*e.g.*, JiGen [31], SelfReg [90]) and NLP tasks (*e.g.*, BERT [45]).

To leverage the continuity of behavior trajectory, inspired by [31], I propose a new multi-task learning model, **Reorder**, with a new pretext task called reordering puzzle (see Fig. 6.2): I shuffle the temporal order of the feature matrix, and train a model to reconstruct the original sequence, jointly optimized with the main classification task over different domains. The model needs to achieve two tasks simultaneously: 1) it will learn to capture the continuity of behavior trajectories, so that it can find the right temporal order of the time-series feature data after shuffling; and 2) it will also learn to solve the main task (*i.e.*, depression prediction in our case). Due to the prevalence of the continuous behavior trajectories based on human nature (analogous to the continuous edges and patterns in images [31]), solving the first task by learning such continuity could assist the model to extract more generalizable representations of behavior trajectories across individuals. This can enable more robust domain generalization in behavior modeling.

Specifically, I create a multi-task learning model function $h$, with the 1D-CNN based embedding (parameters $\theta_f$), fully connected layers for reordering (parameters $\theta_r$), and fully

connected layers for classification (parameters $\theta_c$).

The first task is the main classification task. The loss function of this task is $\mathcal{L}_c(h(x|\theta_f, \theta_c), y)$, where $x$ is the input matrix, and $y$ is the classification label. The second task is the reordering task. I first sliced the feature matrix along the temporal dimension into $n$ segments and then shuffled these segments. I picked the number of segments $n = 10$ ($\lceil 28/3 \rceil$) since 28! or 14! (28/2) is too computationally expensive. Moreover, as 10! total possible permutations is an overly large number, I follow the practice in [31] and predetermine a subset of $P = 200$ permutations by following the Hamming-distance-based method [135]. I then assign an index to each permutation. Within the subset, the reordering task is equivalent to identifying the index of the permutation, which is essentially a classification task. Therefore, the loss function of the reordering task is $\mathcal{L}_r(h(z|\theta_f, \theta_r), p)$, where $z$ is the feature matrix $x$ after the reordering, and $p$ is the permutation index. Overall, the model can be trained via the

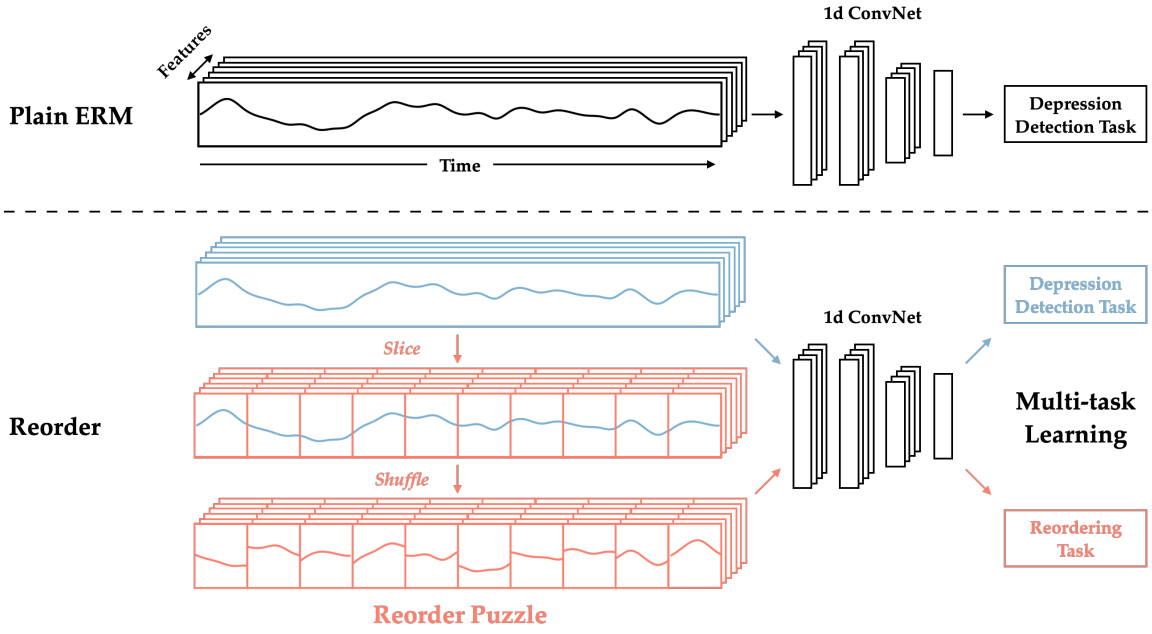



Figure 6.2: The Design of Reorder Compared to ERM. In addition the main behavior modeling task, Reorder further introduces a secondary task of solving a reorder puzzle to force the model to learn the continuity of behavior trajectory.

following objective function:

$$\underset{\theta_f,\theta_c,\theta_r}{\operatorname{argmin}} \sum_{i=1}^{S} \Big( \underbrace{\sum_{j=1}^{N_i} \mathcal{L}_c(h(x_j^i|\theta_f,\theta_c), y_j^i)}_{\text{Loss Func of The Main Task}} + \underbrace{\sum_{j=1}^{\beta N_i} \alpha \mathcal{L}_r(h(z_j^i|\theta_f,\theta_r), p_j^i)}_{\text{Loss Func of The Reordering Task}} \Big)$$

where both $\mathcal{L}_c$ and $\mathcal{L}_r$ are cross-entropy losses. $S$ is the total number of training domains, and $N_i$ is the size of a domain $i$. $\alpha$ is used to control the weight of the reordering task, while $\beta$ is used to control the size of reordering data. $x_j^i, y_j^i, z_j^i, p_j^i$ are specific instances in each domain $i$ with index $j$. Moreover, I also incorporate the Mixup augmentation technique [228] to increase the variation of the data. It is worth noting that the reorder puzzle is only enabled during the training stage. There is no shuffling at the testing stage to avoid extra noise.

## *6.4 Evaluation*

### *6.4.1 Cross-Dataset Evaluation*

Using four datasets (two from UW and two from Dartmouth), I built a leave-one-dataset-out evaluation pipeline [140, 152]. Specifically, each time I took out one dataset as the testing set, and used the three other datasets as the training set. Such a cross-dataset analysis can effectively measure how a model trained on existing datasets could work on a new unseen dataset [109, 184].

#### *6.4.1.1 Baseline Comparison*

Other than the existing nine depression prediction algorithm and my new two algorithms, there are also a range of domain generalization algorithms that can be borrowed from the ML community. I implemented eight well-studied deep learning techniques to cover the major approaches of domain generalization [197], including 1) data manipulation (Mixup [228]), 2) representation learning (IRM [15], DANN [63], CSD [145]), and 3) learning strategy (MLDG [109], MASF [48], Siamese [97]).

1. **ERM** (Empirical Risk Minimization) [191]: the basic model training techniques without particular design for domain generalization. ERM shows competitive performance in

previous CV generalization tasks [69, 197]. I implemented multiple architectures with ERM: a) **ERM-1D-CNN**: one-dimensional CNN that treats the data as a time series of length 28; b) **ERM-2D-CNN**: two-dimensional CNN that treats the data as a one-channel image; c) **ERM-LSTM**: another architecture to model time-series data; d) **ERM-Transformer**: a transformer-based architecture for modeling sequence data.

2. **Mixup** (ERM-Mixup) [228]: a popular data manipulation and augmentation technique that performs a linear interpolation between any two instances with a weight sampled from a Beta distribution. Mixup can be plugged into any model architecture and training pipeline. In this paper, I used 1D-CNN in my experiment as it has a similar performance as 2D-CNN, while being more robust to feature positions in the feature matrix. Similarly, I also used 1D-CNN for the other methods in the rest of this section if they are agnostic of architectures.
3. **IRM** (Invariant Risk Minimization) [15]: a representation learning paradigm to estimate invariant correlations across multiple distributions and learn a data representation such that the optimal classifier can match all training distributions.
4. **DANN** (Domain-Adversarial Neural Network) [63]: another representation learning technique that adversarially trains the generator and discriminator. The discriminator is trained to distinguish different domains, while the generator is trained to fool the discriminator to learn domain-invariant feature representations. For our purposes, I treated each dataset as a domain (**DANN - Dataset as Domain**), or each person as a domain (**DANN - Person as Domain**).
5. **CSD** (Common Specific Decomposition) [145]: a feature disentanglement-based representation learning technique from the multi-component analysis perspective, which extracts the domain-shared and domain-specific features using separate network parameters. Similar to DANN, I also investigated two versions of domain: **CSD - Dataset as Domain**, and **CSD - Person as Domain**.
6. **MLDG** (Meta-Learning for Domain Generalization) [109]: one of the first methods using meta-learning strategy for domain generalization. MLDG splits the data of the training domains into meta-train and meta-test to simulate the domain shift to learn general

features. I again tried **MLDG - Dataset as Domain**, and **MLDG - Person as Domain**.

7. **MASF** (Model-Agnostic Learning of Semantic Features) [48]: a learning strategy that combines meta-learning and feature disentanglement. After simulating domain shift by domain split, MASF further regularizes the semantic structure of the feature space by introducing a global loss (to preserve relationships between classes) and a local loss (to promote domain-independent class clustering). I tested **MASF - Dataset as Domain**, and **MASF - Person as Domain**.
8. **Siamese Network** [97]: a metric-learning based strategy to find a better pair-wise distance metric. It aims to decrease the distance between positive pairs (*i.e.*, same labels) and increase the distance between negative pairs (*i.e.*, different labels).

The input-output format of these models is the same: I picked a subset of important daily features in the most recent traditional depression prediction algorithms [35, 210]. I then used the past-four-week feature matrix as the input, and the depression label as the output.

#### *6.4.1.2 Implementation*

I focused on the weekly depression prediction as the main task, because the small sample size of the end-of-term task makes it infeasible to train deep models, *i.e.*, all of the models (17) other than the prior depression prediction models (9). For the prior depression prediction models, I followed a similar procedure to conduct a hyperparameter tuning by grid search on the three training datasets.

The rest of the methods all used deep models. For methods that used 1D-CNN as the backbone, I used a simple architecture based on a small-range tuning using ERM-1D-CNN: It had three 1D-convolution layers (size 8, stride 3, ReLU activation), each followed by a batch normalization layer, a max-pooling layer, as well as a dropout layer (rate 0.25). A fully connected layer (size 16) was attached after flattening the third convolution layer's output to convert it into a vector of length 16. The following layers were then customized for each model.

For the new method *Reorder*, the feature layer was connected to two fully connected layers (size 16 and 2) for the classification task, and another two layers (size 32 and 200) for the reordering task. Additional model details are listed in codebase's config files. For the new method *Clustering*, I picked the cluster number as 60. The auto-encoder had two 1D-convolution layers (size 64 and 32) with the middle hidden size of 10. The Siamese network in each cluster adopted the same three 1D-convolution layers (size 8) architecture as other models.

Other architectures were also kept simple: 2D-CNN used three 2D-convolution layers with the same size, stride, and activation function as 1D-CNN; LSTM used two bi-directional layers (size 20); Transformer used two transformer blocks, each with 4 self-attention heads (size 4) and a 1D-convolutional feed-forward layer (size 16).

For all models, I used Adam as the optimizer and adopted a cosine annealing schedule to repeatedly decrease the learning rate and then restart with a higher learning rate [116], with an initial learning rate of 0.001, an annealing decay of 0.95, and an annealing step size of 100. I isolated 10% of the training datasets as a validation set. All models are trained with 200 epochs, and the best epoch was picked based on the performance of the train and validation set. In addition to balanced accuracy, I further employed ROC AUC and balanced accuracy as the main evaluation metrics as they indicate the overall results with varying decision boundaries in a detection problem.

#### *6.4.1.3 Results*

Tab. 6.2 lists out the results of all models on each of the four datasets, and Fig. 6.3 presents the barplot of these models' ranked average performance. There are a few noteworthy observations.

First, all nine depression prediction models have worse performance than that of Tab. 6.1. The best model, *Chikersal et al.*, has an average balanced accuracy of 52.0% in the cross-dataset evaluation, compared to 58.8% in the within-dataset evaluation. This performance gap, in addition to the previously described gap between the reported results in prior work and the results on our datasets, indicates that these models do not generalize well across

Table 6.2: Model Performance of Predicting Weekly Depression Status across Datasets. Models are tested on one dataset after being trained on all other datasets. The Adv column indicates the advantage compared to the majority baseline. + or − indicates the algorithm has at least one or no metric better than the baseline, with t-test statistical significance: $p < 0.1_{\bullet}$ (marginal significance), $< 0.05^{*}$, $< 0.01^{**}$, and $< 0.001^{***}$.

| **Model** | **ROC AUC** | | | | | **Balanced Accuracy** | | | | | **Adv** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **DS1** | **DS2** | **DS3** | **DS4** | **Avg** | **DS1** | **DS2** | **DS3** | **DS4** | **Avg** | |
| *Majority Baseline* | 0.500 | 0.500 | 0.500 | 0.500 | 0.500 | 0.500 | 0.500 | 0.500 | 0.500 | 0.500 | |
| Canzian *et al.* [29] | 0.490 | 0.493 | 0.457 | 0.537 | 0.494 | 0.499 | 0.500 | 0.500 | 0.500 | 0.500 | − |
| Saeb *et al.* [156] | 0.516 | 0.491 | 0.554 | 0.504 | 0.516 | 0.504 | 0.499 | 0.510 | 0.510 | 0.506 | + |
| Farhan *et al.* [54] | 0.509 | 0.472 | 0.483 | 0.493 | 0.489 | 0.517 | 0.489 | 0.509 | 0.489 | 0.501 | + |
| Wahle *et al.* [195] | 0.496 | 0.550 | 0.502 | 0.503 | 0.513 | 0.501 | 0.510 | 0.500 | 0.505 | 0.504 | + |
| Lu *et al.* [117] | 0.557 | 0.505 | 0.443 | 0.449 | 0.488 | 0.538 | 0.496 | 0.438 | 0.492 | 0.491 | − |
| Wang *et al.* [201] | 0.536 | 0.539 | 0.500 | 0.500 | 0.519 | 0.501 | 0.502 | 0.500 | 0.500 | 0.501 | + |
| Xu *et al.*- Interpretable [209] | 0.457 | 0.507 | 0.461 | 0.496 | 0.480 | 0.501 | 0.498 | 0.478 | 0.502 | 0.495 | − |
| Xu *et al.*- Personalized [210] | 0.482 | 0.534 | 0.548 | 0.517 | 0.520 | 0.484 | 0.530 | 0.511 | 0.516 | 0.510 | + |
| Chikersal *et al.* [35] | 0.590 | 0.525 | 0.526 | 0.523 | 0.541 | 0.545 | 0.503 | 0.523 | 0.508 | 0.520 | $+^{*}$ |
| ERM - 1D-CNN [191] | 0.511 | 0.530 | 0.512 | 0.511 | 0.516 | 0.503 | 0.510 | 0.522 | 0.520 | 0.514 | $+^{*}$ |
| ERM - 2D-CNN [191] | 0.508 | 0.509 | 0.535 | 0.542 | 0.523 | 0.507 | 0.504 | 0.523 | 0.534 | 0.517 | $+^{*}$ |
| ERM - LSTM [191] | 0.518 | 0.516 | 0.536 | 0.511 | 0.520 | 0.511 | 0.502 | 0.528 | 0.502 | 0.511 | $+^{*}$ |
| ERM - Transformer [191] | 0.508 | 0.464 | 0.527 | 0.486 | 0.496 | 0.513 | 0.474 | 0.510 | 0.488 | 0.496 | − |
| ERM - Mixup [228] | 0.512 | 0.522 | 0.513 | 0.520 | 0.517 | 0.517 | 0.505 | 0.519 | 0.513 | 0.514 | $+^{***}$ |
| IRM [15] | 0.546 | 0.514 | 0.518 | 0.507 | 0.521 | 0.532 | 0.513 | 0.524 | 0.510 | 0.520 | $+^{**}$ |
| DANN - Dataset as Domain [64] | 0.501 | 0.510 | 0.465 | 0.506 | 0.496 | 0.499 | 0.500 | 0.500 | 0.500 | 0.500 | − |
| DANN - Person as Domain [64] | 0.512 | 0.511 | 0.470 | 0.507 | 0.500 | 0.500 | 0.500 | 0.500 | 0.500 | 0.500 | − |
| CSD - Dataset as Domain [145] | 0.519 | 0.530 | 0.516 | 0.504 | 0.517 | 0.517 | 0.530 | 0.504 | 0.494 | 0.511 | $+^{*}$ |
| CSD - Person as Domain [145] | 0.510 | 0.521 | 0.524 | 0.536 | 0.523 | 0.500 | 0.513 | 0.523 | 0.522 | 0.514 | $+^{**}$ |
| MLDG - Dataset as Domain [109] | 0.495 | 0.475 | 0.519 | 0.496 | 0.496 | 0.501 | 0.470 | 0.523 | 0.501 | 0.499 | − |
| MLDG - Person as Domain [109] | 0.509 | 0.539 | 0.512 | 0.488 | 0.512 | 0.499 | 0.522 | 0.498 | 0.483 | 0.501 | + |
| MASF - Dataset as Domain [48] | 0.505 | 0.514 | 0.506 | 0.522 | 0.512 | 0.496 | 0.499 | 0.509 | 0.499 | 0.501 | $+^{*}$ |
| MASF - Person as Domain [48] | 0.508 | 0.502 | 0.485 | 0.523 | 0.504 | 0.507 | 0.507 | 0.489 | 0.498 | 0.500 | + |
| Siamese Network [97] | 0.512 | 0.509 | 0.488 | 0.508 | 0.504 | 0.512 | 0.509 | 0.488 | 0.508 | 0.504 | + |
| Clustering | 0.522 | 0.502 | 0.497 | 0.521 | 0.511 | 0.518 | 0.505 | 0.499 | 0.517 | 0.510 | $+_{\bullet}$ |
| **Reorder** | **0.584** | **0.588** | **0.580** | **0.548** | **0.575** | **0.570** | **0.546** | **0.558** | **0.535** | **0.552** | $+^{***}$ |

datasets. Since all of the evaluation experiments use the same features, the first explanation of missing other sensors' features does not play a role at this stage. Therefore, the gap

between within-dataset and cross-dataset evaluation is mainly caused by the distribution shift among different populations.

