## Title

Lanterns on Black Boxes: Illuminating Engagement, Assurance, Efficiency, and Decision-Making in Clinical AI



## ISBN

9798265476890

## Author

Levine, Lionel Marc

## Publication Date

2025-12-08

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

# Lanterns on Black Boxes:
# Illuminating Engagement, Assurance, Efficiency, and Decision-Making in Clinical AI

A dissertation submitted in partial satisfaction
of the requirements for the degree Doctor of Philosophy
in Computer Science

by

Lionel Marc Levine

2025

ABSTRACT OF THE DISSERTATION

Lanterns on Black Boxes:
Illuminating Engagement, Assurance, Efficiency,
and Decision-Making in Clinical AI

by

Lionel Marc Levine

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2025

Professor Majid Sarrafzadeh, Chair

This dissertation investigates how the vagaries and complexities of human activity influence, and in-turn are impacted by, the computational systems and algorithms designed to engage them. It advances a human-mediated design paradigm spanning: (i) systems for sensing and classifying *perceived episodes* of mental distress and triggering just-in-time support; (ii) assurance work that clarifies what we can (and cannot) infer about model validity and reframes evaluation around information-centric criteria; and (iii) sequence models that treat care as next-action prediction over structured clinical event streams.

**Methods/Procedures.** I built a multi-modal data pipeline (TASC) integrated into a user-oriented smartphone and smartwatch application ecosystem, along with an opportunistic experience-sampling workflow to move from static diagnoses toward episodic labeling and action policies. In the assurance track, I formalize limits of cross-model correlation and introduce an information-guided compression method—MI-to-Mid Distilled Compression (M2M-DC)—that couples block-level, mutual-information pruning with planes-aware inner slicing and brief staged distillation. In the decision-support track, I develop PRISM, a compact language model for structured EHR timelines, and PRISM-Consult, which routes early tokens in an encounter to domain-specialist PRISM models via a lightweight, calibrated router under strict latency and burden constraints.

**Findings.** TASC demonstrates an effective multi-modal fusion pipeline for detecting episodic stress, M2M-DC delivers strong accuracy–efficiency trade-offs across residual and inverted-residual families, reducing parameters and multiply–adds substantially while maintaining or improving accuracy with simple, hardware-friendly edits. In clinical modeling, PRISM specialists trained with a shared backbone and tokenization converge to low perplexities, supporting transferability without enlarging the base model. A calibrated router achieves near-perfect inclusion of the correct specialist(s), high inclusion of all required specialists, preserves safety for life-threatening domains, and reduces the average number of consulted experts per episode—demonstrating a viable path from human-in-the-loop data and assurance-driven efficiency to fast, specialist-routed clinical next-action prediction.

The dissertation of Lionel Marc Levine is approved.

Adnan Youssef Darwiche

Baharan Mirzasoleiman

Alexander Stehle Young

Majid Sarrafzadeh, Committee Chair

University of California, Los Angeles

2025

*To Jamie: my rock throughout this whole process. To my parents: the original Drs. Levine, and to Savanna: my greatest little buddy,*

# Contents

# List of Figures

# List of Tables

# Acknowledgments

## Acknowledgments

I am grateful to my advisors, collaborators, and study participants whose contributions made this work possible. I would like to singularly acknowledge Dr. Majid Sarrafzadeh, my principle investigator on all research undertaken at UCLA, and my doctoral advisor throughout the process. Dr. Sarrafzadeh's mentorship, support, and guidance throughout my tenure at UCLA has been singularly empowering, and I would not be here today without all his support.

The research in this dissertation includes co-authored and previously disseminated material. For transparency and compliance with UCLA filing requirements, I identify the originating publications (or submissions), briefly describe co-author contributions, credit the project director/PI as appropriate, and acknowledge funding and permissions below.

### Chapter-by-chapter notes

**Chapter 2. Systems for Sensing and Episodic Labeling of Mental Distress.** Adapted from: Levine, L.M. et al. (2020). Anxiety Detection Leveraging Mobile Passive Sensing. In: Alam, M.M., Hämäläinen, M., Mucchi, L., Niazi, I.K., Le Moullec, Y. (eds) Body Area Networks. Smart IoT and Big Data for Intelligent Health. BODYNETS 2020. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 330. Springer, Cham. https://doi.org/10.1007/978-3-030-64991-$3_15$.

Co-authors: Migyeong Gwak, Kimmo Kärkkäinen, Shayan Fazeli, Bita Zadeh, Tara Peris,

Alexander S. Young. Project Director/PI: Majid Sarrafzadeh.

Author contributions: L. Levine—conception, system design, methodology, software, investigation, writing; Migyeong Gwak, Kimmo Kärkkäinen—data engineering, software; Tara Peris, Alexander S. Young—study operations/IRB; Shayan Fazeli—analysis/visualization; Majid Sarrafzadeh—supervision.

Funding: MITRE Corporate Grant.

**Chapter 3. Labeling Intractable Classes in Mental Health.** Adapted from: *S. Fazeli et al., "A Self-supervised Framework for Improved Data-Driven Monitoring of Stress via Multi-Modal Passive Sensing," 2023 IEEE International Conference on Digital Health (ICDH), Chicago, IL, USA, 2023, pp. 177-183, doi: 10.1109/ICDH60066.2023.00033..*

Co-authors: Shayan Fazeli, Mehrab Beikzadeh, Baharan Mirzasoleiman, Bita Zadeh, Tara Peris. Project Director/PI: Majid Sarrafzadeh.

Contributions: S. Fazeli, L. Levine—conceptualization, methodology, experiments, writing; Bita Zadeh, Tara Peris—surveys/EMA instrumentation; S. Fazeli—statistical analysis; B. Mirzasoleiman, M. Sarrafzadeh—supervision.

Funding: MITRE Corporate Grant.

**Chapter 4. AI Assurance: Limits of Cross-model Correlation and Information-centric Evaluation.** Based on: *Exploring Cross-model Neuronal Correlations in the Context of Predicting Model Performance and Generalizability - Haniyeh Ehsani Oskouie, Lionel Levine et al., arXiv:2408.08448, January 2025 (https://arxiv.org/abs/2408.08448).*

Co-authors: Haniyeh Ehsani Oskouie, Sajjad Ghiasvand. Project Director/PI: Majid Sarrafzadeh.