Second, modern ML techniques have been developed to deal with the challenge of feature shift across domains. However, these models barely work on our datasets. Among the 15 models I investigated, *CSD - Person as Domain* and *ERM - 2D-CNN* achieve the highest ROC AUC (52.3%). The results are similar to the results of traditional depression prediction models (the best model *Chikersal et al.* achieves a ROC AUC of 54.1%). These evaluation outcomes show that recent domain generalization methods do not work well on our datasets. This can be explained by the fact that most methods were developed for CV or NLP tasks. Their generalizability may be affected when applied to longitudinal behavior data. Another interesting finding is the good performance of the naive ERM-based methods. Most of them (except *ERM-Transformer*) rank top 10 among the total of 26 methods, outperforming many generalization methods such as *DANN* and *MLDG*. Such a finding is consistent with the results in DomainBed [69] and DeepDG [197]. Both pointed out that the ERM baseline often has competitive generalization performance.

Finally and most importantly, among all 26 models, my newly proposed *Reorder* model achieves the highest ROC AUC of 57.5% and the highest balanced accuracy of 55.2%. As shown in Fig. 6.3, *Reorder* stands out among all methods. It outperforms the other models by at least 3.4% on ROC AUC (6.3% relative advantage), and 3.2% on absolute balanced accuracy (6.2% relative advantage), both with statistical significance ($p < 0.05$). Since *Reorder* has the same 1D-CNN backbone as *ERM-1D-CNN*, the comparison between these two models reveals the effect of adding the second reorder puzzle-solving task, which boosts the performance by 5.9% on ROC AUC (11.4% relative advantage) and 3.9% on balanced accuracy (7.6% relative advantage). Such an improvement illustrates that learning the temporal continuity of behavior trajectory can enhance the model's generalizability. However, although *Reorder* shows positive signals on domain generalization, it is worth noting that the model still has great room for improvement. A ROC AUC of 57.5% is still far from being deployable in real-life scenarios, and more future research to improve model generalizability is needed in the community.

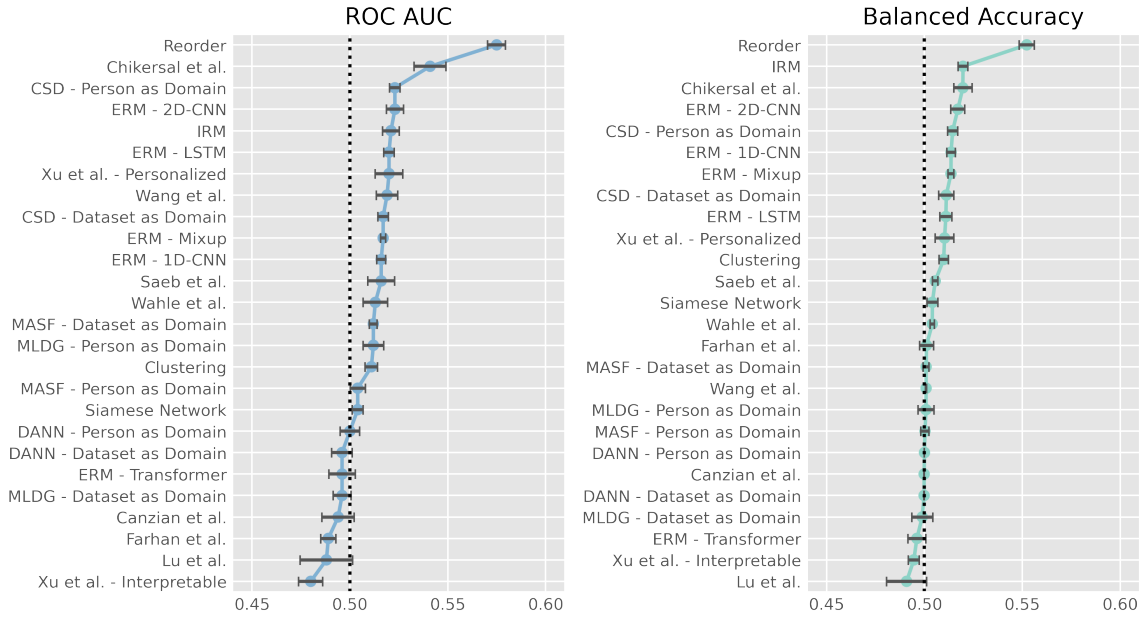



Figure 6.3: Model Performance of Predicting Depression across Datasets. The dashed line indicates naive majority baseline.

### *6.4.2 Cross-Institute and Cross-Year Generalization Analysis*

In addition to the leave-one-dataset-out evaluation, I conducted additional experiments to obtain more insights into the models' generalizability and investigate different generalization challenges. As the four datasets were collected from two institutes across two years, I can evaluate how these models generalize across institutes (*i.e.*, different populations), and across years (*i.e.*, different users within the same population). Moreover, in each institute, there were a small number of people who participated in both years. Thus, I also evaluated the models on these subsets of users across years to test generalization across the same participants at different times.

#### *6.4.2.1 Cross-Institute*

With datasets from Ch. 3, I used the two datasets from one institute as the training set, and the two datasets from the other institute as the testing set. The models' training setup

was the same as the previous section. The left side of Fig. 6.4 presents the ranked average balanced accuracy of all methods (train/test on institute 1/2 and then on institute 2/1). Overall, the performance is not as good compared to Fig. 6.3. *Xu et al.- Personalized* has the best performance, but it only achieves a ROC AUC of 53.1%, followed by my method *Reorder* (52.4%). These lower performance results reflect that cross-institute generalization is a more challenging task.

#### *6.4.2.2 Cross-Year*

Similarly, I used two datasets from one year as the training set, and the rest from the other year as the testing set. The training setup was kept the same. The middle portion of Fig. 6.4 shows the average balanced accuracy (train/test on year 1/2 and then on year 2/1). Compared to the cross-institute evaluation, the results of the cross-year evaluation are slightly better. The method *Reorder* has the best performance, with a ROC AUC of 54.2%. This indicates that cross-year generalization could be easier than cross-institute generalization. Moreover, some models have interesting contrasting results. For example, *Clustering* ranks among the top 10 in the cross-institute evaluation, while it ranks among the bottom 5 in the cross-year evaluation ($\Delta = 2.1\%$). In contrast, *Xu et al.- Interpretable* has the worst performance in the leave-one-dataset-out evaluation, but ranks No.4 and No.9 in the cross-institute and cross-year evaluation ($\Delta = 4.0\%$ and 3.9%). This means that these models capture different aspects of domain generalization.

#### *6.4.2.3 Cross-Year with Overlapping People*

I further narrowed down the evaluation to the overlapping people across years (there are no overlapping users across institutes), *i.e.*, I used trained a model on overlapping users in one dataset, and tested the model on these users in the other dataset, with the same training details and dataset setup as Section 6.4.2.2. The right side of Fig. 6.4 shows the average balanced accuracy. A strong increase in performance was observed. *Reorder* again achieves the best performance with an ROC AUC of 61.6%, which is 7.4% higher than the best result in Section 6.4.2.2. The advantage could be explained by the fact that the same

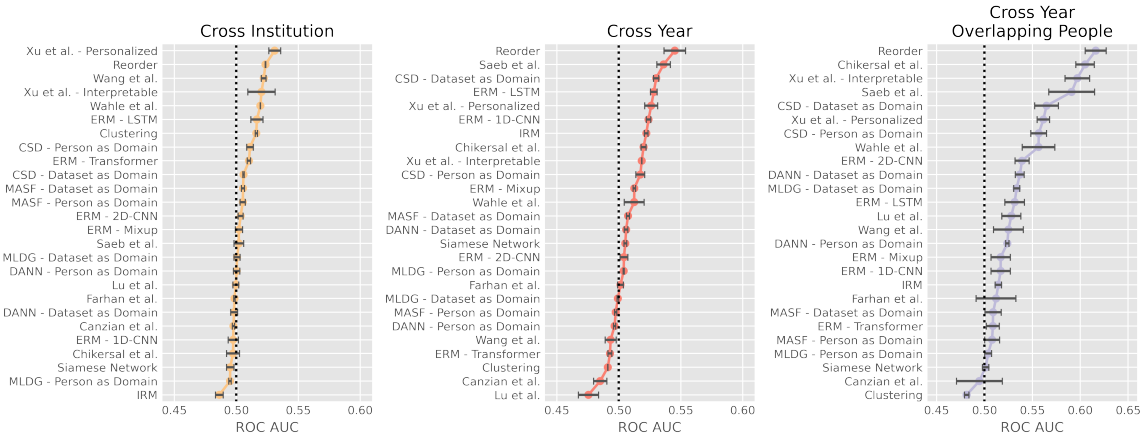



Figure 6.4: Model Performance of Predicting Depression across Institutions and Years. Models are tested on the datasets of one year/institution after being trained on the other year/institution.

users' behavior trajectories could preserve some patterns across years. In addition, both *Xu et al.- Interpretable* and *Xu et al.- Personalized* have good performance in the two cross-year evaluation, which is in line with their reported analysis [209,210]. Moreover, *Chikersal et al.* ranks the top 5 among two of the four different setups. I speculate that the comprehensive feature aggregation and selection pipeline proposed in [35] may identify more generalizable features.

Overall, the cross-institute and cross-year evaluation results further illustrate more insights into model generalizability. My model *Reorder* has the best or the second best results across the different tasks, revealing its advantages over other models. Moreover, the results of the third cross-dataset setup (*i.e.*, different times of the same users) are clearly better than those of the other two setups, which reveals that the individual differences (no matter whether that is within or between populations) may play the most important role in the cross-dataset generalization challenge.

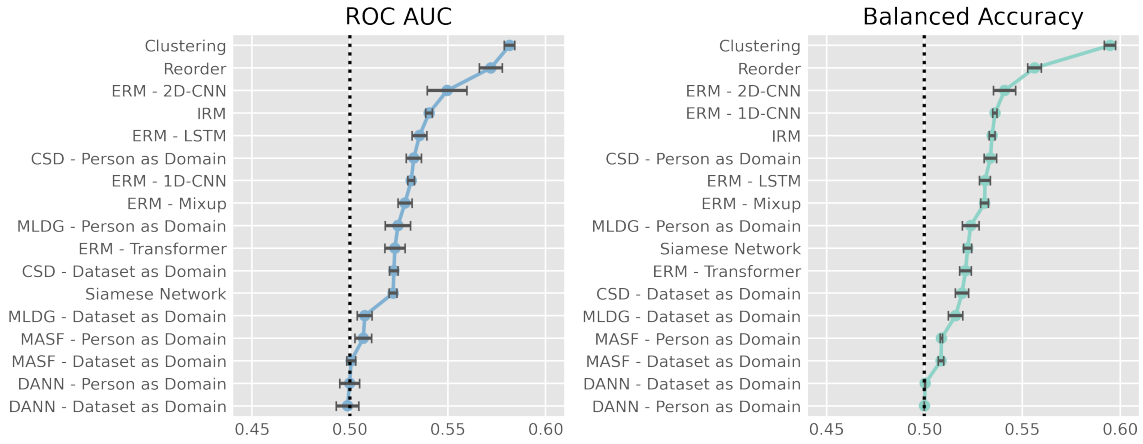



Figure 6.5: Model Performance across Datasets with Information Leakage. Models are tested on one dataset after being trained on all other datasets with the optimal epoch numbers.

### *6.4.3 Optimal Early Stopping Analysis*

One of the major obstacles of domain generalization is the overfitting onto the training set [197, 233]. A similar overfitting issue during the deep models' training process is also observed. For example, during the training, *Reorder* achieves an average training ROC AUC of 74.9% and a training balanced accuracy of 67.7% for the main task, as well as an average training accuracy of 30.5% for the reordering task (as a 200-class classification task). These results of the main task are much better than the ones on the testing set (note that there is no reordering task on the testing set.). As the training epoch increases, there is a generally increasing performance curve on the testing set (learning), and then a decreasing trend (overfitting). If the "optimal" training epoch for each model can be found via early stopping, the results could then reflect the "upper bound" of these models' performance.

I conducted such an experiment with a similar leave-one-dataset-out setup as Section 6.4.1. Note that this experiment was only applicable to deep-learning-based algorithms as their training process has multiple epochs. Specifically, for every model, I iterated through the training epoch from 1 to 200. I performed the same epoch selection at each

epoch number based on the best performance on the train and validation set. I then compared the test performance across the 200 epochs and identified the best epoch, assuming the training could be stopped at this epoch. Note that this step involved a little information leakage as it leveraged the testing set to select the epoch. Thus the results only reflect the theoretical upper-bound performance. When there is a large validation dataset in the future, the method could potentially achieve similar results as on the test set, so that information leakage will no longer be necessary.

Fig. 6.5 summarizes the generalizability of the deep-learning models to a new data set, when training is halted at the optimal training epoch. This figure shows that even under optimal conditions, current algorithms do not generalize well. In addition, looking at *improvement over standard training*. *Clustering* achieves the best performance, with an average ROC AUC of 58.2% and an average balanced accuracy of 59.5%. This is a large benefit from optimal stopping ($\Delta$ equals 7.1% and 8.5%, respectively), which reveals that *Clustering* has an overfitting problem. *Reorder* achieves the second-best performance, but its performance improvement is minor ($\Delta < 0.5\%$). This indicates that there is minimal overfitting in *Reorder*.

### *6.5 GLOBEM Platform*

To gain a fair and reliable performance of these algorithms, I built an open-source benchmark platform, **GLOBEM** (short for **G**eneralization of **LO**ngitudinal **BE**havior **M**odeling), to incorporate all algorithms mentioned above. Compared to the existing platforms DomainBed [69] and DeepDG [197] that mainly aim for image-based domain generalization tasks, GLOBEM specifically focuses on longitudinal passive sensing data.

Fig. 6.6 illustrates the overall structure of the GLOBEM platform. It splits the whole pipeline into three independent modules:

1. The feature preparation module defines behavior features used by the algorithm;
2. The model computation module defines how a behavior model is going to be trained. These two modules are determined by the core algorithm;
3. The configuration module provides the flexibility to adjust hyperparameters.

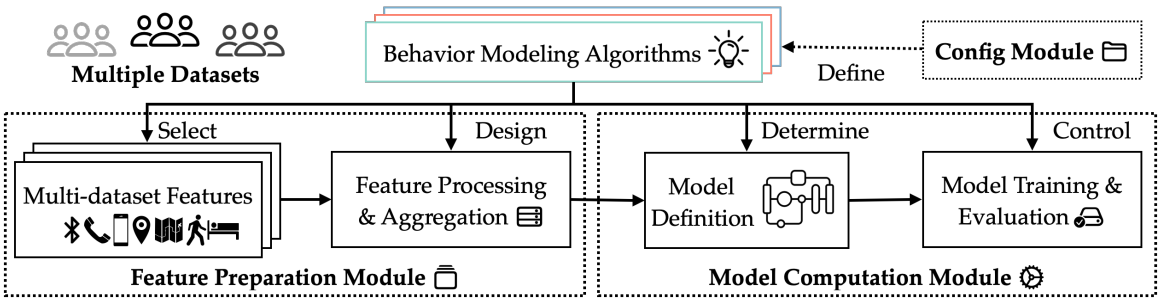



Figure 6.6: Design of The Benchmark Platform GLOBEM. It modularizes the pipeline and supports flexible adjustment of the existing algorithms and easy development of new algorithms.

Researchers and developers can re-use any of these modules to develop new algorithms within the pipeline. Moreover, GLOBEM separates the config setup from the model definition, supporting easy testing and ablation studies of hyperparameters and different features.

## *6.6 Summary*

In this chapter, I highlighted the importance of a behavior model's cross-dataset generalizability. Using depression prediction as an example, I took the first step towards a systematic cross-dataset generalization evaluation in the longitudinal behavior modeling domain. I re-implemented nine prior depression prediction methods, built eight recent domain generalization algorithms, and proposed two new methods for better generalizability. The evaluation of these methods on our datasets demonstrated that existing algorithms barely outperform the baseline on cross-dataset generalization tasks, and that my new method *Reorder* could learn the continuity of behavior trajectories and achieve better generalizability across datasets. Although statistically significant, its performance advantage is marginal, leaving great room for improvement. I integrated all methods and open-sourced a benchmark platform named GLOBEM to assist future researchers in testing existing methods and developing new algorithms. I envision this step as a necessary and essential part of behavior model deployment in our research field.

# Chapter 7

# JUST-IN-TIME BEHAVIOR INTERVENTION

From Ch. 4 to Ch. 6, I have introduced my new algorithms to address three important deployability challenges for behavior modeling: interpretability, personalization, and generalizability. However, building better models is only the first step towards an intelligent system. In order to create a complete loop between users and the system (Fig. 1.1), we need to bring the model back to users. In order to develop better intervention techniques that focus on influencing user behavior and improving their well-being, we need to answer the fourth research question (**RQ4**): *How to leverage the insights of behavior models to develop better intervention techniques?*

In this chapter, I will introduce a new just-in-time (JIT) intervention technique, *TypeOut*, that builds on top of the insights from the personalized behavior interpretation (Ch. 4 and Ch. 5). Specifically, the behavior pattern rules obtained from the population level (Ch. 4.2.2.1) and the individual level (Ch. 5.2.2.2) both indicate that smartphone overuse has a strong association with depression. Although the causal relationship between the two is complicated [49,65], I start with smartphone overuse reduction as a starting point. Future work will be extended to depression intervention after we can robustly control the safety and ethical risks.

## *7.1 Design*

I first give a brief overview of the existing smartphone overuse intervention techniques (Ch. 7.1.1). Then, I will introduce the theoretical foundation (Ch. 7.1.2), which guides the design of my new intervention technique *TypeOut* (Ch. 7.1.3).

| **Intervention Content** | **Just-in-Time Intervention Mechanism** | | |
|---|---|---|---|
| | Blocking | Notification | Typing |
| Non-semantic | — | — | ▲ Random text [94] (**TypingOnly**) |
| Semantic, non-self-affirmation | Rule [120], Goal setting [93] | Passive information [137, 165], Social/context awareness [91, 95], Goal setting [79] | Passive information, Goal setting |
| Semantic, self-affirmation | — | ▲ Self-affirmation notification (**ContentOnly**) | ★ **Self-affirmation typing (TypeOut)** |

Table 7.1: Summary of Prior or Potential Work on JIT Intervention for Smartphone Overuse. ★ indicates our intervention technique TypeOut. Two ▲s indicate the baselines we compare against, one in the same row as TypeOut and the other in the same column. – means not applicable.

### *7.1.1 Smartphone Overuse and Intervention*

There is a growing body of research revealing an increasing population of smartphone overuse caused by constant connectivity (*e.g.*, [76,77,105,119,187]). It may lead to a range of negative consequences such as distraction [50,119], lack of sleep [105], family conflicts [187], anxiety [76], and, as I have pointed out in previous chapters, depression.

Researchers have built a large number of interventions from various perspectives to reduce smartphone overuse [28,119]. The intervention mechanisms of most of these existing techniques can be categorized into two types: blocking users' apps/phones [14, 93, 120], or sending notifications and reminders [79, 91, 95, 137, 165]. Blocking access to smartphones can be effective [92, 93, 120], but may be overly restrictive, creating a bad user experience and even triggering greater usage [41, 94]. Notifications and reminders are the choices of intervention for many previous studies [95, 137, 165]. Some also engaged users in pre-establishing rules or goals [79]. However, these methods did not have a mechanism to encourage users to engage with the intervention content. Researchers found that users

could easily ignore these notifications since they can be readily dismissed [93]. Thus, a new intervention technique that can balance restrictiveness and engagement is needed. Tab. 7.1 summarizes prior work and my new designs (together with the baseline that I compared against).

### *7.1.2 Dual Process and Self-Affirmation*

To design an effective intervention, we need to first understand how users make the decision to engage in smartphone use or non-use. The Dual Process Theory [82] contends that human behavior is controlled by two processes or "systems": System 1, an impulsive process that represents spontaneous, automatic, and non-conscious influences on behavior, and System 2, a deliberative or reflective process which represents rational, deliberative, and conscious decision-making influences [72, 175]. Researchers can explain the failure of well-intended behavior control with this theory: the self-regulation from good intentions (System 2) is usually overridden by momentary impulses (System 1) [112, 119, 147]. In the smartphone overuse scenario, the easy access to rich information and immediate gratification from using smartphones drives users' impulses [106, 203]. Therefore, persuasive technologies usually aim to awaken System 2 and increase its strength, which is mediated by the expected value of control [166], so that System 2 can lead users' behavior [119]. There are 3 factors influencing the expected value of control, including the reward/punishment people perceive they could obtain, the expectancy or likelihood that people would be able to achieve a desired outcome, and the delay before the outcome [119]. These factors illuminate the direction of our smartphone overuse intervention design to effectively strengthen the control of System 2, thus achieving behavior regulation and reducing phone usage.

Self-affirmation is the act of bolstering or restoring a perception of oneself as being adequate [174]. The central assumption of Self-Affirmation Theory [174] is that people are strongly motivated to protect their sense of adaptive and moral adequacy, or "self-integrity" [51, 167]. Self-affirmation methods, such as thinking about core personal values, important personal strengths, or valued social relations, can offset the threats to self-integrity [124]. Moreover, researchers find that the cognitive processes instigated by self-

affirmation can help to better trigger System 2 in the Dual Process Theory [143,190]. Prior studies have shown that self-affirmation is effective in a wide range of behavior change intervention domains, such as improving academic performance [38, 168], reducing stereotyping towards minority group members [18, 56], and promoting health behavior change [52, 53]. Some cognitive behavior therapy techniques employed self-affirmation exercises to reduce smartphone overuse [224, 225]. A typical self-affirmation task usually focuses on a specific value or positive personal characteristics. The specific task can vary, such as responding to specific scales, writing a list or an essay, or using imagery techniques on their positive qualities [124]. Recently, researchers have adapted traditional time-consuming self-affirmation exercises to be short, regularly delivered questionnaires for healthy eating behaviors that are more compatible with the smartphone platform [171]. There is a growing call for JIT intervention techniques that can better engage users at the right moment and that are integrated with self-affirmation content [121].

Therefore, I combine the JIT typing process (leveraging the Dual Process theory) and the self-affirmation theory to develop a novel intervention technique for smartphone overuse reduction.

### *7.1.3 Intervention Design*

I focus on addressing three questions to design an effective intervention. First, *when* should an intervention be triggered? Second, *how* should the intervention be presented? Third, and most importantly, *what* content should the intervention include? To answer these questions, I led a group of students and investigated them from multiple perspectives. Our design follows the Dual Process [82] and Expected Value of Control theories [166] as introduced in Ch. 7.1.2 when answering the three questions.

#### *7.1.3.1 When to intervene?*

A large body of prior work has adopted the JIT approach for smartphone overuse intervention. As overuse naturally occurs when users are using their phones, an intervention is usually introduced during these periods. There are a few options to determine the triggering

moment, such as the moment when users are opening an app [94, 120], or when the usage duration for an app reaches an upper limit defined by users [79, 93]. As a starting point, we choose to trigger a JIT intervention when a target app is being launched. We envision this method is compatible with other JIT designs and plan to explore more JIT options in the future.

#### *7.1.3.2 How to intervene?*

Most of the previous intervention techniques either present passive notifications/reminders that can be ignored by users [93], or introduce coercive prohibition that can cause reversed effect [92]. Recently, researchers proposed a typing-based unlock process (*i.e.*, users need to follow an instruction to type specific content before accessing the app) to balance the effectiveness and the restrictiveness [94]. Our method adopts this mechanism.

On the one hand, typing words following an instruction could enhance users' engagement, as they have to read the text and then type it out. Compared to notifications that can be dismissed easily, typing requires more attention, engagement, and involvement from users [131]. On the other hand, typing does not strictly prevent users from using an app. It introduces additional interaction costs when accessing the app, but leaves users with the option to continue using the phone if they want to. Compared to more coercive prohibition methods, type-to-unlock is more flexible. Meanwhile, the additional interaction cost when entering the app introduces a notable gulf of execution on gratification seeking [42]. Such a micro-boundary can possibly switch a user's mind from System 1 to System 2 (as defined in the Dual Process Theory) for self-reflection/judgment [94].

More importantly, such a typing process provides an opportunity to carefully design the typing content delivered to users. This offers an avenue to take user engagement one step further, leading to the next design question: what intervention content should be presented to users?

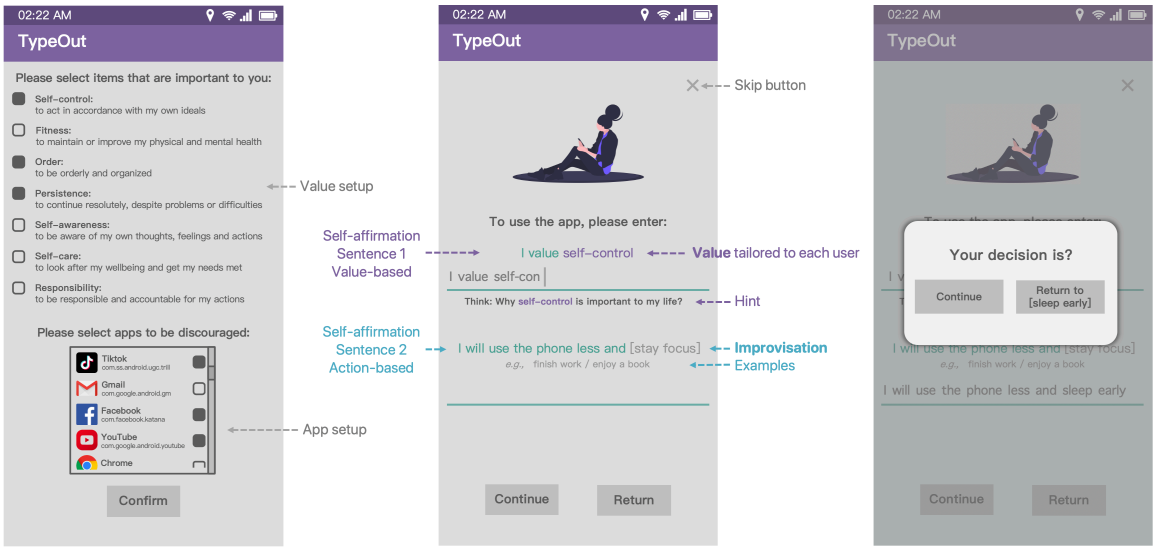



Figure 7.1: Intervention Design of TypeOut. Users set up their individual values list and select apps for which they want to receive an intervention (left). When an intervention appears, users can leave the app at any stage by clicking the return button or system home button, *e.g.*, before or during typing process (middle), or at the confirmation stage after typing is complete (right). Otherwise, users can enter the app after typing the self-affirmation and clicking the Continue button.

#### *7.1.3.3 What to be delivered as intervention content?*

As the typing process will better engage users with the intervention content, the content can go beyond presenting non-semantic content [94] or objective information (*e.g.*, the duration of app usage [79]), and be more thought-provoking (System 2) to improve its effectiveness. Leveraging Self-Affirmation Theory [174], we propose a content design that integrates value-based self-affirmation and JIT improvisation. It can stimulate users to reflect on their own core personal values and connect these with the smartphone overuse behavior, thus motivating users to change their current behavior to protect their self-integrity [132]. Fig. 7.1 presents our intervention design.

**Value-based Self-Affirmation.** Self-affirmation exercises have been employed by a wide range of behavior change interventions, mostly in a traditional way, such as answering

surveys or writing an essay [52,178,183]. The main idea of self-affirmation-based intervention is to leverage users' intrinsic motivation for protecting their self-integrity to regulate their behavior (so that their adequacy is not violated). Our design adopts value-based self-affirmation, one of the most common self-affirmation exercises [51, 124]. Since a given value exists as a long-term belief for users, our design can be customized to each user based on their own set of values.

We employ a value list that is commonly used in acceptance and commitment therapy (ACT) [78], which contains a list of 58 common value items [75]. To narrow down the list and filter out the ones unrelated to smartphone usage, we invited three experts to independently select no more than 20 related items. Moreover, we also delivered an online survey to ask end-users to select items from the long list that they perceive are related to smartphone overuse (N=98). We triangulated the results and found a consistent set of top seven values from both experts and end-users: *Self-control, Fitness, Order, Persistence, Self-awareness, Self-care,* and *Responsibility*. Based on the list of values, we adopt the common practices in affirmation [5, 159, 205] and propose a few short sentence templates that instantiate a value-based self-affirmation exercise, such as "I value $X$", "$X$ is important to me" ($X$ indicates a specific value tailored to each individual). Tab. 7.2 summarizes our templates. Each new user initializes their own list by performing a self-affirmation writing exercise and picking the value items from the list they think are important to themselves (see the left of Fig. 7.1). After this initial setup, when an intervention is triggered, one value item will be randomly sampled from a user's personal list and inserted into the template. Moreover, we also present a hint to encourage users to reflect on the value.

**Just-in-Time Improvisation.** In addition to the sentence that emphasizes value, we also follow affirmation practices and design a second brief sentence template that states the specific actions to reduce overuse. Examples include "I can put down the phone", "I can let go of the app". Moreover, we also append a JIT improvisation at the end of the sentence to encourage users' engagement and stimulate more reflection. Self-improvisation is also encouraged during regular self-affirmation exercises [167, 174]. At the moment when the intervention is introduced, users are asked to come up with a short phrase (no less than two words) about what they can do *if* they reduce overuse, such as "sleep early", "get focused",

| Templates of Sentence 1 - Value | Templates of Sentence 2 - Action |
|---|---|
| I { value, cherish } **X** | I can put down the phone { to, and } [**improvisation**] |
| **X** is { important, crucial, meaningful } to me | I can use the { phone, app } less { to, and } [**improvisation**] |
| I { think, believe } **X** is { important, crucial, meaningful } | I can { leave, quit } the app { to, and } [**improvisation**] |
| I { think, believe } I am a **X** person | I can lock the screen { to, and } [**improvisation**] |

Table 7.2: Templates for the two sentences delivered to users for the JIT self-affirmation typing exercise. Words in the brackets are picked randomly. "X" indicates a specific value (or its adjective form when appropriate), and "improvisation" indicates the just-in-time affirmation content created by users.

"finish my work". Concatenating the first half of the specific overuse-reducing actions and the second half of the improvised activities, an example sentence is: "I can put down the phone and finish my work". The templates of the second short sentence are summarized in Tab. 7.2. When users go through the content, finish typing, and click the Continue button, a confirmation dialogue box will pop up asking for users' decision on whether to access the app, in which the Return button's text is replaced by users' improvisation (see the right of Fig. 7.1). During the intervention, users can leave the app at any stage by clicking the return button: 1) before the typing process, 2) after typing some words, or 3) at the confirmation stage when they finish typing.

It is worth noting that users do not always accept or follow an intervention. When they decide not to follow the intervention, such a fact can become a challenge to their belief and sometimes leads to the *backfire effect*: instead of changing their behavior to be consistent with their belief, users would alter their belief and strengthen their original behavior (*i.e.*, phone overuse) [136, 161]. Our personalized value item list and self-improvisation allow users to customize the content themselves, leading to a better consistency between their beliefs and behavior. Moreover, to further reduce the likelihood of the intervention backfiring (and resulting in negative experience, increased app usage frequency or duration), we intentionally frame these sentences in a neutral tone [207], and unbind the value sentence

and the action sentence [179]. Specifically, we avoid using verbs that may cause pressure or cognition distortion (*e.g.*, "should", "need" statements) [153, 176]. We also do not add any conjunction (*e.g.*, "so", "and", "thus") between the first and the second sentence, and break them into two separate lines [179]. We do this so that if users cannot achieve the target behavior (*e.g.*, continue to use the app), the neutral tone introduces less threat to users' self-integrity, and the unbinding can loosen the connection between their personal value and their current behavior, thus reducing the likelihood of a backfire.

## *7.2 Evaluation*

Combining the three parts in this section, we hypothesize that the integration of a typing-based unlock process and self-affirmation-based content can effectively reduce smartphone overuse than each component itself. We verify our hypothesis via a field experiment in this section. Ch. 7.2.1 gives an overview of the implementation of TypeOut and two baseline methods. Ch. 7.2.2 introduces the deployment study in the wild. Ch. 7.2.3 summarizes the results of the evaluation experiment.

### *7.2.1 Implementation*

#### *7.2.1.1 TypeOut*

We built a mobile application on Android system to instantiate our TypeOut design. We then conducted a one-week pilot field study with five authors of this paper and finalized the design of the application. After the initial self-affirmation exercise, users pick items from the value list that they think are important to themselves. Then, users can select the apps (*i.e.*, target apps) for which they want to receive an intervention. The left of Fig. 7.1 presents the initial setup interface.

We employed the AWARE Framework [57] to detect the screen status and foreground application activities. A typing-based intervention with generated content (as described in Ch. 7.1.3.2 and Ch. 7.1.3.3) will be triggered when one of the target apps is launched. To avoid text auto-completion during typing, we disable any smart typing function during the intervention. Users can press the Return button or system Home button to leave the app at

any stage, or finish typing and continue to use the app. Sometimes, users may have urgent needs to use a target app (*e.g.*, replying to messages). In these cases, users can press a skip button on the right-top corner of the interface to bypass the intervention. To prevent overly frequent intervention, when users enter a target app via typing or pressing the skip button, the intervention for this target app will not be triggered in the next five minutes.

#### *7.2.1.2 Baselines*

We hypothesize that the integration of the two components – the typing process and the self-affirmation content – can effectively reduce phone overuse. To test this hypothesis, we compare TypeOut against two baseline techniques that separate the two components, as shown in Fig. 7.2.

The first baseline only has the self-affirmation content but not the typing process, namely *ContentOnly*. When an intervention is triggered, it displays a pop-up window with the same content as TypeOut. The difference is that users do not need to type to unlock the app (see Fig. 7.2 left). This is similar to a common notification or reminder-based intervention technique [79, 95, 120].

In contrast, the second baseline only has the typing process but not the self-affirmation content, namely *TypingOnly*. When an intervention is triggered, it introduces a JIT typing process similar to TypeOut. However, instead of typing self-affirmation-based content, it presents random numerals that contain no specific meaning (see Fig. 7.2 right). This is a variant of a recent intervention technique LocknType [94]. LocknType uses digits (0-9) while our baseline uses the digits spelled out (one to nine) to maintain a more consistent comparison against TypeOut. Moreover, we set the total character length of numerals close to but shorter than that of TypeOut's content, because the non-semantic contents would slow down the typing. We used eight to ten numeral words based on a pilot study with five users so that the total typing time is similar.

It is worth noting that we did not choose typing non-self-affirmation content as the baseline to keep the baseline consistent with the recent work LocknType [94], as our main purpose is to evaluate the advantage of self-affirmation-based content over the prior work,

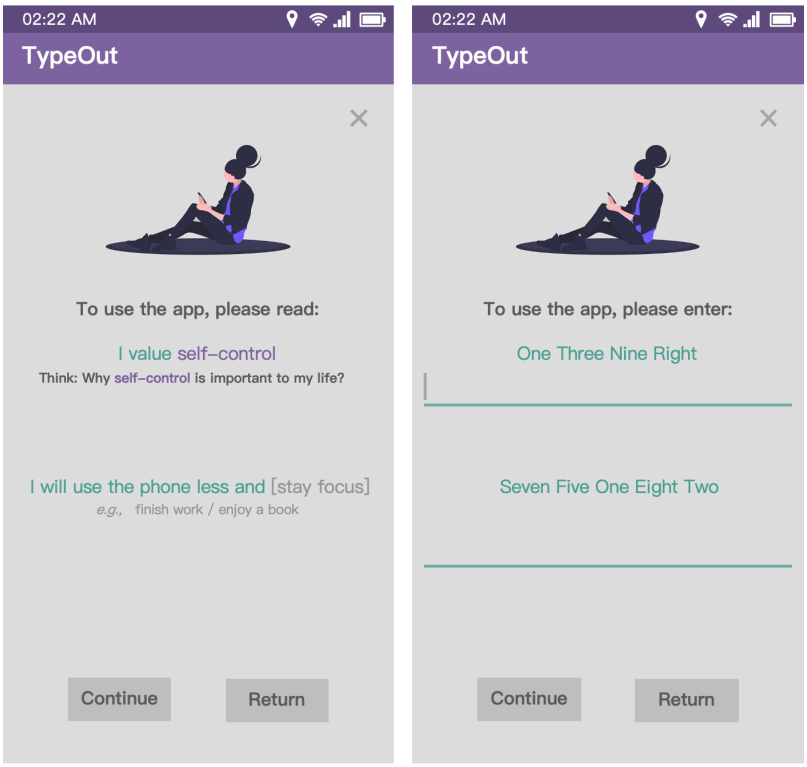



Figure 7.2: Two Baseline Methods to Compare against TypeOut: Content-Only (left) and Typing-Only (Right). The Content-Only baseline has the same self-affirmation content as TypeOut but not the tying process. The Typing-Only baseline has the same typing process as TypeOut but using random numerals as the typing content.

not to show it is the best. Moreover, the design space of the non-self-affirmation content lacks an established theory like value-based self-affirmation and can be overly large, which is hard to control.

### *7.2.2 Deployment*

#### *7.2.2.1 Experiment Design*

We adopt a within-subject design with the intervention techniques as the main independent variable: TypeOut, ContentOnly, and TypingOnly. Users use each intervention technique for two weeks. We counter-balance the order of the intervention to reduce the order effect.