Contributions: Haniyeh Ehsani Oskouie, L. Levine—theory, analysis, writing; Sajjad Ghiasvand—proof assistance/experiments; Majid Sarrafzadeh—supervision.

Funding: N/A

**Chapter 5. MI-to-Mid Distilled Compression (M2M-DC).** Adapted from: *MI-to-Mid Distilled Compression (M2M-DC): An Hybrid-Information-Guided-Block Pruning with Progressive Inner Slicing Approach to Model Compression - Lionel Levine, et al., arXiv:2511.06842, November 2025 (https://arxiv.org/abs/2511.06842).*

Co-authors: Haniyeh Ehsani Oskouie, Sajjad Ghiasvand. Project Director/PI: Majid Sarrafzadeh.

Contributions: L. Levine—method design, implementation, experiments, writing; Haniyeh Ehsani Oskouie, Sajjad Ghiasvand - Methodology Development, Writing; Majid Sarrafzadeh—supervision.

Funding: N/A

**Chapter 6. PRISM and PRISM-Consult: Next-Action Prediction over Clinical Event Streams.** Adapted from: *PRISM: A Transformer-based Language Model of Structured Clinical Event Data – Lionel Levine et al., arXiv:2506.11082, June 2025 (https://arxiv.org/abs/2506.11082).*

Co-authors: John Santerre, Alex S. Young, T. Barry Levine, Francis Campion. Project Director/PI: Majid Sarrafzadeh.

Contributions: L. Levine—model design, tokenization, routing policy, experiments, writing, data engineering; Alex S. Young, T. Barry Levine, Francis Campion—EHR curation; Alex S. Young, T. Barry Levine, Francis Campion—clinical validation; Majid Sarrafzadeh, John Santerre—supervision.

Funding: N/A

And also adapted from: *PRISM-Consult: A Panel-of-Experts Architecture for Clinician-Aligned Diagnosis - Lionel Levine, et al., arXiv:2510.01114, October 2025 (https://arxiv.org/abs/2510.01114).*

Co-authors: John Santerre, Alex S. Young, T. Barry Levine, Francis Campion. Project Director/PI: Majid Sarrafzadeh.

Contributions: L. Levine—model design, tokenization, routing policy, experiments, writing, data engineering; Alex S. Young, T. Barry Levine, Francis Campion—EHR curation; Alex S. Young, T. Barry Levine, Francis Campion—clinical validation; Majid Sarrafzadeh, John Santerre—supervision.

Funding: N/A

## Funding acknowledgments

This material is based upon work in-part supported by the MITRE Corporation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of MITRE.

## Ethics and data use

Human-subjects components were reviewed/approved by UCLA's Office of Research Administration Insitutional Review Board. All data were de-identified and analyzed in accordance with institutional guidelines. We thank UCLA Medical for collaboration and secure data access.

## General thanks

In addition to my previously mentioned gratitude to my PI, Dr. Majid Sarrafzadeh, I am indebted to my committee—Adnan Darwiche, Baharan Mirzasoleiman, Alexander S. Young—for their guidance and support throughout this process, to my research colleagues both inside and outside of UCLA, who also include Tara Peris and John Santerre. To my colleagues and friends at MITRE, including Dr. Linda Desens who co-led the MITRE effort, and Dr. Sybil Russell who directly supported our work across multiple years. I would like to also express my gratitude to my lab colleagues, including Babak Moatamed, Migyeong Gwak, Mohammad Kachuee, Sajad Darabi, Shayan Fazeli, Ghazaal Ershadi, Tyler Davis, Kimmo Kärkkäinen, Orpaz Goldstein, Davina Zamanzadeh, Mehrab Beikzadeh, and Haniyeh Ehsani Oskouie for their insight, collaboration, and camaraderie over the years. Finally, and most critically to my family, including my wife, daughter, parents, and sister for their unwavering support throughout.

# Vita

## Education

Ph.D. (Candidate), Computer Science, University of California, Los Angeles (Dissertation Defense: Dec 2025)

M.S., Computer Science, University of California, Los Angeles, 2020

M.S., Healthcare Policy & Management, Carnegie Mellon University, 2012

B.S., Economics; B.A., Philosophy, University of Pittsburgh, 2010

## Research Interests

Clinical sequence modeling and next-action prediction; model assurance, calibration, and reliability; information-centric model compression; passive sensing and opportunistic experience sampling; health informatics and FHIR-based data integration; responsible and agentic AI in healthcare.

## Selected Publications

Levine, L., *et al.* "MI-to-Mid Distilled Compression (M2M-DC): A Hybrid-Information-Guided Block Pruning with Progressive Inner Slicing." arXiv:2511.06842, 2025.

Levine, L., *et al.* "PRISM-Consult: A Panel-of-Experts Architecture for Clinician-Aligned Diagnosis." arXiv:2510.01114, 2025.

Levine, L., *et al.* "PRISM: A Transformer-based Language Model of Structured Clinical Event

Data." arXiv:2506.11082, 2025.

Ehsani Oskouie, H., Levine, L., *et al.* "Exploring Cross-model Neuronal Correlations in Predicting Model Performance and Generalizability." arXiv:2408.08448, 2025.

Hatamian, A., Levine, L., *et al.* "Dataset Statistical Effect Size and Model Performance/Data Sufficiency." arXiv:2501.02673, 2025.

Fazeli, S., Levine, L., *et al.* "Self-supervised Framework for Monitoring Stress via Multi-modal Passive Sensing." arXiv:2303.14267, 2023.

Levine, L., *et al.* "Anxiety Detection Leveraging Mobile Passive Sensing." arXiv:2008.03810, 2020.

# Chapter 1

# Introduction

The work we do at the eHealth Research and Analytics Lab at UCLA has, from its outset, come with a dual mandate: firstly, to build novel computational systems and analytics in the healthcare domain, and secondly, that such systems must have practical end-user applications.

This has had a profound influence on our theoretical work, which while ultimately driven to advance the domains systems design, artificial intelligence, and data science, has been anchored by a need to integrate such systems in end-user applications, and the necessary practicalities such anchoring entails. The human element of these systems impacts these projects in notable and often vexing ways, whether it be the inherent challenges and costs to obtaining accurate data, to the needs of making the systems and models more responsive to user-needs, to developing mechanisms to increase trust and assurance in model-use, has had a profound influence on all aspects of our system design.

Throughout the projects I've contributed to in the lab, we've had to adapt our systems to be more responsive to end-user engagement, and addressed the challenges of curating accurate labeling under circumstances when omission or even adversarial labeling conditions hold true. We've had to adapt our model development to anticipate costs and efficacy of models at the outset, and explore novel mechanisms to improve the quality of engagement with AI models. This has necessitated experimentation and innovation in both the software and systems design elements of our

products, as well as novel analytics and modeling efforts. In so doing we have contributed novel and substantive insights to these respective domains.

For this dissertation I present and summarize a body of work I have contributed to in the course of my studies. In detailing the technical challenges, research and experimental approach, results and discussions, I intend to draw special focus to the ways in which a human-centric approach to our design has profoundly shaped the resulting work.

This dissertation includes a detailed review of the following work:

- Building healthcare applications to monitor individual's mental health and provide just-in-time therapeutic interventions. In addition to detailing this work it's technical specifications, experimental approach, and results, I will also explore the challenges from both a systems' design and analytic perspective of building out and integrating a multi-modal data collection framework, to attempting to engage users in platforms that while beneficial to them, aren't necessarily engaging, as well as the challenges around generating discrete labels for nebulous and intractable domains like mental health, where achieving a ground-truth label is almost impossible.

- Developing novel learning frameworks to better attend to scenarios of limited and inaccurate labeling of data. I will also explore the particular challenges around label collection in the mental health space. These challenges, while not unique, are representative of a broader challenge of feature-data being relatively easily obtained, while label data is costly, infrequently input, and typically inaccurate. I will present a theoretical deep-learning framework we developed to address these challenges, as well as future possible steps in this domain.

- Composing analytic tools to better understand ML model requirements and the anticipated performance. Given the increasing centrality of AI models, and the growing costs to train increasingly complex model architectures, I will introduce ongoing work and current experimental results we've developed to better assess the objective performance of models, as well as better anticipate data sufficiency prior to engaging in model performance.

- Introducing an information-guided approach to model compression. I will detail a residual-safe recipe that couples block-level mutual-information pruning with planes-aware inner slicing and a brief staged knowledge-distillation repair, presenting method, experimental setup, and results across residual and inverted-residual families. I will show that we can remove substantial compute and parameters while preserving—or even improving—accuracy.

- Constructing a novel foundation-model framing of clinical care as next-action prediction over structured EHR event streams. I will introduce the baseline model and its routed, multi-specialist extension; describe tokenization, training protocol, routing policy, and datasets; and present results showing smooth specialist convergence with low perplexities and a lightweight router that preserves safety while reducing consulted experts per episode. I will conclude with implications for decision support, calibration, and operational deployment.

While the breadth of this work spans everything from systems design, to data collection, to model architecture, and novel deep learning approaches, the common thread through each is the human element driving its design. Whether as an attempt to hold up a mirror to elusive human behavioral patterns, or in reaction to human impulses, this work seeks to illuminate this fascinating symbiotic interplay of man and machine, to enlighten the computational black-boxes we have increasingly intertwined into our daily-lives.

# Chapter 2

# Systems Design: Multimodal Mental Health Monitoring

### 2.0.1 Background

The rising epidemic of mental health disorders, worsened by the recent COVID-19 pandemic, speaks to the growing need for effective and timely management of mental health.

The pandemic's acute effects led both to an increase in the need for mental health services, while concurrently, given the circumstances surrounding the outbreak, limited access to traditional modalities of care. This necessitated the explosion in the usage of alternative mechanisms to deliver mental health services, mainly through remote formats [114].

As a consequence, in spite of increased barriers to access, ranging from financial constraints and insurance-related inhibitors to the limited number of therapists and psychologists available to provide mental health services, unprecedented levels of funding have gone into programs to address mental health issues among the general public. For instance, in 2020 alone, the United States government spent around $280$ billion dollars on mental health services [52].

Yet in spite of truly monumental spending levels, in the US, over $27$ million people experiencing a mental illness in 2022 did not receive any treatment, comprising $56\%$ of adults with a mental disorder. Additionally, $11.1\%$ of adults suffering from a mental illness were uninsured [80].

This has served as a catalyst for a wave of experimentation exploring improved modalities for the detection, monitoring, and treatment of mental health. Mobile technologies, and their associated analytics, have been at the forefront of these efforts.

### 2.0.2 Stress and Anxiety Disorders

Mental Health encompasses a broad spectrum of conditions and associated disorders, and approaches to diagnoses, treatment, and assessment vary widely. Our work touches on several specific disorders, and we will detail the unique challenges and approaches taken to model systems to their particular challenges.

Stress and anxiety-related disorders are common mental health challenges. Such disorders can have significant negative impacts on people's lives, including higher chances of depression and suicide as well as associated comorbidities with physical health issues [28].

Stress, commonly defined as "physical, mental, or emotional strain or tension," is a widespread problem with numerous potential causes. According to the American Institute of Stress, $73\%$ of people suffer from acute bouts of stress to a degree of magnitude that impacts their mental well-being. Incidents of Anxiety often manifest similarly to stress, however, it is notable that it is not always immediately tied to a specific triggering or inciting event and may take longer to resolve.

All told, both stress and anxiety problems are very common, to the extent that most adults have been affected by at least one anxiety-related disorder [91, 106]. Anxiety-related disorders can have a significant negative impact on the quality of life, leading to other mental health disorders such as depression, as well as causing physical health problems [28].