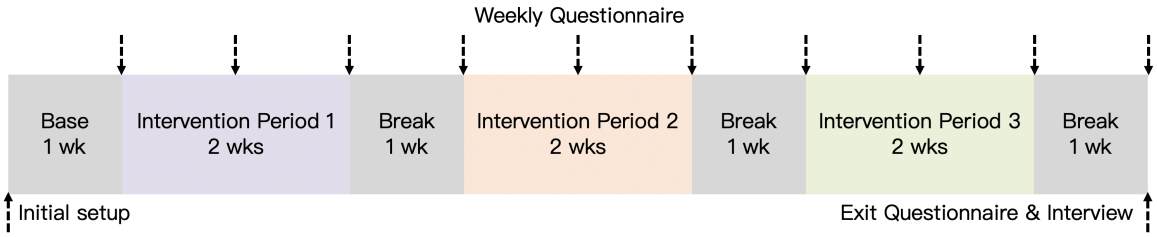



Figure 7.3: The Design of the 10 week Field Experiment. The order of the three intervention techniques is counter-balanced. We insert the break week after each technique to observe the last effect of each technique, and further reduce its influence on the next technique.

The first week of the experiment is used for the base measurement and does not have any intervention. Moreover, we add a one-week break after each technique with two purposes: 1) We can measure whether there is any lasting effect (within that break week) when the intervention is removed, *i.e.*, whether users relapse or self-regulate; 2) The break week can serve as a grace period to further reduce the influence of the previous intervention technique on the next one. The total length of the study is 10 weeks (4 base/break weeks + 3 interventions $\times$ 2 intervention weeks each).

Our dependent variables include the intervention acceptance rate (when the users accept the intervention and leave the target app), the usage duration, and the frequency of all applications. These variables are logged by our mobile app, stored locally on users' phones, and uploaded to our server automatically once the phone is connected to WiFi. In addition to the objective measurement, we deliver the Smartphone Addiction Scale (SAS) [102] to users at the end of each week to collect subjective feedback. Moreover, the final week's questionnaire also asks users to rank the three techniques based on effectiveness. The experiment ends with a brief exit interview. Fig. 7.3 presents the overall design of the experiment. Our experiment was approved by the university institutional review boards (IRB).

#### *7.2.2.2 Participants*

We recruited participants from our local community via sending fliers on social platforms (Wechat and Tencent QQ, the two most widely used platforms in the local community). We used a screening questionnaire (SAS plus a question about the subjective motivation for their current smartphone usage) to collect basic demographics (gender, age, occupation) and filter out users that either did not have a degree of smartphone addiction (SAS $< 99.0$) [102] or were not willing to reduce smartphone overuse to focus on the target users [79, 96]. We received 123 responses in total. None of the participants had experience using any digital or non-digital smartphone overuse intervention. 56 subjected were filtered with a SAS score lower than the threshold, and two subjects were filtered due to a lack of motivation. We invited 65 participants for the experiment. 5 of them chose to quit during the first two weeks of the study, and 3 of them left between week 3 and week 5. No more participants left the study after week 5. We also removed 3 users who did not follow the requirement but skipped most of the interventions. Finally, 54 of them completed the experiment (Female = 25, Male = 29, Age = 22.1 $\pm$ 5.5). 24 participants were college students, while the rest were working professionals. At the beginning of the study, all participants had a moderate to severe smartphone addiction (SAS = 119.0 $\pm$ 20.5).

#### *7.2.2.3 Procedure*

We hosted a 30-minute on-boarding session before the start of the experiment, during which participants familiarized themselves with the study procedure, signed the consent form, installed our application, completed a 10-minute value-based self-affirmation writing exercise, and set up their own value list and target apps accordingly (see Ch. 7.1.3.3). Due to the pandemic, the onboarding session was virtual. We had six groups (permutations of ordering the three intervention techniques) and randomly assigned participants to one group. Then, participants used the different techniques for 10 weeks, following the procedure shown in Fig. 7.3. By the end of the experiment, we conducted a brief semi-structured interview with participants (20 to 30 minutes) and asked for their comments on the different intervention techniques. Participants were compensated with up to $100 based on the number

of questionnaires they completed and the number of days they uploaded data.

### *7.2.3 Results*

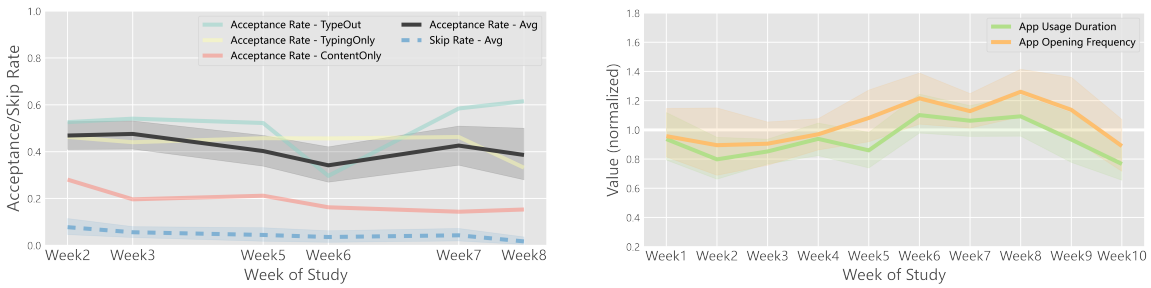
Over the 10 weeks, we collected 358,138 app opening events, 1,358,064 minutes of app usage duration, and 30,754 intervention encounters (9,837, 11,052, and 9,865 for TypeOut, ContentOnly, and TypingOnly, respectively). We analyze the quantitative data and the qualitative data collected via questionnaires and interviews. We start by analyzing the basic compliance and workload of different intervention techniques in Ch. 7.2.3.1. We then investigate users' intervention acceptance rate (Ch. 7.2.3.2), app usage behavior (Ch. 7.2.3.3), as well as a subjective measure towards these techniques (Ch. 7.2.3.4).

#### *7.2.3.1 Compliance and Workload*

We first investigate users' compliance during the 10-week period. We then examined the completion time, number of typing attempts (for TypeOut and TypingOnly), and perceived task workload to understand the interaction cost of each technique.

**Study Compliance.** Fig. 7.4 suggests that participants' behavior fluctuated during the experiment. Therefore, we incorporate order as a main effect in all the following analyses. As for skipping interventions, participants were instructed to skip only when necessary in the onboarding session. The blue line in Fig. 7.4a shows the skip rate during the intervention weeks. The low skip rate indicates that participants did follow our instructions.

**Completion Time.** Overall, ContentOnly took the shortest time (Mean=2.9±1.9s), while TypeOut and TypingOnly took similar time (Mean=10.8±6.1s and Mean=13.3±7.6s) for participants to complete the typing. The average character length was 51.93±2.88 for TypeOut, which was longer than that of TypingOnly (38.68±1.33). This supports our design choice in Ch. 7.2.1.2 on shorter content for TypingOnly to balance typing time. Fig. 7.5 shows boxplots of the time distribution around the median. A Shapiro–Wilk normality test showed that the completion time did not follow a normal distribution. Thus we used a Generalized Linear Mixed Model (*i.e.*, GLMM. For each model, the link function was chosen from Gaussian, Log-Gaussian, Gamma, and Log-Gamma, based on Kolmogorov–




(a) Intervention Acceptance Rate.

(b) App Usage Pattern

Figure 7.4: The Overall Study Compliance over The 10-week Field-Experiment. The shadowed area indicates standard deviation across participants.

Smirnov testing on the distribution of the outcome variables. Participant ID is controlled as a random effect. For simplicity, we do not repeat this description for the rest of the analysis in this section.) for the statistical analysis [206]. We compared the completion time with *Techniques* as the only main factor ($\chi^2(2) = 225.9, p < 0.001$). In a pairwise post-hoc Tukey's HSD test, we found that TypeOut and TypingOnly required similar times ($p = 0.11$), both more than that of ContentOnly. To comprehensively compare TypeOut and TypingOnly, we ran another GLMM on the data of the two typing *Techniques*, with the *Order* of techniques, their interaction (*Order* × *Techniques*), average *Typing Length*, and *Skip Rate* as additional factors. The results indicate that these two intervention techniques introduced a similar temporal cost, as they did not show significance for any factors ($p_{technique} = 0.79, p_{order} = 0.45, p_{technique \times order} = 0.20, p_{typing\ length} = 0.87, p_{skip\ rate} = 0.12$).

**Number of Typing Attempts.** We also measured the number of typing attempts during the intervention for TypeOut and TypingOnly (skipped encounters were excluded as they did not involve typing). A number of 1 meant that participants completed the typing task on the first trial. Higher numbers indicate more input errors, which could aggravate the perceived workload from both the input and time perspectives. On average, participants tried similar times: 1.2±0.4 for TypeOut and 1.1±0.2 for TypingOnly (see the middle of Fig. 7.5). We ran a GLMM on the number of attempts, with *Technique*,

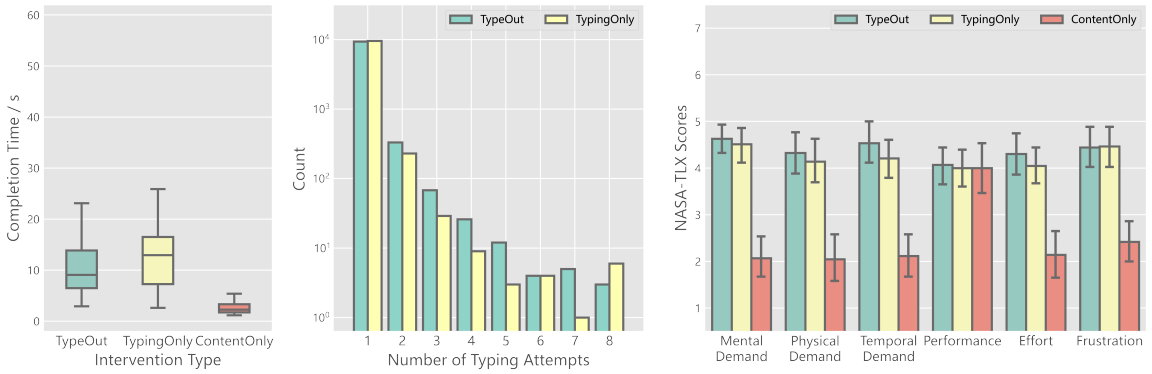



Figure 7.5: Workload Comparison among The Three Intervention Techniques. (Left) Intervention completion time (Middle) Number of typing attempts for TypeOut and TypingOnly in log-scale. (Right) Perceived workload measured via NASA TLX.

*Order*, *Technique* × *Order*, and *Typing Length* as factors. The results indicate that the two techniques had similar input costs, as they did not show any significant difference between the two techniques ($p_{technique} = 0.16, p_{order} = 0.22, p_{technique \times order} = 0.70, p_{typing\ length} = 0.19$).

**Perceived Workload.** We also investigated participants' perceived workload via a NASA TLX assessment (see the right of Fig. 7.5). We compared all three techniques on the six elements of the TLX using a non-parametric ANOVA based on the Aligned Rank Transform and found a significant difference in the techniques ($F(2) = 134.4, p < 0.001$). Post-hoc Wilcoxon signed-rank tests with a Bonferroni correction showed that ContentOnly required significantly lower demand, effort, and frustration, while there was no significant difference between TypeOut and TypingOnly.

In summary, our measure of the workload of the three techniques showed that ContentOnly has the lowest workload, which is not surprising as it only required a single button click to exit the intervention. The two techniques with the typing process introduced higher but similar interaction costs. In the rest of the section, we analyze the effectiveness of each technique in impacting app usage.

#### *7.2.3.2 Intervention Acceptance Rate*

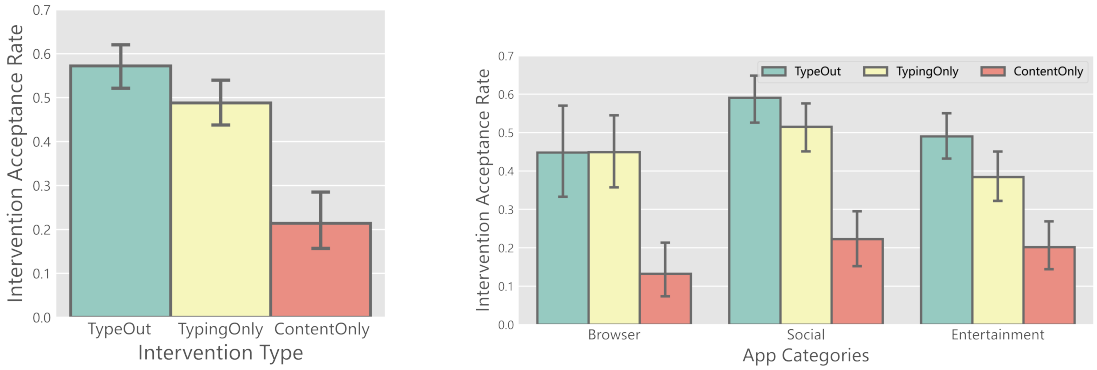
One of the direct indicators of the effectiveness of an intervention is how many times the intervention successfully discourages users from using the target apps. We defined *acceptance rate* as the proportion of times when participants encountered an intervention (the denominator), and decided not to enter the app (the numerator). In general, our results showed that TypeOut achieved a higher acceptance rate.

**Intervention Acceptance Rate.** We first investigated the overall intervention acceptance rate across all apps, and observed that TypeOut (Mean=57.2±28.5%) had a higher acceptance rate than TypingOnly (Mean=48.8±28.8%) and ContentOnly (Mean=21.3±21.2%), as shown in Fig. 7.6a. Our method outperformed the baselines by at least 8.4% on the absolute acceptance rate.

We compared the acceptance rate using a GLMM that included intervention *Technique*, *Order*, *App Category*, *Technique* × *Order*, and *Technique* × *App Category* as factors. Note that typing length was excluded as ContentOnly did not involve typing, the same below. The results showed significance for *Technique* ($\chi^2(2) = 127.1, p < 0.001$) *App Category* ($\chi^2(2) = 12.0, p < 0.01$), and *Order* ($\chi^2(2) = 10.2, p < 0.01$), but no interaction effects ($p_{technique \times order} = 0.12, p_{technique \times app\ category} = 0.71$). A post-hoc Tukey's HSD test on *Technique* showed that TypeOut achieved a higher acceptance rate than TypingOnly ($Z = 13.2, p < 0.001$) and ContentOnly ($Z = 2.4, p < 0.05$).

A post-hoc Tukey's HSD test on *App Category* showed that browser apps had a significantly lower acceptance rate compared to social apps ($Z = 3.2, p < 0.01$) or entertainment apps ($Z = 2.4, p < 0.05$). Fig. 7.6b showed the acceptance rate of different app categories, which indicates that TypeOut outperformed TypingOnly mainly on entertainment apps ($p < 0.05$). Although we observe a difference on social platform apps, the results did not indicate significance ($p = 0.32$). A post-hoc Tukey's HSD test on *Order* showed that the acceptance rate of the first intervention period is higher than the second ($Z = 2.4, p < 0.05$), but other pairs (the first *vs.* the third, second *vs.* the third) did not show a significant difference, as indicated by the black line in Fig. 7.4a.

**Leaving Stage Upon Acceptance.** As introduced in Ch. 7.1.3.3, during the in-




(a) Overall Acceptance Rate

(b) Acceptance Rate Broken-down by App Categories

Figure 7.6: Average Intervention Acceptance Rate of The Three Intervention Techniques.

tervention of TypeOut, participants could leave the app at different stages. When using TypeOut, about 12.5% of participants left after typing something while this number was around 8.7% for TypingOnly. Participants' longer stay suggested deeper participation in the typing content. More specifically, 85.4±16.8% TypeOut participants left before typing, 12.5±14.6% left after typing a few words, and 2.1±6.2% left at the confirmation stage (after typing is completed, see the right of Fig. 7.1). For TypingOnly, these numbers were 91.2±11.5%, 8.7±11.5%, and 0.1±0.2%, respectively. We ran a GLMM on the ratio of people leaving at each stage, with *Technique*, *Order*, *Leaving Stage*, *Technique* × *Order*, and *Technique* × *Leaving Stage* as the factors. The results showed significant difference on *Leaving Stage* ($\chi^2(2) = 2876.8, p < 0.001$) and an interaction effect of *Technique* × *Leaving Stage* ($\chi^2(2) = 7.3, p < 0.05$), but not others ($p_{technique} = 0.66, p_{order} = 0.82, p_{technique \times order} = 0.71$). A post-hoc Tukey's HSD test on the interaction showed that participants had deeper engagement in the self-affirmation content. TypeOut had a marginally higher leaving rate during the typing stage ($Z = 3.6, p = 0.06$) and a significantly higher leaving rate during the confirmation stage ($Z = 8.06, p < 0.01$).

**User Behavior after Accepting Interventions.** We further looked into participants' behavior after acceptance, *i.e.*, the immediate behavior right after users decided to

leave the app after encountering the intervention. We measured three post-intervention behavior [94]: 1) turning off the screen, 2) using another target app, and 3) using another non-target app. Results show that participants using all three interventions were most likely to turn off the screen among the three situations, but TypeOut participants were more likely to do so. When using TypeOut, 48.2±13.0% of the time, participants would turn off the screen, compared to 42.3±15.3% for ContentOnly, and 47.5±10.2% for TypingOnly. Moreover, participants had a lower rate of going to another target app when using TypeOut (25.4±12.3%, similar to ContentOnly 25.0±9.9%) than when using TypingOnly (32.1±14.8%). As for non-target apps, the three techniques had similar percentages (26.4±11.6%, 27.4±8.5%, 25.6±12.1% for TypeOut, TypingOnly, and ContentOnly, respectively). We ran a GLMM on the post-intervention behavior ratio, with *Technique*, *Order*, the post-intervention *Behavior Type*, *Technique* × *Order*, and *Technique* × *Behavior Type* as the factors. The results showed significance for *Behavior Type* ($\chi^2(2) = 4.1, p < 0.05$) and a marginal interaction effect *Technique* × *Behavior Type* ($\chi^2(4) = 5.8, p = 0.06$), but not others ($p_{technique} = 0.19, p_{order} = 0.78, p_{technique \times order} = 0.27$).