Within the spectrum of mental health disorders, Anxiety disorders are the most common class of psychiatric problems affecting both children and adults [16, 21, 81], with up to one in three people in the US meeting full diagnostic criteria by early adulthood [113, 44]. This manifests in the form of roughly to 7 to 9% of the population in the US suffering from a specific phobia, 7% from social anxiety disorder, and 2 to 3% each from panic disorder, agoraphobia, generalized anxiety disorder, and separation anxiety disorder [9]. Individuals with anxiety disorders contend

with substantial distress and impairment. They are at heightened risk for a host of negative long-term outcomes including depression, substance abuse, educational underachievement, and poor physical health [89, 12, 124].

The optimal method for the prevention or care of mental illness is early identification, diagnosis, and proactive treatment [117]. Time-sensitive intervention is therefore crucial for preventing conditions from becoming chronic and debilitating. However, traditional methods of psychiatric assessment, including clinical interviews and self-reports, are limited in their ability to provide just-in-time interventions as well as early identification. They depend heavily on retrospective summaries collected in clinical settings, conditions that often result in reporting biases, inaccurate recall, or late and ineffectual treatment.

Additionally, anxiety disorders are, for the most part, vastly overlooked and under-treated in the community; only 15-30% of anxious individuals in the community receive treatment of any kind. Recent research has found strikingly high levels of anxiety among college-age youth. Indeed, 58.4% of college-aged youth report feeling "overwhelmed by anxiety" [8]. Several other recent studies document the high proportion of college students meeting full diagnostic criteria for an anxiety disorder [18]. At the same time, young adults are particularly overlooked within the health care system, with rates of screening, identification, and referral falling below those of either children or full-fledged adults [124]. Unfortunately, in many cases, these issues remain inadequately treated due to challenges ranging from lack of viable access to therapeutic services to associated stigmas with utilization [91, 106]. However, even when an individual decides to seek psychotherapeutic help to alleviate these problems, challenges persist in the diagnosis and effective treatment of their disorder. At the inception of care, the steps to diagnose and monitor often include clinical evaluation and comparing personalized symptoms to standardized criteria, for example, the Diagnostic and Statistical Manual of Mental Disorders (DSM-5), which is commonly used for this matter. Researchers continue to study and improve the practicality and accuracy of guidelines such as DSM-5[14, 134], but there are challenges in converting aggregated and generalized diagnostic criteria, down to episodic-level incidents of stress and anxiety.

Given this landscape, there remains a pressing need for tools that improve early identification of anxiety symptoms, provide users with the platforms to monitor their activities, raise awareness of factors impacting on their wellbeing, and provide a mechanism for intervention should an anxiety episode escalate for all demographics, but most urgently among adolescents.

### 2.0.3 Overview of Efforts

In the course of my tenure at the ER Lab, we have pioneered several different platforms exploring potential avenues for the therapeutic management of mental health, leveraging diverse mobile technology solutions.

This has included an Android Smartphone-based system designed to monitor for signs of an acute mental health crisis, called eWellness, a Smartphone-based platform for the monitoring chronic stress, called VetThrive, and a Smartwatch-based platform for tracking physiological indicators of stress, called TASC (Technology-Assisted Stress Control).

I will detail the technical specifications of these platforms, including challenges posed in the design and implementation of them. I will also focus on the vexing challenge of developing accurate discrete labeling for an inherently amorphic and intractable state like one's mental health state.

## 2.1 Remote Monitoring Technologies

There has been considerable progress in recent decades around the capabilities and efficacy of personal digital devices, including smartwatches, smartphones, and wearable devices. This fact has made such devices attract a lot of research and commercial attention, employing them for various monitoring objectives [2, 112].

These monitoring approaches focus primarily on fitness and health-related aspects, resulting in a large body of research and countless commercialized applications. Examples include tracking athletes' training, detecting falls for the elderly, tracking post-surgery therapeutic and rehabilitation

exercises, and posture correction [32, 116, 42, 123]. While the central focus of health monitoring applications has undoubtedly been on physical health, a wide range of research works has focused on understanding the relationship between observations obtained leveraging digital devices and some aspects of individuals' mental health status. It is noteworthy that a primary goal in designing smart and automated approaches for mental health monitoring has to do with proposing meaningful passive-sensing tools so that informative observations regarding health status can be made by eliminating or diminishing the need to interfere with users' daily activities or request repeated active interactions.

As a remote mental health monitoring task, social anxiety was studied in the previous literature, and it was shown that analyzing trajectories obtained via smartphone location services can paint a comprehensive picture concerning individuals' proneness to it [17, 24, 56, 39]. Smartphones have also been helpful in developing an understanding of anxiety [70].

Another choice of hardware for gathering data pertinent to health data is application-specific wearable sensors. For instance, wearable electrocardiogram (ECG) sensors were used to recognize perceived anxiety via pattern recognition [64].

Smartwatches have a unique position amongst the wide range of various commonly used digital devices. They are in close contact with the skin and, given their attachment user's wrist, which is a distal point of a major appendage, make it possible to obtain most measurements (e.g., activity) at higher accuracy, as well as enabling additional measurements such as heart-rate or pulse oximeter. In case of the need for brief questions, interactions, or Ecological Momentary Assessments (EMAs), smartwatches can also be used to issue messages and acquire responses and entries by the user [2, 112, 58]. Additionally, smartwatches are prevalent, and relying on them as the hardware for health applications provides a better alternative in most cases to application-specific wearable devices in terms of cost, comfort, and user-friendliness.

The most obvious approach to doing so leverages biometric data, extracted from wearable sensors embedded in smart devices, that measure a physiological stress response. However, while such data is incredibly valuable, and notably, sensing devices have become increasingly sophisti-

cated at monitoring physiological stress, the resulting analyses are incomplete at best. This owes to the fact that from the standpoint of straightforward correlative analytics, it is known that there is not a direct monotonic correlation between the emotional perception of stress an individual may feel and the manifestation of the underlying physiological stress response. A meta-analysis in the social stress domains, for instance, has recently shown that merely $25\%$ of studies in the domain demonstrated a significant correlation between physiological stress and perceived emotional stress [19].

Given that self-reports of perceived stress often do not contain information on the physiological stress response, understanding the complex relationship between the two becomes a crucial matter[19]. It is also plausible to assume that such complexity also arises from various other confounding factors (e.g., demographics, occupation, and other mental health disorders such as attention-deficit hyperactivity disorder (ADHD) can influence how prone someone is to stress).

This discrepancy has meaningful impacts on the utility of passive detection of stress based largely on physiological indicators. While sensors may be returning accurate readings on physiological stress, if they do not align with the user's own perceptions of stress, notably if they fail to properly account for moments when a user feels acute emotional distress, then it will demotivate further engagement with a mental health platform.

### 2.1.1 Conscious Perceptions of Stress

While smart devices, like smartwatches, have demonstrated the technical capacity to monitor physiological stress to varying degrees of accuracy, they demonstrate only a partial picture of a wearer's overall mental health. This stems from the intrinsically complex relationship between emotional perceptions of stress and the underlying physiological manifestations of stress response. There is growing evidence that there are notable differences in how individuals perceive their stress levels and the actual physiological manifestation of a stress response. For instance, a meta-analysis found that a significant correlation between physiological stress and perceived emotional stress was found in only $25\%$ of social stress studies [20]. There are a number of factors that are presumed

to contribute to this low correlation. Foremost among them is that, somewhat counter-intuitively, a direct linear relationship between these two stress profiles is not present but instead is influenced by many additional factors. A detailed study concluded that self-reports of perceived stress did not provide useful information about physiological stress responses [92]. Differing populations may also exhibit different correspondence patterns that population-level models fail to account for (e.g., individuals with ADHD, those who experience chronic stress, those with family histories of substance abuse, and so on.) [20]. It is also theorized that differing response patterns may impact how stress manifests physiologically. For instance, differing degrees of cognitive regulation of negative emotions may result in individuals being able to adapt to stress differently [65]. Previous work demonstrating a higher correspondence between self-reported anxiety and physiological arousal in women when compared to men is hypothesized to be explained by men showing emotion-suppressing coping strategies to a greater extent than women [65]. Finally, it has been observed that positive emotions are more strongly associated with physiological responses than negative emotions. This may be due to their social acceptability and disinclination to control or dampen a physiological response when compared to negative ones [65]. Accurate and consistent measurement of emotional stress is a cumbersome task to carry out, and it comprises the main challenge in effectively mapping the aforementioned stress profiles [92]. While multiple scales have been developed to assess differing quantifiable measures of mental health, there is no clinically validated gold standard for assessing emotional well-being accurately and consistently, both across a population and across individuals over time. This stems from issues with an individual's recall and perception of stress, particularly if data is collected after the fact, their evolving willingness to be forthright and transparent regarding emotional well-being, and differences in the subjective assessment of stress across individuals (what constitutes a *moderate* level of stress may differ dramatically across individuals) [92].

Therefore, even as the pandemic, and associated restrictions on in-person activities, have subsided, the gaps it revealed in traditional in-person-based therapeutic services persist, and demand for remote solutions remains high. Concurrently, given the high costs associated with mental health

services, and their associated scarcity, optimizing resource allocation, and increasing the efficiency of treatments are of paramount importance.

While much of the focus of remote mental health services has been around the use of video conferencing, instant messaging, and other modes of communication to facilitate interactions between therapists and patients, the use of mHealth technology and passive monitoring has the potential to be equally impactful at addressing barriers to care and gaps in monitoring. Furthermore, by leveraging personal digital devices equipped with numerous sensors that are capable of monitoring many aspects of an individual's physiology and lifestyle (e.g., heart rate, activity level), remote health monitoring provides a novel pathway to not only monitor existing indicators of mental health, but also to improve upon our understanding mental health disorders and their impacts on one's life.

It is critical, however to define more precisely what we are endeavoring to monitor, and ultimately what our programmatic objective is. For while there is an obvious and superficially attractive basis for analysis (The difficulty in rendering effective in-person diagnosis and care coupled with the explosion of remote monitoring tools), effectively pairing data and methodology with a useful label to monitor, is a suprisingly challenging exercise.

### 2.1.2 The Disconnect Between Mental Health Classification and Remote Monitoring

Ultimately our research found that classifying Mental Health, is an intrinsically nebulous challenge.

For one thing, unlike physiological diseases or conditions, that are typically tied to clear and discrete biomarkers (e.g., the presence of a biological compound in a test result), Mental Health diagnoses typically do not manifest in distinct and clinically verifiable ways. Diagnosis is typically the result of expert opinion, or increasingly, through the employment of questionnaires and standardized exams, designed to standardize the diagnosis and provision of care.

There are even efforts underway to establish a gold-standard for diagnosis by pegging it to bio-

metric monitoring, however this is limited in its applicability owing to the need to collect intensive biometric data limited to certain observational windows.

Yet even if we accept the veracity of labeling discrete disease states in this matter, it is not clear that a diagnosis would be meaningfully identifiable in the sorts of feature data we have ready access to.

Such diagnoses generally represent singular, binary defining classification, as mental illness is not typically viewed as a 'curable' disease that can be eradicated from an individual, but rather an ongoing condition to be chronically managed.

In contrast, data collection is typically done in a continuous time-series manner, and variations contained within are more reflective of immediate proximal emotional states rather than the presence of overall diagnostic conditions.

Further separating the nature of the data collection from the desired labeling, as we will note in detail below, is that the over-reliance on user self-reports is often the only way to obtain sufficiently large volumes of labeled chronological data regarding a person's mental health overtime, yet this data is often challenging to obtain, relying on consistent and proactive patient completion of questionnaires.

In contrast, behavioral and physiological feature data can increasingly be collected in seamless and automated fashion, (further details on this as well below). Yet, as the feature data most likely to be used in determining a classification is typically collected periodically, or episodically, it's an uncomfortable overlap to try and assign permanent diagnostic labels as a result of data collected in a limited and episodic fashion.

This is particularly manifest in mental health, where even the presence of a diagnosis does not conform to a consistent set of associated behaviors.

Unlike certain labels like sex, where there are associated fixed feature values (height), mental health indicators most typically manifest in terms of behavioral trends and behavioral trends do not remain consistent over time.