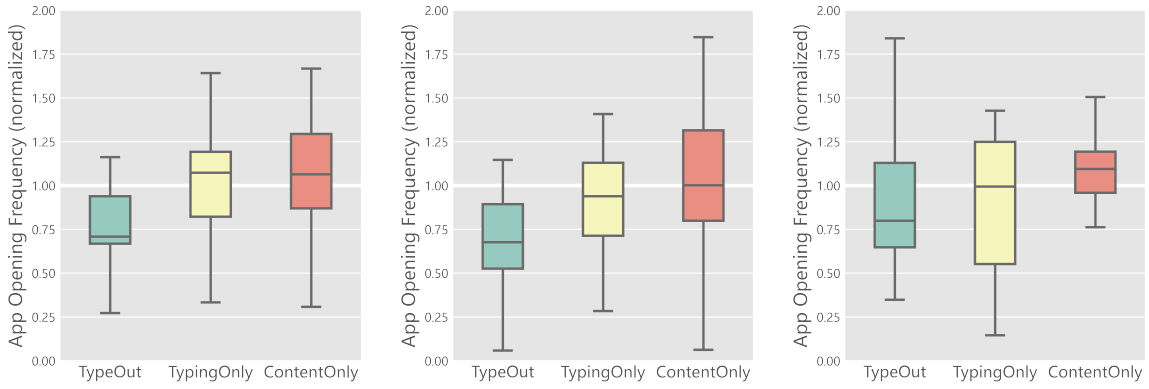
#### *7.2.3.3 App Usage Behavior*

We then investigated the influence of the intervention on participants' overall app usage behavior. Due to the large app usage variation among individuals, we normalized each participant's data by calculating the ratio against their own data during the base week. A ratio smaller or greater than 1 indicated that participants reduced or increased app usage compared to their ordinary behavior. Overall, participants had a smaller ratio when using TypeOut compared to other intervention weeks.

**App Opening Frequency.** We counted the number of app opening attempts for both target apps and non-target apps. It is worth noting that the opening counts included *any* attempt to open the app, regardless of users' final decision on whether to continue accessing the app after encountering an intervention. Such a counting method could emphasize the *overall* effect of an intervention instead of its *in-situ* effect (which was already reflected in the intervention acceptance rate results in Ch. 7.2.3.2). A lower value would suggest that par-

ticipants initiate fewer app openings. Fig. 7.7 presents the relative opening frequency of all apps (Fig. 7.7a), target apps (Fig. 7.7b), and non-target apps (Fig. 7.7c) during the periods of using the three techniques. In general, participants had the lowest app opening frequency during the weeks of TypeOut (Mean=73.2±26.8% compared to the base week), followed by TypingOnly (Mean=99.8±35.7%), and then ContentOnly (Mean=106.5±40.2%). Although TypingOnly and ContentOnly have the potential to discourage app usage when participants receive that intervention (on the overall acceptance rate metric), participants still maintained a similar app opening frequency as their ordinary behavior without any interventions. We ran a GLMM comparing the opening frequency on all apps. As indicated by Fig. 7.4b, we include *Technique*, *Order*, and *Technique* × *Order* in the model. The results showed significance on *Technique* ($\chi^2(2) = 22.4, p < 0.001$), but not on *Order* ($p = 0.87$) or their interaction ($p = 0.14$). A post-hoc Tukey's HSD test on *Technique* showed that participants had significantly lower app opening frequency during the TypeOut weeks than during the two baseline periods ($Z_{ContentOnly} = 4.0, p < 0.001$ and $Z_{TypingOnly} = 4.3, p < 0.001$), while the ContentOnly-TypingOnly pair did not show a significant difference ($p = 0.99$). We found similar results for another GLMM with the same setup but on the opening frequency on target apps ($\chi^2(2) = 15.5, p < 0.001$, $Z_{ContentOnly} = 3.4, p < 0.01$, and $Z_{TypingOnly} = 3.5, p < 0.01$). As for non-target apps, a GLMM did not indicate significance on all factors ($p_{technique} = 0.11, p_{order} = 0.19, p_{technique \times order} = 0.27$).

**App Usage Duration.** In addition to app opening frequency, we also measured app usage duration as it is another important indicator for phone overuse. Similar to Fig. 7.7, Fig. 7.8 presents the relative usage duration of all apps (Fig. 7.8a), target apps (Fig. 7.8b), and non-target apps (Fig. 7.8c). Participants had the lowest app usage duration during the weeks of TypeOut (Mean=74.6±31.0% compared to the base week), followed by TypingOnly (Mean=90.6±27.7%), and ContentOnly (Mean=99.1±28.1%), which is the same order as the results for app opening frequency. When using ContentOnly and TypingOnly, participants still maintained over 90% app usage duration compared to that of the base week. TypeOut can reduce app usage duration more than two baselines. We ran a GLMM with the same setup as those in app usage frequency on all apps' usage duration. The results showed significance for *Technique* ($\chi^2(2) = 12.1, p < 0.01$), but not others ($p_{order} = 0.24, p_{technique \times order} =$

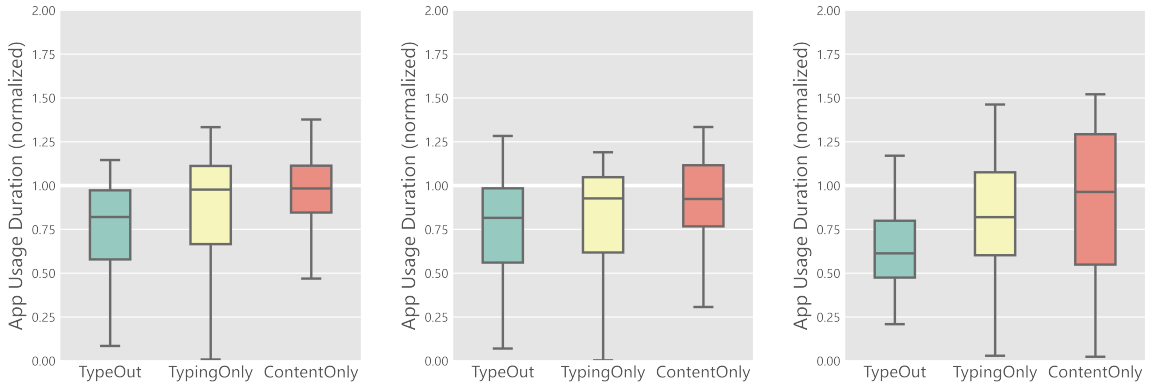



(a) Freq of All Apps (b) Freq of Target Apps (c) Freq of Non-Target Apps

Figure 7.7: App Opening Frequency with Three Intervention Techniques. Each participant's data are normalized by calculating the ratio between the frequency of intervention weeks and that of baseline weeks. It is worth noting that the frequency includes *any* attempt to open the app, regardless of the final decision after intervention, thus a lower frequency suggests users initiate less app opening.

0.27). A post-hoc Tukey's HSD test on *Technique* showed that participants had significantly lower app usage duration during the TypeOut weeks ($Z_{ContentOnly} = 3.4, p < 0.01$ and $Z_{TypingOnly} = 2.7, p < 0.05$). Another GLMM on target apps' data showed similar results with significance for *Technique* ($\chi^2(2) = 6.1, p < 0.05$). A post-hoc Tukey's HSD test found significance between TypeOut *vs.* ContentOnly ($Z = 2.5, p < 0.05$). As for non-target apps, a GLMM on non-target apps' data showed significance for *Technique* ($\chi^2(2) = 6.5, p < 0.05$). A post-hoc Tukey's HSD test found significance between TypeOut *vs.* ContentOnly ($Z = 2.3, p < 0.05$), and marginal significance between TypeOut *vs.* TypingOnly ($Z = 2.1, p = 0.08$).

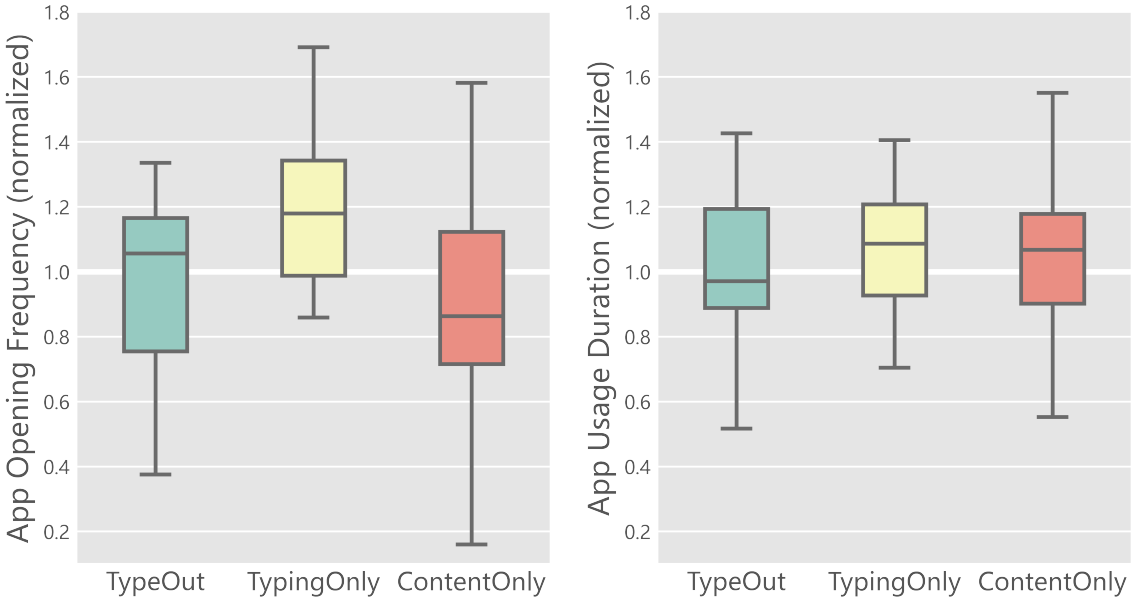
**Lasting Effect on App Usage.** We used the data during break weeks to measure the lasting effect when the intervention was removed. We calculated the app usage ratio between the break weeks after intervention techniques against the base week. A ratio lower than 1 indicates that users reduced smartphone usage compared to their ordinary behavior.




(a) Dur. of All Apps (b) Dur. of Target Apps (c) Dur. of Non-Target Apps

Figure 7.8: App Usage Duration with Three Intervention Techniques. Similar to Fig. 7.7, data are normalized by calculating the ratio between the duration of intervention weeks and that of baseline weeks. A lower ratio indicates less app usage duration.

Fig. 7.9 presents the results of opening frequency and usage duration for all apps. Both the frequency and duration during the break weeks were similar among the three interventions. TypeOut had a slightly lower app usage duration and ContentOnly had a slightly lower app opening frequency. The ratios of both app opening frequency and app usage duration are not significantly different from 1. Specifically, for app opening frequency, we ran a GLMM with *Technique* of the previous intervention period, *Order*, and *Technique* × *Order* as factors. The results showed significance for *Technique* ($\chi^2(2) = 11.1, p < 0.01$) and *Order* ($\chi^2(2) = 6.5, p < 0.05$), but not their interaction ($p = 0.53$). A post-hoc Tukey's HSD test on *Technique* found that both the opening frequency of TypeOut ($Z = 2.7, p < 0.05$) and ContentOnly ($Z = 2.8, p < 0.05$) were significantly lower than that of TypingOnly, but that of TypeOut and ContentOnly were similar ($p = 0.92$). A post-hoc Tukey's HSD test on *Order* showed significance between the first and the third intervention period ($Z = 2.5, p < 0.05$). As for app usage duration, another GLMM with the same setup only showed significance on *Order* ($\chi^2(2) = 18.23, p < 0.001$). A post-hoc Tukey's HSD test on *Order* showed that the usage duration of the third period was significantly lower than that of the




(a) Freq of All Apps

(b) Duration of All Apps

Figure 7.9: App Usage Frequency and Duration of the break weeks after each intervention technique. Data are normalized in the same way as Figure 7.7 and Figure 7.8.

first ($Z = 2.8, p < 0.05$) and the second period ($Z = 4.2, p < 0.001$), as indicated by the lines in Fig. 7.4b. These results indicated that after using them for two weeks, these techniques did not have a strong lasting effect after the intervention was removed.

#### *7.2.3.4 Subjective Measure*

The weekly questionnaires and summative interviews also provided insights on the effectiveness of the three techniques. We employed Affinity diagramming [162] to analyze the interview data. Two researchers independently made notes based on the recording of interviews and collaboratively analyzed and categorized the data with several iterations. Overall, our technique showed better acceptance and user experience than the baselines.

**Smartphone Addiction Scale Scores.** Similar to app usage behavior, we also normalized each participant's SAS scores by calculating the ratio against their own scores of

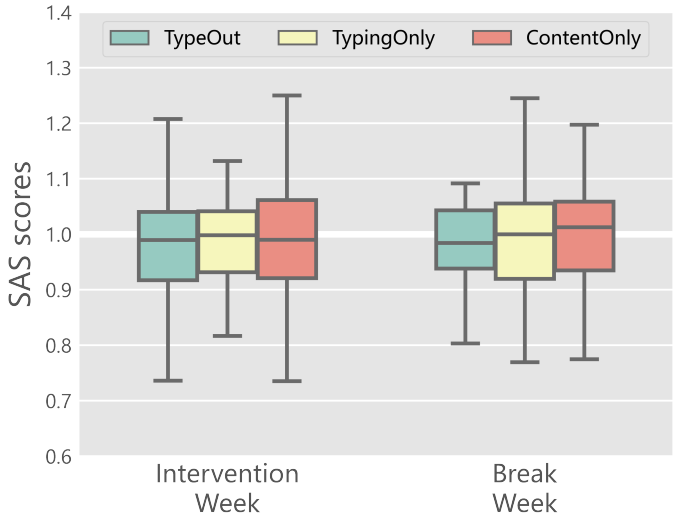



Figure 7.10: Smartphone Addiction Scale Score during/after using the three intervention techniques. Since each intervention technique period had two weeks, SAS scores during the intervention weeks are the average of the two weekly questionnaires.

the base week. A ratio lower than 1 indicated that users had less smartphone addiction. Fig. 7.10 shows the results of the SAS scores during the intervention weeks (average score of the two weekly questionnaires) and the following break week. We found that TypeOut has the lowest SAS scores during the intervention weeks and the following week. For each period, we ran a GLMM on the SAS scores, with *Technique*, *Order*, and *Technique* × *Order* as factors. The two GLMMs did not show a significant difference among the three techniques ($\chi^2_{Intervention\ Week}(2) = 0.6, p = 0.73$, $\chi^2_{Following\ Week}(2) = 1.2, p = 0.54$), nor other factors (Intervention Week: $p_{order} = 0.13, p_{technique \times order} = 0.29$, Following Week: $p_{order} = 0.36, p_{technique \times order} = 0.90$).

**User Reactions.** Our interviews helped us to better understand participants' user experience when using the three techniques. Participants could easily ignore the content of ContentOnly. *"Sometimes I completely skip reading the content during the [ContentOnly] weeks, because I just need to click the continue button"* (P17). Compared to TypingOnly, participants found that the content in TypeOut can cause more self-reflection and is more

acceptable. *"I have to read the sentence seriously before the typing. After reading them, I often think it is okay to use the phone later"* (P30). *"Typing some random words actually can help. But it is a bit annoying. I prefer the meaningful words as they can remind me of my decision [to reduce usage]"* (P5). Participants mentioned that the combination of value affirmation and improvisation was particularly helpful. *"During the typing, I would pause and re-think whether I actually need to use the phone right now... especially at the creation [improvisation] part where I would refer back to the previous [value] sentence and think about what I really need to do"* (P37). Even after typing the content and entering the app, participants could still recall the content. *"When using the app [after finishing typing], sometimes I remembered what I just typed [and leave the app]"* (P19). For some participants, this affirmation content affected their behavior during the break week. *"The content I typed would leave an impression in my mind and it sometimes pop up even when there is no intervention anymore"* (P37). These results indicated the advantages of TypeOut over baselines.

**Subjective Effectiveness.** Moreover, we also found a surprising finding in users' ranking of the three techniques: an equal number of participants picked TypeOut and TypingOnly (both 41.9%) as the most effective method. This is very interesting since our objective measure showed that TypeOut was significantly more effective than TypingOnly in terms of intervention acceptance rate, app opening frequency, as well as app usage duration, but a large proportion of participants thought the opposite. Our interviews revealed that these participants picked the TypingOnly mainly because it was the most "troublesome" technique. *"Compared to the meaningful content, the random content is more difficult to type since I have to type them one by one separately. So I often give up and quit the app. That's why I think [TypingOnly] is more effective."* (P2). *"[TypingOnly] pops up in my mind immediately because this one was so annoying and it intervened me many times. This is the most effective technique."* (P50). TypingOnly did have a fairly high intervention acceptance rate of 48.8% (compared to TypeOut's 57.2%) and this was also reflected by participants' feedback. However, participants did not realize that their overall app opening frequency and usage duration during the weeks of TypingOnly did not decrease compared to those of base week. Meanwhile, during the break week after the TypingOnly weeks,

participants had a big relapse in app opening frequency (19.0% more compared to the base week). Therefore, there was a clear discrepancy between users' perceived effectiveness and the actual effectiveness of these techniques.

### *7.3 Summary*

In summary, we propose a new JIT self-affirmation-based intervention technique, TypeOut. Based on the interpretation results from our depression prediction models, our first step applies this technique to problematic smartphone usage reduction. Our design integrates a JIT typing component that requires users to type a few words before accessing apps, and a brief self-affirmation exercise component is embedded in the typing content. We hypothesized that the combination of the two components can introduce more effective intervention than each component alone. We conducted a 10-week field experiment with 54 young adults to evaluate the effectiveness and usability of our technique. Our results indicate that TypeOut discourages 57.2% of app usage, and reduces overall app opening frequency by 26.8% and usage duration by 25.4%, all significantly outperforming baseline techniques. Moreover, our questionnaires and interview reveal that users find TypeOut to be more acceptable and cause more reflection than the baseline techniques. These results verify our hypothesis. Future work may want to consider including a wider range of application examples to replicate the work and help assess generalizability that goes beyond smartphone overuse reduction.

# Chapter 8
# CONCLUSION

As smart devices become more embedded in our everyday lives, achieving deployable behavior modeling techniques for longitudinal health and well-being has been increasingly important. In my thesis, I first introduced my previous work on passive sensing dataset collection (Ch. 3). Our team has collected and released the first multi-year passive sensing dataset with over 700 person-years across four years. I then introduced the new algorithms and frameworks that aim to address the three key challenges to achieve a deployable model: interpretability (revealing human-readable insights about behavior, Ch. 4), personalization (adapting to every individual, Ch. 5), and generalizability (working robustly on new users and contexts, Ch. 6). Using depression detection as an example, my methods have boosted the SOTA performance by 5-15% on various single- or multiple-dataset settings. Meanwhile, the new methods can generate interpretable behavior patterns for both populations and individuals to help researchers to better understand user behavior patterns.