All individuals, irrespective of a formal diagnosis, will experience a wide spectrum of emotions

and associated behavioral and physiological metrics. If observed over a long-enough timeframe there may be statistically significant shifts in the distribution of observed features that could be accounted for by an underlying mental health condition, but even if such a trend could be definitively established it isn't clear that there is any real utility to such an exercise. It is unlikely that such models, regardless of real-world validation, would ever be relied upon as anything more than a tool to flag individuals for potential additional screening and formal diagnosis. Such individuals are likely aware of the presence of mental distress, even in the absence of a formal diagnosis, and the failure to act on it (due to barriers to access or an unwillingness to engage with mental healthcare professionals) will not likely be resolved by the classification.

In our own journey on this topic, we therefore evolved in two notable ways: 1. We moved away from an effort to accurately classify Mental Health conditions, instead focusing on trying to classify distinct periods of mental distress irrespective of underlying condition. 2. We ultimately side-stepped the question of a fully-accurate measure of acute mental distress, to instead attempt to predict an individual's perceptions of their own mental distress. This allowed us to rely more directly on user-self-labeling as issues with their perceptual disconnect from underlying physiological states is rendered irrelevant, and by aligning labeling with a user's own perceived experiences it further validates the reliability of the classifier.

### 2.1.3 Continuing Challenges in Episodic Distress Classification and Actionability

Even after reframing away from static diagnostic labels toward classifying perceived *episodes* of mental distress, several challenges persist at the levels of time-series design, label semantics, and the practical collection of reliable ground truth. As noted earlier, our data streams are inherently temporal and continuous, while mental-health constructs remain amorphous and difficult to anchor to a single gold standard; this mismatch is not eliminated by moving from diagnoses to episodes.

**Time-series framing and windowing ("blocking")**

Episodic classification presupposes a windowed view of the stream. Choosing the window length ($w$), stride ($s$), and anchoring strategy (event-anchored, symptom-anchored, or sliding) is consequential:

- **Window length and overlap.** Short windows increase temporal precision but reduce statistical sufficiency; long windows smooth variance but risk blending pre-, during-, and post-episode dynamics. Overlap can improve recall for short, sharp episodes, but may inflate apparent support via near-duplicate positives unless carefully de-duplicated.

- **Anchoring.** EMA-anchored windows (centered on a self-report) reduce label drift but may bias models toward moments when users choose to respond; sliding windows better reflect the deployment reality of continuous monitoring yet require a policy to attribute labels to windows without explicit anchors.

- **Context padding.** Many autonomic changes precede subjective distress. Including symmetric (or causal) padding around the putative episode improves capture of precursors but introduces leakage when episodes cluster.

- **Within-subject vs. across-subject blocking.** Given heterogeneity, baselining within subject (and learning deviations from personal norms) is often more faithful than across-subject blocking; however, deployment typically requires both, which raises normalization and drift questions (seasonality, routines, medication changes).

**From label to action: mapping predictions to interventions**

Episodic labels carry implicit operational meaning: they should trigger something. In practice, an action policy must specify *when*, *what*, and *how often* to intervene given model scores:

- **Thresholding and hysteresis.** Fixed thresholds can produce oscillatory user experience

(alert/no-alert) around the decision boundary. Introducing hysteresis and minimum inter-alert intervals stabilizes behavior and reduces alert fatigue.

- **Severity routing.** The label semantics should map onto tiered actions (self-guided breathing, social support outreach, clinician escalation), but this assumes the label captures *actionable* severity rather than only subjective arousal. Misalignment here degrades trust and engagement.

- **Attribution and accountability.** When labels are subjective, the intervention policy must respect user autonomy and privacy while still offering timely support. This tension is not resolved by accurate prediction alone.

**Label collection in practice: surveys vs. interviews**

Two remote strategies dominated our work: self-completed surveys/EMAs and periodic interviews. Each has distinct failure modes:

- **Self-completed surveys/EMAs.** We observed delayed and evasive reporting, reluctance to label *in situ* stressful events, and recall error when completing daily summaries (users approximated "typical days" or guessed at times). Because only completion (not accuracy) was verifiable, some participants appeared to satisfice—answering quickly and consistently rather than carefully—especially on sensitive items. These issues echo the broader absence of a validated population-wide gold standard and the role of recall/perception inaccuracy.

- **Interviews.** Structured interviews reduced some evasiveness over time as rapport developed with a consistent interviewer; participants became more candid and specific across sessions. Paradoxically, this improved honesty increased *within-subject* variance across time (later sessions contained richer, more sensitive disclosures), while *between-subject* differences remained large. Interviews thus produced higher-fidelity but sparser and time-varying labels, with cost and scalability trade-offs.

**Mitigations attempted and remaining limitations**

We experimented with methodological remedies—careful questionnaire design, internal consistency checks, mixed-method protocols, more regular sampling, and alternative labeling pathways (e.g., interviews)—but these only partially addressed the core issues. Even with cross-validation and increased sampling cadence, labeling inconsistency and bias persisted, and we lacked a principled way to verify correctness beyond completion. These shortcomings should be treated as design constraints for any future study rather than nuisances to be averaged away.