However, developing algorithms is only the first step. To bring the results back to users, I further designed and deployed a JIT intervention technique to influence their behavior (Ch. 7). The insights of behavior models' personalized interpretation indicate that depression is strongly associated with smartphone overuse. As an initial step, I start with smartphone overuse intervention as the application. Our 10-week deployment study indicates that the new intervention technique has a significantly higher acceptance rate and can effectively reduce smartphone usage duration and frequency.

## *8.1 Contribution*

The contribution of my dissertation can be summarized as follows:

### *8.1.1 Algorithmic Contribution*

- **An Interpretable Behavior Modeling Algorithm**

  I developed the first algorithm that can automatically extract a set of highly interpretable behavior rules that can not only improve the model accuracy, but also reveal insights about users' behavior patterns.

- **A Personalized Behavior Modeling Algorithm**

  I developed the first algorithm to effectively leverage the large volume of behavior data and limited label data to enable accurate, personalized, and interpretable modeling for longitudinal behavior.

- **A Generalizable Behavior Modeling Algorithm**

  I developed the first algorithm that aims to address the challenges of cross-dataset generalization on longitudinal behavior data. The new algorithm significantly outperforms the state-of-the-art on various cross-dataset setups.

### *8.1.2 Design Contribution*

I developed the first intervention technique that combines self-affirmation theory and JIT design. I applied this design to smartphone overuse reduction. Our deployment study results reveal its effectiveness.

### *8.1.3 Open-source Contribution*

Working with a great team, we open-sourced the first multi-year passive mobile sensing dataset, as well as the first research platform **GLOBEM** to investigate the cross-dataset generalizability of longitudinal behavior models. The introduction of our datasets and platform are all available at https://the-globem.github.io/

### *8.1.4 Empirical Contribution*

My algorithms can also generate human-readable behavior rules at both the population level and the individual level. Many of our empirical findings of depression-related behavior are supported by psychology and psychiatry literature (*e.g.*, sleep patterns, phone usage,

physical activities). We also find some interesting potential new insights that may suggest new research questions for behavioral science researchers.

### *8.2 Limitation, Reflection, and Future Vision*

Like any other research, there are several limitations in this thesis. For our datasets (Ch. 3) and algorithms (Ch. 4-6), the drawbacks of the passive sensing-based methodology include the potential bias in EMA and other self-report surveys and the high data missing rate. In this thesis, I didn't investigate the reliability of self-report results or explore more depth into data imputation beyond naive methods. How to mitigate contextual bias in the self-report results? How to distinguish the causes of missing data and impute accordingly (e.g., software/hardware problems *vs.* changes/events of individuals' life experiences)? Future work can explore these questions in depth. Besides, my method relies on the behavior features extracted from the dataset using RAPIDS. Thus the capabilities of my methods are limited by these features. If these features do not capture some aspect of users' behavior, neither can my methods. There may exist more meaningful features to be extracted at the feature extraction stage. Moreover, the population of my research has mainly focused on young adults (*i.e.*, college students). Their behaviors are not representative of the general population. Future work can explore the population with different professionals and at various age groups.

This limitation also applies to the intervention technique deployment study (Ch. 7), where the main population is young adults with smartphone addiction problems. Additionally, researchers and practitioners have found that sometimes self-affirmation exercises can backfire when people fail to control their behavior, especially for those with low self-esteem: This can lead to disappointment and self-blame, and sometimes cause people to give up on their self-regulation, strengthening their original behavior (phone overuse, in our case). There is a wide future space to explore more intelligent and appropriate intervention timing and content, leading to the vision of just-in-time adaptive intervention (JITAI). Such a vision needs the combination of both novel ML algorithmic and HCI design.

I have made contributions from both ML and HCI aspects. Combining these efforts, I want to build next-generation AI-powered devices that can sense, understand, model, and

influence our daily behavior to facilitate our health and well-being.

I envision a future where everyone can design their own smart assistant that will be distributed around all devices. It can not only track our health behavior, but also really understand and help us to achieve our long-term goals for better health and well-being. This is the future I want to build.

### *8.3 Funding*

Some part of my research is supported by grants from UW, NSF, NiDILRR, Google, Samsung, and Adobe. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the view of the funders.

## ACKNOWLEDGMENTS

Memories come flooding back. The past five years have been fruitful, fulfilling, and passionate. I am lucky enough to have the freedom and support to explore a wide range of research questions that I am interested in. I feel as if I've been transported back to the days as a kid, when I traversed the world with a heart brimming with unadulterated wonder. I feel as if I've transformed into an eagle, soaring through the sky with nary a worry of descent. I feel as if I've metamorphosed into a whale, plunging into the depths of the ocean with an insatiable desire to unveil its hidden treasures. Five years have passed as an arrow, gone in the blink of an eye.

All things must come to an end. Time to graduate.

None of my great experiences will be even possible without meeting these amazing people, who have become good teachers, companions, and friends. I would like to thank many ones, who helped me learn and grow through the academic journey: to my amazing advisors, Anind Dey and Jennifer Mankoff, who are the best advisors I can ever imagine, the best supporters for any topic I want to explore, and the best teachers/friends providing consistent guidance and encouragement; to my committee members: Tim Althoff, who always provides extremely insightful advice when I got stuck; Andrew Campbell, who kindly connected the resources between multiple institutions and made substantial progress in our field together; and James Fogarty, whose research in health informatics has been inspiring my modeling and intervention work.

I have been fortunate to have a large number of collaborators, colleagues, and friends. I significantly benefit from collaborations, communications, and conversations. I would like to thank Tousif Ahmed, Ahmed Hassan Awadallah, Jennifer Brown, Carolina Brum, Prerna Chikersal, David Creswell, Mary Czerwinski, Alexandru Dancu, Aashaka Desai, Afsaneh Doryab, Susan Dumais, Jon Froehlich, Jun Alex Gao, Jun Gong, Liang He, Scott Hudson, Jilong Kuang, Gierad Laput, Xin Liu, Danli Luo, Charlie Maalouf, Kelly Mack,

Pattie Maes, Alexander Mariakakis, Daniel McDuff, Margaret Morris, Ebrahim Nemati, Subigya Nepal, Paula Nurius, Shwetak Patel, Venkatesh Potluri, Mahbubur Rahman, Yiyi Ren, Eve Riskin, Yasaman Sefidgar, Woosuk Seo, Ather Sharif, Yuanchun Shi, Misha Sra, Yuntao Wang, Jacob Wobbrock, Yukang Yan, Xin Yi, Chun Yu, Daqing Zhang, Han Zhang, Mingrui Ray Zhang, Tengxiang Zhang, Mingyuan Zhong. The list is so long, and I just list a small subset of the names alphabetically.

Most importantly, I want to thank my family and my dear partner, Ruoyu Peng, without whose support, care, and companionship I could not have reached this stage of my journey. Ruoyu is always there through all of my ups and downs, sharing my tears and joys. She is the North Star in the sky, illuminating my path that lies ahead. She is the candle in the dark, banishing shadows and bringing warmth. She is the sunshine in the morning, filling my soul with energy and inspiration.

Well, I guess that's it.

New life ahead. New goals to be achieved.

I've done my best. I will keep doing my best.

# BIBLIOGRAPHY

[1] Perceived Stress Scale 4 (PSS-4). http://www.ohnurses.org/wp-content/uploads/2015/05/Perceived-Stress-Scale-41.pdf.

[2] Phq-4: The four-item patient health questionnaire for anxiety and depression. https://www.oregonpainguidance.org/app/content/uploads/2016/05/PHQ-4.pdf.

[3] Positive and negative affect schedule (panas-sf). https://ogg.osu.edu/media/documents/MB%20Stream/PANAS.pdf.

[4] Measuring Discrimination Resource. https://scholar.harvard.edu/files/davidrwilliams/files/measuring_discrimination_resource_june_2016.pdf, 2016.

[5] 1,132 positive affirmations: Your daily list of simple mantras, Sep 2021.

[6] Saeed Abdullah, Nicholas D Lane, and Tanzeem Choudhury. Towards population scale activity recognition: A framework for handling data diversity. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[7] Saeed Abdullah, Mark Matthews, Elizabeth L. Murnane, Geri Gay, and Tanzeem Choudhury. Towards circadian computing: "early to bed and early to rise" makes some of us unhealthy and sleep deprived. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, page 673–684, New York, NY, USA, 2014. Association for Computing Machinery.

[8] Substance Abuse et al. Key substance use and mental health indicators in the united states. *National Survey on Drug Use and Health*, 2020.

[9] Alexander T. Adams, Jean Costa, Malte F. Jung, and Tanzeem Choudhury. Mindless computing: Designing technologies to subtly influence behavior. *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 719–730, 2015.

[10] Daniel A Adler, Fei Wang, David C Mohr, and Tanzeem Choudhury. Machine learning for passive mental health symptom prediction: Generalization across different longitudinal mobile sensing studies. *PLOS ONE*, page 20, 2022.

[11] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proceedings of 20th International Conference on Very Large Data Bases, VLDB*, volume 1215, pages 487–499, 1994.

[12] Mental Health America. The state of mental health in america, 2022.

[13] Maria-Luiza Antonie, Osmar R. Zaiane, and Alexandru Coman. Application of data mining techniques for medical image classification. In *Proceedings of the Second International Conference on Multimedia Data Mining*, pages 94–101. Springer-Verlag, 2001.

[14] Forest APP. Stay focused, be present, 2021.

[15] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant Risk Minimization. *arXiv:1907.02893 [cs, stat]*, March 2020. arXiv: 1907.02893.

[16] American Psychiatric Association et al. *Diagnostic and statistical manual of mental disorders (dsm-5®)*. American Psychiatric Pub, 2013.

[17] Christopher G Atkeson, Andrew W Moore, and Stefan Schaal. Locally weighted learning. In *Lazy Learning*, pages 11–73. Springer, 1997.

[18] Constantina Badea and David K Sherman. Self-affirmation and prejudice reduction: When and why? *Current Directions in Psychological Science*, 28(1):40–46, 2019.

[19] Sangwon Bae, Denzil Ferreira, Brian Suffoletto, Juan C. Puyana, Ryan Kurtz, Tammy Chung, and Anind K. Dey. Detecting drinking episodes in young adults using smartphone-based sensors. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 1(2), Jun 2017.

[20] Tadas Baltrušaitis, Marwa Mahmoud, and Peter Robinson. Cross-dataset learning and person-specific normalisation for automatic action unit detection. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 6, pages 1–6. IEEE, 2015.

[21] Nikola Banovic, Tofi Buzali, Fanny Chevalier, Jennifer Mankoff, and Anind K. Dey. Modeling and understanding human routine behavior. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 248–260, 2016. ISBN: 9781450333627.

[22] Samuel L. Battalio, David E. Conroy, Walter Dempsey, Peng Liao, Marianne Menictas, Susan Murphy, Inbal Nahum-Shani, Tianchen Qian, Santosh Kumar, and Bonnie Spring. Sense2Stop: A micro-randomized trial using wearable sensors to optimize a just-in-time-adaptive stress management intervention for smoking relapse prevention. *Contemporary Clinical Trials*, 109:106534, October 2021.

[23] Aaron T Beck, Robert A Steer, Roberta Ball, and William F Ranieri. Comparison of beck depression inventories-ia and-ii in psychiatric outpatients. *Journal of personality assessment*, 67(3):588–597, 1996.

[24] Dror Ben-Zeev, Emily A. Scherer, Rui Wang, Haiyi Xie, and Andrew T. Campbell. Next-generation psychiatric assessment: Using smartphone sensors to monitor behavior and mental health. *Psychiatric Rehabilitation Journal*, 38(3):218, 2015.

[25] José Bertolote. The roots of the concept of mental health. *World Psychiatry*, 7(2):113–116, June 2008.

[26] Lawrence D Bobo, Melvin L Oliver, Jr James H Johnson, and Valenzuela Abel Jr. *Prismatic metropolis: inequality in Los Angeles*. Russell Sage Foundation, 2000.

[27] George W. Brown, Bernice Andrews, Tirril Harris, Zsuzsanna Adler, and L. Bridge. Social support, self-esteem and depression. *Psychological medicine*, 16(4):813–831, 1986.

[28] Peter André Busch and Stephen McCarthy. Antecedents and consequences of problematic smartphone use: A systematic literature review of an emerging research area. *Computers in Human Behavior*, 114:106414, January 2021.

[29] Luca Canzian and Mirco Musolesi. Trajectories of depression: Unobtrusive monitoring of depressive states by means of smartphone mobility traces analysis. *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 1293–1304, 2015.

[30] Liangliang Cao, Zicheng Liu, and Thomas S Huang. Cross-dataset action detection. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1998–2005. IEEE, 2010.

[31] Fabio Maria Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain Generalization by Solving Jigsaw Puzzles. *arXiv:1903.06864 [cs]*, April 2019. arXiv: 1903.06864.

[32] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1721–1730, 2015.

[33] Charles S Carver. You want to measure coping but your protocol'too long: Consider the brief cope. *International journal of behavioral medicine*, 4(1):92–100, 1997.

[34] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[35] Prerna Chikersal, Afsaneh Doryab, Michael Tumminia, Daniella K Villalba, Janine M Dutcher, Xinwen Liu, Sheldon Cohen, Kasey G. Creswell, Jennifer Mankoff, J. David Creswell, Mayank Goel, and Anind K. Dey. Detecting Depression and Predicting its Onset Using Longitudinal Symptoms Captured by Passive Sensing. *ACM Transactions on Computer-Human Interaction*, 28(1):1–41, January 2021.

[36] Philip I. Chow, Karl Fua, Yu Huang, Wesley Bonelli, Haoyi Xiong, Laura E. Barnes, and Bethany A. Teachman. Using mobile sensing to test clinical models of depression, social anxiety, state affect, and social isolation among college students. *Journal of Medical Internet Research*, 19(3), 2017.

[37] Yucel Cimtay and Erhan Ekmekcioglu. Investigating the use of pretrained convolutional neural network on cross-subject and cross-dataset eeg emotion recognition. *Sensors*, 20(7):2034, 2020.

[38] Geoffrey L Cohen, Julio Garcia, Valerie Purdie-Vaughns, Nancy Apfel, and Patricia Brzustoski. Recursive processes in self-affirmation: Intervening to close the minority achievement gap. *science*, 324(5925):400–403, 2009.

[39] Sheldon Cohen and Harry M Hoberman. Positive events and social supports as buffers of life change stress 1. *Journal of applied social psychology*, 13(2):99–125, 1983.

[40] Sheldon Cohen, Tom Kamarck, and Robin Mermelstein. A global measure of perceived stress. *Journal of health and social behavior*, pages 385–396, 1983.

[41] Vicki S Conn, Adam R Hafdahl, Pamela S Cooper, Lori M Brown, and Sally L Lusk. Meta-analysis of workplace physical activity interventions. *American journal of preventive medicine*, 37(4):330–339, 2009.

[42] Anna L. Cox, Sandy J.J. Gould, Marta E. Cecchinato, Ioanna Iacovides, and Ian Renfree. Design Frictions for Mindful Interactions: The Case for Microboundaries. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '16, pages 1389–1397, New York, NY, USA, May 2016. Association for Computing Machinery.

[43] Taylor Cox. *Cultural diversity in organizations: Theory, research and practice*. Berrett-Koehler Publishers, 1994.

[44] Hal Daumé III. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[45] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, May 2019. arXiv: 1810.04805.

[46] Ed Diener, Derrick Wirtz, William Tov, Chu Kim-Prieto, Dong-won Choi, Shigehiro Oishi, and Robert Biswas-Diener. New well-being measures: Short scales to assess flourishing and positive and negative feelings. *Social indicators research*, 97(2):143–156, 2010.

[47] Afsaneh Doryab, Jun Ki Min, Jason Wiese, John Zimmerman, and Jason Hong. Detection of behavior change in people with depression. *AAAI Workshop - Technical Report*, WS-14-08:12–16, 2014. ISBN: 9781577356691.

[48] Qi Dou, Daniel C. Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain Generalization via Model-Agnostic Learning of Semantic Features. *arXiv:1910.13580 [cs]*, October 2019. arXiv: 1910.13580.

[49] Jon D Elhai, Robert D Dvorak, Jason C Levine, and Brian J Hall. Problematic smartphone use: A conceptual overview and systematic review of relations with anxiety and depression psychopathology. *Journal of affective disorders*, 207:251–259, 2017.

[50] Richard Emanuel, Rodney Bell, Cedric Cotton, Jamon Craig, Danielle Drummond, Samuel Gibson, Ashley Harris, Marcus Harris, Chelsea Hatcher-Vance, Staci Jones, et al. The truth about smartphone addiction. *College Student Journal*, 49(2):291–299, 2015.

[51] Tracy Epton and Peter R. Harris. Self-affirmation promotes health behavior change. *Health Psychology*, 27(6):746–752, 2008.

[52] Tracy Epton, Peter R. Harris, Rachel Kane, Guido M. van Koningsbruggen, and Paschal Sheeran. The impact of self-affirmation on health-behavior change: A meta-analysis. *Health Psychology*, 34(3):187–196, 2015.

[53] Emily B Falk, Matthew Brook O'Donnell, Christopher N Cascio, Francis Tinney, Yoona Kang, Matthew D Lieberman, Shelley E Taylor, Lawrence An, Kenneth Resnicow, and Victor J Strecher. Self-affirmation alters the brain's response to health messages and subsequent behavior change. *Proceedings of the National Academy of Sciences*, 112(7):1977–1982, 2015.

[54] Asma Ahmad Farhan, Chaoqun Yue, Reynaldo Morillo, Shweta Ware, Jin Lu, Jinbo Bi, Jayesh Kamath, Alexander Russell, Athanasios Bamis, and Bing Wang. Behavior vs. introspection: refining prediction of clinical depression via smartphone sensing data. In *2016 IEEE Wireless Health (WH)*, pages 1–8. IEEE, October 2016.

[55] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

[56] Steven Fein and Steven J Spencer. Prejudice as self-image maintenance: Affirming the self through derogating others. *Journal of personality and Social Psychology*, 73(1):31, 1997.

[57] Denzil Ferreira, Vassilis Kostakos, and Anind K. Dey. Aware: Mobile context instrumentation framework. *Frontiers in ICT*, 2:6, 2015.

[58] Bj Fogg. The behavior grid: 35 Ways behavior can change. *Proceedings of the International Conference on Persuasive Technology,*, 350:1–5, 2009.

[59] Philippe Fournier-Viger, Antonio Gomariz, Ted Gueniche, Azadeh Soltani, Cheng-Wei Wu, and Vincent S. Tseng. Spmf: A java open-source pattern mining library. *The Journal of Machine Learning Research*, 15(1):3389–3393, 2014.

[60] Keith Frankish. Dual-Process and Dual-System Theories of Reasoning. *Philosophy Compass*, 5(10):914–926, 2010. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1747-9991.2010.00330.x.