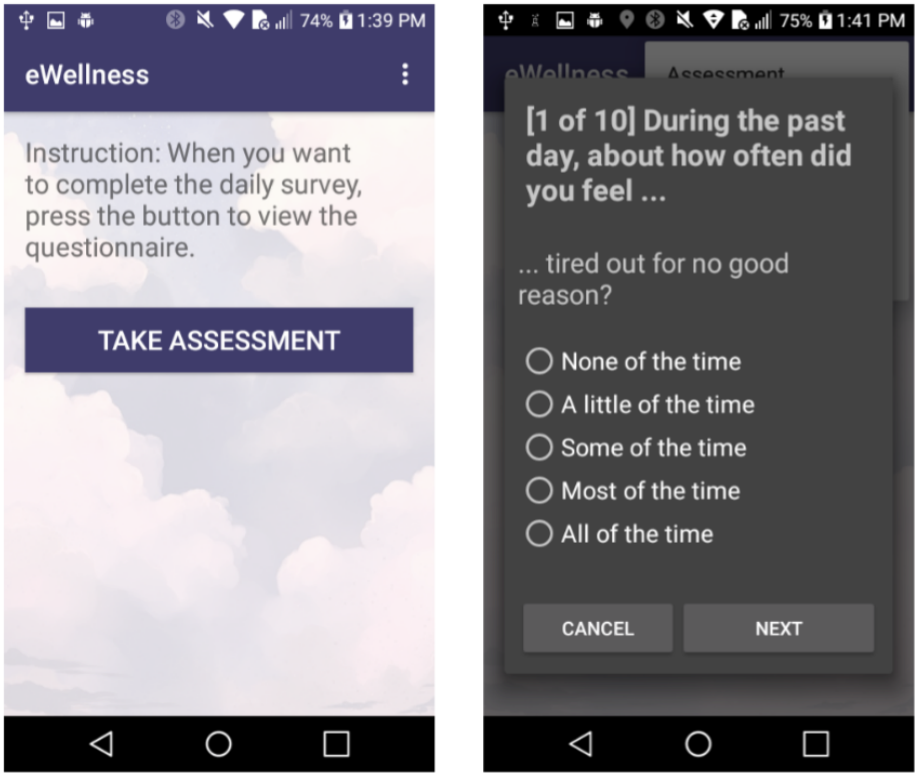
**Summary.** Reframing toward episodic distress improved construct alignment between data and label, yet left unresolved: (i) how to block time to faithfully isolate episodes, (ii) how to map labels to calibrated, non-fatiguing actions, and (iii) how to obtain labels that are timely, honest, and verifiable at scale. The remainder of this chapter and the next examines systems-level choices (EMAs, opportunistic sampling) and analytic choices (within-subject modeling, normalization) in light of these constraints.

## 2.2 Systems Design

### 2.2.1 Introduction

In the course of my tenure in the ER Lab, we constructed three distinct systems designed to engage and inform users about their mental health. While these systems were all distinct in their packaging, focus, modality of monitoring, and targeted end-user, overall they represented an evolution in platform sophistication, with elements of an preceding platform informing and often directly being incorporated into the platform itself.

What these systems all share is an autonomous data collecting mechanism, in which data needed to model mental health is collected in a discreet, continuous, and automated fashion, independent of the user. In-turn all platforms selectively engage the user in differing ways, with diverse mechanisms for in-taking user-feedback for labeling, and presenting the user with actionable in-




(a) Landing page. (b) Daily EMA questionnaire.

Figure 2.1: eWellness Android Mobile Application

formation.

### 2.2.2 eWellness

eWellness represented the first effort to build a system for the collection of mental-health related feature data, and associated labeling. The platform consisted of a native Android Mobile application, which wrote data to a cloud-hosted server.

#### eWellness Data Collection Hierarchy

The eWellness mobile application, developed for android devices, collected passive behavioral data derived from communications logs, embedded sensors, and user-logs and captured the following metrics:

- **Communication**: was monitored by tracking incoming and outgoing phone calls and text messages, including the duration of phone calls, the number of texts and phone calls, and unique individuals contacted. This metric notably did not assess the content of communications or the recipient of the communication, beyond establishing that a unique contact occurred.

- **Location**: was periodically sampled using GPS, network, and Wi-Fi detection. Prompts for a new location after moving 5 meters, up to once a minute. This metric leveraged the Google Fused Location API. The application did not track specific locations; instead, it kept a total distance traveled using the vectorized haversine distance function.

- **Ambient Sound**: was a numeric measure, designed to detect speech and communication above 50 decibels using the phone's microphone. It sampled audio every 5 minutes for 5 seconds. This metric did not capture the audio files of communications, instead merely documented the sound frequency and decibel level as numeric values.

- **Activity and Movements**: leveraged the device's accelerometer, gyroscope, and GPS tracking. Activity was sampled every 60 seconds. In order to determine stationary and moving activity-type, the application leveraged Google's Activity Recognition API.

- **Light**: detected light level associated with possibly being in an outdoor or indoor location, but could also be used as an indicator for whether or not a phone was left in an exposed open position, or tucked away in a pocket or equivalent storage median. This sensor was sampled every 6 seconds.

- **Phone use**: was a user-log monitoring the device's screen on-time.

From these raw values, we derived daily aggregated features from these metrics to infer both a user's sociability, and behavioral patterns. These were then used to learn a model for prediction of anxiety symptom severity over that timeframe. We obtained statistical characteristics, such as minimum, maximum, mean, standard deviation, the 25th, 50th, and 75th percentiles, of the numeric

values of noise exposure and the ambient luminescence. The number of activity transitions and duration of each physical activity per day also became a significant metric of identifying mentally distressed days.

### Labeling

The eWellness system also including a survey system where users could complete survey forms that were pre-coded. A push-notification mechanism was employed to remind users to complete these surveys at predetermined intervals. This mechanism was used to collect structured labeled data on users' mental health. Figure 2.1 demonstrates this UI.

The eWellness platform was a hybridized mix of open-sourced, and originally coded components. Much of the core data collection functionality was sourced from the Aware Framework, an open-source library of data collection elements to facilitate the collection and transmission of passively generated data.This framework was incorporated into our own native architecture to develop the eWellness Application.

### Server

Data from the study, both sensor feeds and usage-logs, along with user-generated EMA responses, were first encrypted, cached locally on the user's device, and then transmitted to a secure remote server, when a WiFi connection for the device was established, where it was stored in an encrypted scalable MySQL database. Figure 2.2 demonstrates the data collection hierarchy.