[61] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[62] Jerome Friedman, Ron Kohavi, and Yeogirl Yun. Lazy decision trees. *Proceedings of the AAAI*, 1, 09 1997.

[63] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-Adversarial Training of Neural Networks. In *Domain Adaptation in Computer Vision Applications*, pages 189–209. Springer International Publishing, Cham, 2017. Series Title: Advances in Computer Vision and Pattern Recognition.

[64] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. In Gabriela Csurka, editor, *Domain Adaptation in Computer Vision Applications*, pages 189–209. Springer International Publishing, 2017. Series Title: Advances in Computer Vision and Pattern Recognition.

[65] Yaoguo Geng, Jingjing Gu, Jing Wang, and Ruiping Zhang. Smartphone addiction and depression, anxiety: The role of bedtime procrastination and self-control. *Journal of affective disorders*, 293:415–421, 2021.

[66] D. Goldberg, K. Bridges, P. Duncan-Jones, and D. Grayson. Detecting anxiety and depression in general medical settings. *British Medical Journal*, 297(6653):897–899, 1988.

[67] Shirley Gregor and Izak Benbasat. Explanations from intelligent systems: Theoretical foundations and implications for practice. *MIS quarterly*, pages 497–530, 1999.

[68] James J Gross and Oliver P John. Individual differences in two emotion regulation processes: implications for affect, relationships, and well-being. *Journal of personality and social psychology*, 85(2):348, 2003.

[69] Ishaan Gulrajani and David Lopez-Paz. In Search of Lost Domain Generalization. *International Conference on Learning Representations 2021*, page 29, 2021.

[70] David Gunning and David Aha. Darpa's explainable artificial intelligence (xai) program. *AI magazine*, 40(2):44–58, 2019.

[71] Xifeng Guo, Xinwang Liu, En Zhu, and Jianping Yin. Deep clustering with convolutional autoencoders. In *International conference on neural information processing*, pages 373–382. Springer, 2017.

[72] Martin S Hagger. Non-conscious processes and dual-process theories in health psychology. *Health Psychology Review*, 10(4):375–380, 2016.

[73] Jonathan Haidt and Nick Allen. Scrutinizing the effects of digital technology on mental health, 2020.

[74] Gabriella M Harari, Sandrine R Müller, Min SH Aung, and Peter J Rentfrow. Smartphone sensing methods for studying behavior in everyday life. *Current Opinion in Behavioral Sciences*, 18:83–90, 2017.

[75] Russ Harris. The complete set of client handouts and worksheets from act books, 2009.

[76] Andree Hartanto and Hwajin Yang. Is the smartphone a smart choice? the effect of smartphone separation on executive functions. *Computers in Human Behavior*, 64:329–336, 2016.

[77] Joshua Harwood, Julian J Dooley, Adrian J Scott, and Richard Joiner. Constantly connected–the effects of smart-devices on mental health. *Computers in Human Behavior*, 34:267–272, 2014.

[78] Steven C Hayes, Kirk D Strosahl, and Kelly G Wilson. *Acceptance and commitment therapy*. American Psychological Association Washington, DC, 2009.

[79] Alexis Hiniker, Sungsoo Hong, Tadayoshi Kohno, and Julie A. Kientz. MyTime: Designing and evaluating an intervention for smartphone non-use. *Conference on Human Factors in Computing Systems - Proceedings*, pages 4746–4757, 2016.

[80] Geoffrey M Hodgson. The ubiquity of habits and rules. *Cambridge journal of economics*, 21(6):663–684, 1997.

[81] Steven Hoffman, Renu Sharma, and Arun Ross. Convolutional neural networks for iris presentation attack detection: Toward cross-dataset and cross-sensor generalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1620–1628, 2018.

[82] Wilhelm Hofmann, Malte Friese, and Fritz Strack. Impulse and Self-Control From a Dual-Systems Perspective. *Perspectives on Psychological Science*, 4(2):162–176, March 2009.

[83] Daniel Holliday, Stephanie Wilson, and Simone Stumpf. User trust in intelligent systems: A journey over time. In *Proceedings of the 21st international conference on intelligent user interfaces*, pages 164–168, 2016.

[84] Xiao Hu and Yi-Hsuan Yang. Cross-dataset and cross-cultural music mood prediction: A case on western and chinese pop songs. *IEEE Transactions on Affective Computing*, 8(2):228–240, 2017.

[85] Rob J Hyndman and Yanan Fan. Sample quantiles in statistical packages. *The American Statistician*, 50(4):361–365, 1996.

[86] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5001–5009, 2018.

[87] Raghavendra Katikalapudi, Sriram Chellappan, Frances Montgomery, Donald Wunsch, and Karl Lutzen. Associating internet usage with depressive behavior among college students. *IEEE Technology and Society Magazine*, 31(4):73–80, 2012.

[88] James M Keller, Michael R Gray, and James A Givens. A fuzzy k-nearest neighbor algorithm. *IEEE Transactions on Systems, Man, and Cybernetics*, pages 580–585, 1985.

[89] Rohit Ashok Khot, Jeewon Lee, Deepti Aggarwal, Larissa Hjorth, and Florian'Floyd' Mueller. Tastybeats: Designing palatable representations of physical activity. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 2933–2942, 2015.

[90] Daehee Kim, Youngjun Yoo, Seunghyun Park, Jinkyu Kim, and Jaekoo Lee. SelfReg: Self-Supervised Contrastive Regularization for Domain Generalization. *Proceedings of the IEEE/CVF International Conference on Computer Vision.*, page 10, 2021.

[91] Inyeop Kim, Gyuwon Jung, Hayoung Jung, Minsam Ko, and Uichin Lee. Let's focus: location-based intervention tool to mitigate phone use in college classrooms. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, UbiComp '17, pages 101–104, New York, NY, USA, September 2017. Association for Computing Machinery.

[92] Jaejeung Kim, Chiwoo Cho, and Uichin Lee. Technology Supported Behavior Restriction for Mitigating Self-Interruptions in Multi-device Environments. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):64:1–64:21, September 2017.

[93] Jaejeung Kim, Hayoung Jung, Minsam Ko, and Uichin Lee. GoalKeeper: Exploring Interaction Lockout Mechanisms for Regulating Smartphone Use. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(1):1–29, 2019.

[94] Jaejeung Kim, Joonyoung Park, Hyunsoo Lee, Minsam Ko, and Uichin Lee. LocknType: Lockout task intervention for discouraging smartphone app use. *Conference on Human Factors in Computing Systems - Proceedings*, pages 1–12, 2019.

[95] Minsam Ko, Seungwoo Choi, Koji Yatani, and Uichin Lee. Lock n' LoL: Group-based Limiting Assistance App to Mitigate Smartphone Distractions in Group Activities. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 998–1010. Association for Computing Machinery, New York, NY, USA, May 2016.

[96] Minsam Ko, Subin Yang, Joonwon Lee, Christian Heizmann, Jinyoung Jeong, Uichin Lee, Daehee Shin, Koji Yatani, Junehwa Song, and Kyong-Mee Chung. NUGU: A Group-based Intervention App for Improving Self-Regulation of Limiting Smartphone Use. page 11, 2015.

[97] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese Neural Networks for One-shot Image Recognition. *Proceedings of the 32nd International Conference on Machine Learning*, page 8, 2015.

[98] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

[99] Jan-Niklas Kramer, Florian Künzler, Varun Mishra, Bastien Presset, David Kotz, Shawna Smith, Urte Scholz, and Tobias Kowatsch. Investigating intervention components and exploring states of receptivity for a smartphone app to promote physical activity: Protocol of a microrandomized trial. *JMIR research protocols*, 8(1):e11540, 2019.

[100] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E*, 69(6):066138, 2004.

[101] Kurt Kroenke, Robert L Spitzer, Janet BW Williams, and Bernd Löwe. An ultra-brief screening scale for anxiety and depression: The phq–4. *Psychosomatics*, 50(6):613–621, 2009.

[102] Min Kwon, Dai-Jin Kim, Hyun Cho, and Soo Yang. The Smartphone Addiction Scale: Development and Validation of a Short Version for Adolescents. *PLoS ONE*, 8(12):e83558, December 2013.

[103] Nicholas D. Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T. Campbell. A survey of mobile phone sensing. *IEEE Communications Magazine*, 48(9), 2010.

[104] Nicholas D. Lane, Ye Xu, Hong Lu, Shaohan Hu, Tanzeem Choudhury, Andrew T. Campbell, and Feng Zhao. Enabling large-scale human activity inference on smartphones using community similarity networks (csn). In *Proceedings of the 13th International Conference on Ubiquitous Computing*, UbiComp '11, page 355–364, New York, NY, USA, 2011. Association for Computing Machinery.

[105] Liette Lapointe, Camille Boudreau-Pinsonneault, and Isaac Vaghefi. Is smartphone usage truly smart? a qualitative investigation of it addictive behaviors. In *2013 46th Hawaii international conference on system sciences*, pages 1063–1072. IEEE, 2013.

[106] Robert LaRose. Uses and Gratifications of Internet Addiction. In *Internet Addiction*, pages 55–72. John Wiley & Sons, Ltd, 2007. Section: 04 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118013991.ch4.

[107] Ben Lengerich, Bryon Aragam, and Eric P Xing. Learning sample-specific models with low-rank personalized regression. In *Advances in Neural Information Processing Systems*, pages 3570–3580, 2019.

[108] Benjamin Letham, Cynthia Rudin, Tyler H McCormick, and David Madigan. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371, 2015.

[109] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to Generalize: Meta-Learning for Domain Generalization. *arXiv:1710.03463 [cs]*, October 2017. arXiv: 1710.03463.

[110] Brian Y. Lim and Anind K. Dey. Assessing demand for intelligibility in context-aware applications. In *Proceedings of the 11th international conference on Ubiquitous computing*, pages 195–204, Orlando Florida USA, September 2009. ACM.

[111] Brian Y. Lim, Anind K. Dey, and Daniel Avrahami. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2119–2128, Boston MA USA, April 2009. ACM.

[112] Hajin Lim, Ian Arawjo, Yaxian Xie, Negar Khojasteh, and Susan R. Fussell. Distraction or Life Saver? The Role of Technology in Undergraduate Students' Boundary Management Strategies. *Proceedings of the ACM on Human-Computer Interaction*, 1(CSCW):68:1–68:18, December 2017.

[113] Bing Liu, Wynne Hsu, and Yiming Ma. Integrating classification and association rule mining. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, 1998.

[114] Edwin A Locke and Gary P Latham. New directions in goal-setting theory. *Current directions in psychological science*, 15(5):265–268, 2006.

[115] Daniel Lopez-Martinez, Ognjen Rudovic, and Rosalind Picard. Physiological and behavioral profiling for nociceptive pain estimation using personalized multitask learning. *Neural Information Processing Systems Workshop on Machine Learning for Health*, 2017.

[116] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[117] Jin Lu, Jinbo Bi, Chao Shang, Chaoqun Yue, Reynaldo Morillo, Shweta Ware, Jayesh Kamath, Athanasios Bamis, Alexander Russell, and Bing Wang. Joint Modeling of Heterogeneous Sensing Data for Depression Assessment via Multi-task Learning. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(1):1–21, 2018. ISBN: 9781450351980.

[118] Scott M. Lundberg and Su In Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 2017-Decem(Section 2):4766–4775, 2017.

[119] Ulrik Lyngs, Kai Lukoff, Petr Slovak, Reuben Binns, Adam Slack, Michael Inzlicht, Max Van Kleek, and Nigel Shadbolt. Self-control in cyberspace: Applying dual systems theory to a review of digital self-control tools. *Conference on Human Factors in Computing Systems - Proceedings*, pages 1–18, 2019.

[120] Markus Löchtefeld, Matthias Böhmer, and Lyubomir Ganev. AppDetox: helping users with mobile app addiction. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia*, MUM '13, pages 1–2, New York, NY, USA, December 2013. Association for Computing Machinery.

[121] Kody J Manke, Shannon T Brady, Mckenzie D Baker, and Geoffrey L Cohen. Affirmation on the go: A proof-of-concept for text message delivery of values affirmation in education. *Journal of Social Issues*, 2021.

[122] Stephen M Mattingly, Julie M Gregg, Pino Audia, Ayse Elvan Bayraktaroglu, Andrew T Campbell, Nitesh V Chawla, Vedant Das Swain, Munmun De Choudhury, Sidney K D'Mello, Anind K Dey, et al. The tesserae project: Large-scale, longitudinal, in situ, multimodal sensing of information workers. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–8, 2019.

[123] Michael E McCullough, Robert A Emmons, and Jo-Ann Tsang. The grateful disposition: a conceptual and empirical topography. *Journal of personality and social psychology*, 82(1):112, 2002.

[124] Amy McQueen and William MP Klein. Experimental manipulations of self-affirmation: A systematic review. *Self and Identity*, 5(4):289–354, 2006.

[125] Susan Michie, Maartje M van Stralen, and Robert West. The behaviour change wheel: A new method for characterising and designing behaviour change interventions. *Implementation Science*, 6(1):42, December 2011.

[126] Frances J Milliken and Luis L Martins. Searching for common threads: Understanding the multiple effects of diversity in organizational groups. *Academy of management review*, 21(2):402–433, 1996.

[127] Jun-Ki Min, Afsaneh Doryab, Jason Wiese, Shahriyar Amini, John Zimmerman, and Jason I. Hong. Toss "n" turn: Smartphone as sleep and sleep quality detector. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, page 477–486, New York, NY, USA, 2014. Association for Computing Machinery.

[128] Shayan Mirjafari, Kizito Masaba, Ted Grover, Weichen Wang, Pino G. Audia, Andrew T. Campbell, Nitesh V. Chawla, Vedant Das Swain, Munmun De Choudhury, Anind K. Dey, et al. Differentiating higher and lower job performers in the workplace using mobile sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 3(2):37:1–37:24, 2019.

[129] Varun Mishra, Florian Künzler, Jan-Niklas Kramer, Elgar Fleisch, Tobias Kowatsch, and David Kotz. Detecting Receptivity for mHealth Interventions in the Natural Environment. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(2):1–24, June 2021.

[130] Varun Mishra, Sougata Sen, Grace Chen, Tian Hao, Jeffrey Rogers, Ching Hua Chen, and David Kotz. Evaluating the reproducibility of physiological stress detection models. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(4), 2020.

[131] Brenton Muñoz, Joseph P Magliano, Robin Sheridan, and Danielle S McNamara. Typing versus thinking aloud when reading: Implications for computer-based assessment and training tools. *Behavior research methods*, 38(2):211–217, 2006.

[132] Inbal Nahum-Shani, Shawna N Smith, Bonnie J Spring, Linda M Collins, Katie Witkiewitz, Ambuj Tewari, and Susan A Murphy. Just-in-time adaptive interventions (jitais) in mobile health: key components and design principles for ongoing health behavior support. *Annals of Behavioral Medicine*, 52(6):446–462, 2018.

[133] Daniel Nettle. The evolution of personality variation in humans and other animals. *American Psychologist*, 61(6):622, 2006.

[134] Shirin Nilizadeh, Hojjat Aghakhani, Eric Gustafson, Christopher Kruegel, and Giovanni Vigna. Think Outside the Dataset: Finding Fraudulent Reviews using Cross-Dataset Analysis. pages 3108–3115, 2019. ISBN: 9781450366748.

[135] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

[136] Brendan Nyhan and Jason Reifler. When corrections fail: The persistence of political misperceptions. *Political Behavior*, 32(2):303–330, 2010.

[137] Fabian Okeke, Michael Sobolev, Nicola Dell, and Deborah Estrin. Good vibrations: can a digital nudge reduce digital overload? In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '18, pages 1–12, New York, NY, USA, September 2018. Association for Computing Machinery.

[138] World Health Organization et al. Covid-19 pandemic triggers 25% increase in prevalence of anxiety and depression worldwide. *Retrieved from World Health Organization: https://www. who. int/news/item/02-03-2022-covid-19-pandemic-triggers-25-increase-in-prevalence-of-anxiety-and-depression-worldwide*, 2022.

[139] Rita Orji and Karyn Moffatt. Persuasive technology for health and wellness: State-of-the-art and emerging trends. *Health informatics journal*, 24(1):66–91, 2018.

[140] Jaya L. Padmanabhan, Danielle Cooke, Juho Joutsa, Shan H. Siddiqi, et al. A human depression circuit derived from focal brain lesions. *Biological Psychiatry*, 86(10):749–758, 2019. Cortical Pathology and Depression.

[141] Jong Soo Park, Ming-Syan Chen, and Philip S. Yu. Using a hash-based method with transaction trimming for mining association rules. *IEEE Transactions on Knowledge and Data Engineering*, 9(5):813–825, 1997.

[142] Olga Perski, Ann Blandford, Robert West, and Susan Michie. Conceptualising engagement with digital behaviour change interventions: a systematic review using principles from critical interpretive synthesis. *Translational behavioral medicine*, 7(2):254–267, 2017.

[143] Richard E Petty and John T Cacioppo. The elaboration likelihood model of persuasion. In *Communication and persuasion*, pages 1–24. Springer, 1986.

[144] Charlie Pinder, Jo Vermeulen, Benjamin R. Cowan, and Russell Beale. Digital Behaviour Change Interventions to Break and Form Habits. *ACM Transactions on Computer-Human Interaction*, 25(3):1–66, 2018.

[145] Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Efficient Domain Generalization via Common-Specific Low-Rank Decomposition. *arXiv:2003.12815 [cs, stat]*, April 2020. arXiv: 2003.12815.

[146] Alun Preece, Dan Harborne, Dave Braines, Richard Tomsett, and Supriyo Chakraborty. Stakeholders in explainable ai. *arXiv preprint arXiv:1810.00184*, 2018.

[147] Andrew K. Przybylski, Kou Murayama, Cody R. DeHaan, and Valerie Gladwell. Motivational, emotional, and behavioral correlates of fear of missing out. *Computers in Human Behavior*, 29(4):1841–1848, July 2013.

[148] Pearl Pu and Li Chen. Trust building with explanation interfaces. In *Proceedings of the 11th international conference on Intelligent user interfaces*, pages 93–100, 2006.

[149] J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[150] Mashfiqui Rabbi, Min Hane Aung, Mi Zhang, and Tanzeem Choudhury. MyBehavior: Automatic Personalized Health Feedback from User Behaviors and Preferences using Smartphones. *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, (September):707–718, 2015.

[151] Beatrice Rammstedt and Oliver P John. Measuring personality in one minute or less: A 10-item short version of the big five inventory in english and german. *Journal of research in Personality*, 41(1):203–212, 2007.

[152] Markus Riester, Wei Wei, Levi Waldron, Aedin C Culhane, , et al. Risk prediction for late-stage ovarian cancer by meta-analysis of 1525 patient samples. *Journal of the National Cancer Institute*, 106(5):dju048, 2014.

[153] Katerina Rnic, David JA Dozois, and Rod A Martin. Cognitive distortions, humor styles, and depression. *Europe's journal of psychology*, 12(3):348, 2016.

[154] George Rosen. *A history of public health*. Jhu Press, 2015.

[155] Daniel W Russell. Ucla loneliness scale (version 3): Reliability, validity, and factor structure. *Journal of personality assessment*, 66(1):20–40, 1996.

[156] Sohrab Saeb, Mi Zhang, Christopher J. Karr, Stephen M. Schueller, Marya E. Corden, Konrad P. Kording, and David C. Mohr. Mobile phone sensor correlates of depressive symptom severity in daily-life behavior: An exploratory study. *Journal of Medical Internet Research*, 17(7):1–11, 2015.

[157] Tara N Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4580–4584. IEEE, 2015.

[158] Suhila Sawesi, Mohamed Rashrash, Kanitha Phalakornkule, Janet S Carpenter, and Josette F Jones. The impact of information technology on patient engagement and health behavior change: a systematic review of the literature. *JMIR medical informatics*, 4(1):e4514, 2016.

[159] Brandon J Schmeichel and Kathleen Vohs. Self-affirmation and self-control: affirming core values counteracts ego depletion. *Journal of personality and social psychology*, 96(4):770, 2009.

[160] Peter Schulam and Suchi Saria. A framework for individualizing predictions of disease trajectories by exploiting multi-resolution structure. In *Advances in Neural Information Processing Systems*, pages 748–756, 2015.

[161] Natalie Schüz, Benjamin Schüz, and Michael Eid. When risk communication backfires: Randomized controlled trial on self-affirmation and reactance to personalized risk feedback in high-risk individuals. *Health Psychology*, 32(5):561, 2013.

[162] Raymond Scupin. The kj method: A technique for analyzing data derived from japanese ethnology. *Human organization*, 56(2):233–237, 1997.

[163] Yasaman S. Sefidgar, Woosuk Seo, Kevin S. Kuehn, Tim Althoff, Anne Browning, Eve Riskin, Paula S. Nurius, Anind K. Dey, and Jennifer Mankoff. Passively-sensed behavioral correlates of discrimination events in college students. *Proc. ACM Hum.-Comput. Interact.*, 3(CSCW), Nov 2019.

[164] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[165] Edwin Shen, Justin Shen, and Tsorng-Lin Chia. Development of an App to Support Self-monitoring Smartphone Usage and Healthcare Behaviors in Daily Life. In *Proceedings of the 3rd International Conference on Big Data and Internet of Things*, BDIOT 2019, pages 29–34, New York, NY, USA, August 2019. Association for Computing Machinery.

[166] Amitai Shenhav, Matthew M. Botvinick, and Jonathan D. Cohen. The Expected Value of Control: An Integrative Theory of Anterior Cingulate Cortex Function. *Neuron*, 79(2):217–240, July 2013.

[167] David K Sherman and Geoffrey L Cohen. The psychology of self-defense: Self-affirmation theory. *Advances in experimental social psychology*, 38:183–242, 2006.

[168] David K Sherman, Kimberly A Hartson, Kevin R Binning, Valerie Purdie-Vaughns, Julio Garcia, Suzanne Taborsky-Barba, Sarah Tomassetti, A David Nussbaum, and Geoffrey L Cohen. Deflecting the trajectory and changing the narrative: how self-affirmation affects academic performance and motivation under identity threat. *Journal of Personality and Social Psychology*, 104(4):591, 2013.

[169] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. *34th International Conference on Machine Learning, ICML 2017*, 7:4844–4866, 2017. ISBN: 9781510855144.

[170] Bruce W Smith, Jeanne Dalen, Kathryn Wiggins, Erin Tooley, Paulette Christopher, and Jennifer Bernard. The brief resilience scale: Assessing the ability to bounce back. *International journal of behavioral medicine*, 15(3):194–200, 2008.

[171] Aaron Springer, Anusha Venkatakrishnan, Shiwali Mohan, Lester Nelson, Michael Silva, and Peter Pirolli. Leveraging self-affirmation to improve behavior change: a mobile health app experiment. *JMIR mHealth and uHealth*, 6(7):e157, 2018.

[172] Misha Sra, Xuhai Xu, and Pattie Maes. BreathVR: Leveraging Breathing as a Directly Controlled Interface for Virtual Reality Games. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 1–12, Montreal QC Canada, April 2018. ACM.

[173] Keith E Stanovich and Richard F West. Individual differences in rational thought. *Journal of experimental psychology: general*, 127(2):161, 1998.

[174] Claude M Steele. The psychology of self-affirmation: Sustaining the integrity of the self. In *Advances in experimental social psychology*, volume 21, pages 261–302. Elsevier, 1988.

[175] Fritz Strack and Roland Deutsch. Reflective and impulsive determinants of social behavior. *Personality and social psychology review*, 8(3):220–247, 2004.

[176] Craig W Strohmeier, Brad Rosenfield, Robert A DiTomasso, and J Russell Ramsay. Assessment of the relationship between self-reported cognitive distortions and adult adhd, anxiety, depression, and hopelessness. *Psychiatry research*, 238:153–158, 2016.

[177] Yoshihiko Suhara, Yinzhan Xu, and Alex 'Sandy' Pentland. Deepmood: Forecasting depressed mood based on self-reported histories via recurrent neural networks. In *Proceedings of the 26th International Conference on World Wide Web*, pages 715–724. International World Wide Web Conferences Steering Committee, 2017.

[178] Allison M Sweeney and Anne Moyer. Self-affirmation and responses to health messages: A meta-analysis on intentions and behavior. *Health Psychology*, 34(2):149, 2015.

[179] Jennifer M Taber, Amy McQueen, Nicolle Simonovic, and Erika A Waters. Adapting a self-affirmation intervention for use in a mobile application for smokers. *Journal of behavioral medicine*, 42(6):1050–1061, 2019.

[180] J Eric T Taylor and Graham W Taylor. Artificial cognition: How experimental psychology can help generate explainable artificial intelligence. *Psychonomic Bulletin & Review*, 28(2):454–475, 2021.

[181] Michael E. Thase. Depression, sleep, and antidepressants. *The Journal of Clinical Psychiatry*, 1998.

[182] Lu Tian, Tianxi Cai, Els Goetghebeur, and LJ Wei. Model evaluation based on the sampling distribution of estimated absolute prediction error. *Biometrika*, 94(2):297–311, 2007.

[183] Catalina L Toma and Jeffrey T Hancock. Self-affirmation underlies facebook use. *Personality and Social Psychology Bulletin*, 39(3):321–331, 2013.

[184] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *CVPR 2011*, volume 2011, pages 1521–1528. IEEE, June 2011. Issue: 28.

[185] Paul D Trapnell and Jennifer D Campbell. Private self-consciousness and the five-factor model of personality: distinguishing rumination from reflection. *Journal of personality and social psychology*, 76(2):284, 1999.

[186] Norifumi Tsuno, Alain Besset, and Karen Ritchie. Sleep and depression. *The Journal of Clinical Psychiatry*, 2005.

[187] Ofir Turel, Alexander Serenko, and Nick Bontis. Blackberry addiction: Symptoms and outcomes. *AMCIS 2008 Proceedings*, page 73, 2008.

[188] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE international conference on computer vision*, pages 4068–4076, 2015.

[189] European Union. General data protection regulation (gdpr), Sep 2019.

[190] Guido M Van Koningsbruggen, Enny Das, and David R Roskos-Ewoldsen. How self-affirmation reduces defensive processing of threatening health information: evidence at the implicit level. *Health Psychology*, 28(5):563, 2009.

[191] Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

[192] Julio Vega, Meng Li, Kwesi Aguillera, Nikunj Goel, Echhit Joshi, Kirtiraj Khandekar, Krina C Durica, Abhineeth R Kunta, and Carissa A Low. Reproducible analysis pipeline for data streams: Open-source software to process data collected with mobile devices. *Frontiers in Digital Health*, 3, 2021.

[193] Shyam Visweswaran and Gregory F Cooper. Learning instance-specific predictive models. *Journal of Machine Learning Research*, 11(Dec):3333–3369, 2010.

[194] Michael Von Korff and Gregory Simon. The relationship between pain and depression. *The British Journal of Psychiatry*, 168(S30):101–108, 1996.

[195] Fabian Wahle, Tobias Kowatsch, Elgar Fleisch, Michael Rufer, and Steffi Weidt. Mobile Sensing and Support for People With Depression: A Pilot Trial in the Wild. *JMIR mHealth and uHealth*, 4(3):e111, 2016.

[196] Gregory M Walton and Geoffrey L Cohen. A question of belonging: race, social fit, and achievement. *Journal of personality and social psychology*, 92(1):82, 2007.

[197] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S. Yu. Generalizing to Unseen Domains: A Survey on Domain Generalization. *arXiv:2103.03097 [cs]*, December 2021. arXiv: 2103.03097.

[198] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. Publisher: Elsevier B.V.

[199] Rui Wang, Fanglin Chen, Zhenyu Chen, Tianxing Li, Gabriella Harari, Stefanie Tignor, Xia Zhou, Dror Ben-Zeev, and Andrew T. Campbell. Studentlife: Assessing mental health, academic performance and behavioral trends of college students using smartphones. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 3–14. ACM, 2014.

[200] Rui Wang, Gabriella Harari, Peilin Hao, Xia Zhou, and Andrew T Campbell. Smartgpa: how smartphones can assess and predict academic performance of college students. In *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*, pages 295–306, 2015.

[201] Rui Wang, Weichen Wang, Alex daSilva, Jeremy F. Huckins, William M. Kelley, Todd F. Heatherton, and Andrew T. Campbell. Tracking Depression Dynamics in College Students Using Mobile Phone and Wearable Sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(1):1–26, 2018. ISBN: 2474-9567.

[202] David Watson, Lee Anna Clark, and Auke Tellegen. Development and validation of brief measures of positive and negative affect: the panas scales. *Journal of personality and social psychology*, 54(6):1063, 1988.

[203] Pei-Shan Wei and Hsi-Peng Lu. Why do people play mobile social games? An examination of network externalities and of uses and gratifications. *Internet Research*, 24(3):313–331, May 2014.

[204] David R Williams, Yan Yu, James S Jackson, and Norman B Anderson. Racial differences in physical and mental health: Socio-economic status, stress and discrimination. *Journal of health psychology*, 2(3):335–351, 1997.

[205] R.M. Winters. *10,000 Positive Affirmations*. 2020.

[206] Russ Wolfinger and Michael O'connell. Generalized linear mixed models a pseudo-likelihood approach. *Journal of statistical Computation and Simulation*, 48(3-4):233–243, 1993.

[207] Joanne V Wood, WQ Elaine Perunovic, and John W Lee. Positive self-statements: Power for some, peril for others. *Psychological Science*, 20(7):860–866, 2009.

[208] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised Deep Embedding for Clustering Analysis. *Proceedings of the 33 rd International Conference on Machine Learning*, page 10, 2016.

[209] Xuhai Xu, Prerna Chikersal, Afsaneh Doryab, Daniella K. Villalba, Janine M. Dutcher, Michael J. Tumminia, Tim Althoff, Sheldon Cohen, Kasey G. Creswell, J. David Creswell, Jennifer Mankoff, and Anind K. Dey. Leveraging Routine Behavior and Contextually-Filtered Features for Depression Detection among College Students. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(3):1–33, September 2019.

[210] Xuhai Xu, Prerna Chikersal, Janine M. Dutcher, Yasaman S. Sefidgar, Woosuk Seo, Michael J. Tumminia, Daniella K. Villalba, Sheldon Cohen, Kasey G. Creswell, J. David Creswell, Afsaneh Doryab, Paula S. Nurius, Eve Riskin, Anind K. Dey, and Jennifer Mankoff. Leveraging Collaborative-Filtering for Personalized Behavior Modeling: A Case Study of Depression Detection among College Students. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(1):1–27, March 2021.

[211] Xuhai Xu, Jun Gong, Carolina Brum, Lilian Liang, Bongsoo Suh, Shivam Kumar Gupta, Yash Agarwal, Laurence Lindsey, Runchang Kang, Behrooz Shahsavari, Tu Nguyen, Heriberto Nieto, Scott E Hudson, Charlie Maalouf, Jax Seyed Mousavi, and Gierad Laput. Enabling Hand Gesture Customization on Wrist-Worn Devices. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 1–19, New Orleans LA USA, April 2022. ACM.

[212] Xuhai Xu, Xin Liu, Han Zhang, Weichen Wang, Subgiya Nepal, Kevin S Kuehn, Jeremy Huckins, Margaret E Morris, Paula S Nurius, Eve A Riskin, Shwetak Patel, Tim Althoff, Andrew Campell, Anind K Dey, and Jennifer Mankoff. GLOBEM: Cross-Dataset Generalization of Longitudinal Human Behavior Modeling. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(4):32, 2022.

[213] Xuhai Xu, Ebrahim Nemati, Korosh Vatanparvar, Viswam Nathan, Tousif Ahmed, Md Mahbubur Rahman, Daniel McCaffrey, Jilong Kuang, and Jun Alex Gao. Listen2Cough: Leveraging End-to-End Deep Learning Cough Detection Model to Enhance Lung Health Assessment Using Passively Sensed Audio. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(1):1–22, March 2021.

[214] Xuhai Xu, Haitian Shi, Xin Yi, Wenjia Liu, Yukang Yan, Yuanchun Shi, Alex Mariakakis, Jennifer Mankoff, and Anind K Dey. EarBuddy: Enabling On-Face Interaction via Wireless Earbuds. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, page 14, Honolulu HI USA, 2020. ACM.

[215] Xuhai Xu, Anna Yu, Tanya Jonker, Kashyap Todi, Feiyu Lu, Xun Qian, João Marcelo Evangelista Belo, Tianyi Wang, Michelle Li, Aran Mun, Te-Yen Wu, Junxiao Shen, Ting Zhang, Narine Kokhlikyan, Fulton Wang, Paul Sorenson, Sophie Kim, and Hrvoje Benko. XAIR: A Framework of Explainable AI in Everyday Augmented Reality. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2023.

[216] Xuhai Xu, Chun Yu, Yuntao Wang, and Yuanchun Shi. Recognizing Unintentional Touch on Interactive Tabletop. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1):1–24, March 2020.

[217] Xuhai Xu, Han Zhang, Yasaman Sefidgar, Yiyi Ren, Xin Liu, Woosuk Seo, Jennifer Brown, Kevin Kuehn, Mike Merrill, Paula Nurius, Shwetak Patel, Tim Althoff, Margaret E Morris, Eve Riskin, Jennifer Mankoff, and Anind K Dey. GLOBEM Dataset: Multi-Year Datasets for Longitudinal Human Behavior Modeling Generalization. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, page 18, 2022.

[218] Xuhai Xu, Tianyuan Zou, Han Xiao, Yanzhang Li, Ruolin Wang, Tianyi Yuan, Yuntao Wang, Yuanchun Shi, Jennifer Mankoff, and Anind K Dey. TypeOut: Leveraging Just-in-Time Self-Affirmation for Smartphone Overuse Reduction. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 1–17, New Orleans LA USA, April 2022. ACM.

[219] Makoto Yamada, Takeuchi Koh, Tomoharu Iwata, John Shawe-Taylor, and Samuel Kaski. localized lasso for high-dimensional regression. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 325–333, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR.

[220] Yi Yao and Gianfranco Doretto. Boosting for transfer learning with multiple sources. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 1855–1862. IEEE, 2010.

[221] Lucy Yardley, Bonnie J Spring, Heleen Riper, Leanne G Morrison, David H Crane, Kristina Curtis, Gina C Merchant, Felix Naughton, and Ann Blandford. Understanding and promoting effective engagement with digital behavior change interventions. *American journal of preventive medicine*, 51(5):833–842, 2016.

[222] Tal Yarkoni and Jacob Westfall. Choosing prediction over explanation in psychology: Lessons from machine learning. *Perspectives on Psychological Science*, 12(6):1100–1122, 2017.

[223] Xiaoxin Yin and Jiawei Han. Cpar: Classification based on predictive association rules. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, pages 331–335. SIAM, 2003.

[224] Kimberly S Young. Cognitive behavior therapy with internet addicts: treatment outcomes and implications. *Cyberpsychology & behavior*, 10(5):671–679, 2007.

[225] Kimberly S Young. Cbt-ia: The first treatment model for internet addiction. *Journal of Cognitive Psychotherapy*, 25(4):304–312, 2011.

[226] Han Zhang, Margaret E. Morris, Paula S. Nurius, Kelly Mack, Jennifer Brown, Kevin S. Kuehn, Yasaman S. Sefidgar, Xuhai Xu, Eve A. Riskin, Anind K. Dey, and Jennifer Mankoff. Impact of Online Learning in the Context of COVID-19 on Undergraduates with Disabilities and Mental Health Concerns. *ACM Transactions on Accessible Computing*, page 3538514, July 2022.

[227] Han Zhang, Paula Nurius, Yasaman Sefidgar, Margaret Morris, Sreenithi Balasubramanian, Jennifer Brown, Anind K. Dey, Kevin Kuehn, Eve Riskin, Xuhai Xu, and Jen Mankoff. How Does COVID-19 impact Students with Disabilities/Health Concerns? In *arXiv*. arXiv, May 2020. arXiv:2005.05438 [cs].

[228] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond Empirical Risk Minimization. *arXiv:1710.09412 [cs, stat]*, April 2018. arXiv: 1710.09412.

[229] Xu Zhang, Junghyun Kim, Qingwei Lin, Keunhak Lim, Shobhit O Kanaujia, Yong Xu, Kyle Jamieson, Aws Albarghouthi, Si Qin, Michael J Freedman, et al. Cross-dataset time series anomaly detection for cloud systems. In *2019 USENIX Annual Technical Conference (USENIX ATC 19)*, pages 1063–1076, 2019.

[230] Yi Zhang and Rui Wang. Cross-domain dependency parsing using a deep linguistic grammar. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 378–386, 2009.

[231] Yongli Zhang and Yuhong Yang. Cross-validation for selecting a model selection procedure. *Journal of Econometrics*, 187(1):95–112, 2015.

[232] Zijian Zheng and Geoffrey I Webb. Lazy learning of bayesian rules. *Machine Learning*, 41(1):53–84, 2000.

[233] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain Generalization in Vision: A Survey. *arXiv:2103.02503 [cs]*, July 2021. arXiv: 2103.02503.