### Limiting Personally Identifiable Data Collection

Recognizing the potentially invasive nature of applications like this, data collection was carefully scoped to avoid the collection of Personally Identifiable Information (PII) that could link a particular user to a particular dataset. For example, when attempting to gauge sociability, the application logs the total number of phone calls made, total time on the phone, and the number of unique contacts called; the identities of specific callers were not tracked. This has the consequence of

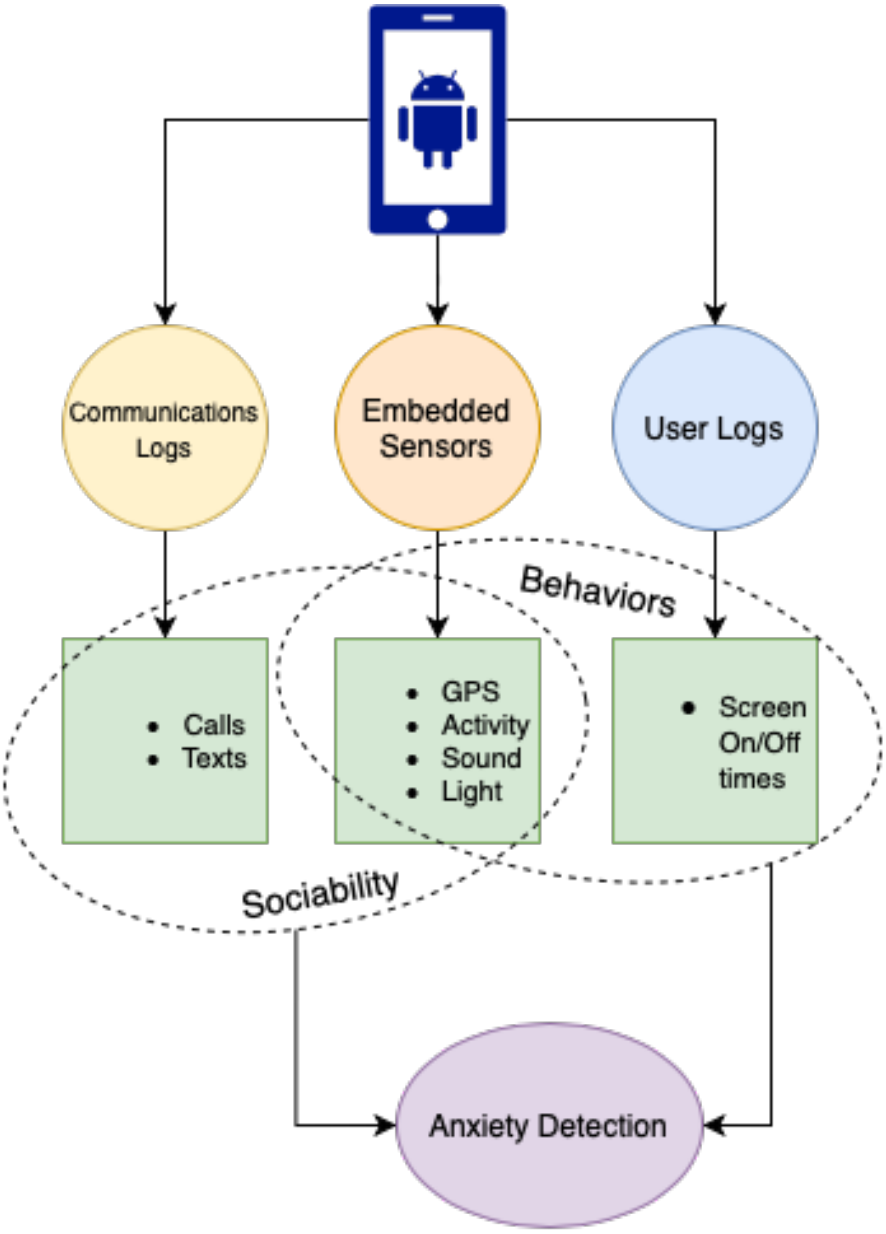



Figure 2.2: eWellness Data Collection hierarchy

introducing a degree of obscurity into an observed finding (e.g., as the application is unable to differentiate between calls to friends and calls to a customer-service hotline). At the same time, in the interest of both respecting privacy and ensuring the acceptability of the app, these efforts were felt to be necessary constraints on data collection.

### 2.2.3 VetThrive

VetThrive represented a direct evolution and maturation of the eWellness platform. VetThrive was a system for the remote monitoring of mental health symptoms, their fluctuation, and their attendant disruption to personal functioning. The VetThrive framework was similarly designed to capture a broad spectrum of remote monitoring, survey data acquisition, secure data transmission and management, data analytics, and visualization.

The primary component of VetThrive was a mobile application that facilitated the collection and transmission of data harvested from an array of sensors and usage logs from a user's smartphone. The data was collected passively, pre-processed, and transmitted through a secure gateway to the cloud, where it was securely stored, and indexed using a scalable database.

While VetThrive did inherit many root elements from eWellness, it represented a considerable evolution in the design. Foremost among them, the VetThrive application was expanded to include iOS devices as well as Android, so all major smartphone hardware was now compatible with the platform. The enhanced VetThrive application additionally included a front-end user-facing interactive dashboard, developed independently in React.js, hosted as a website, and embedded in both the iOS and Android applications leveraging embedded browsers. This dashboard included visualizations on key trends like number and duration of calls, and levels of physical activity.

Figure 2.3 Demonstrates the dashboard functionality.

#### Server

The backend was also considerably evolved. While the core feature-data collection framework remained the same, the nature of the intended end-user, Veterans in active care by the West Los


Figure 2.3: VetThrive User Dashboard

Angeles VA, necessitated additional privacy and security elements.

For one thing, our previous cloud-hosted environment was not cleared to host VA data, so the overall framework had to migrate to an AWS GovCloud solution. For another, the static data storage application, MySQL, was not approved for VA use, so the framework had to be rebuilt using OracleRDS, an approved relational database software application.

The raw feature data remained siloed in the OracleRDS DB to ensure the integrity of the data for data analysis purposes.

Additionally the front-end dashboard, which was a web-application, was now hosted on a separate cloud instance, in this case Google Cloud which also included firebase applications to handle a number of web application functionalities (e.g., accounts management) that were now required.

This architecture was complemented by an additional back-end analytic engine, capable of mapping observed metrics and exogenous data sources to a user's mental health state, leveraging adaptive statistical models, and advanced machine learning algorithms to generate summary findings for both the end-user dashboards, and associated mental health monitoring. The system thus represented a critical leap in capabilities towards a mature system designed to monitor overall mental health as well as acute crisis events in both a retrospective and predictive capacity.

### 2.2.4 Technology-Assisted Stress Control (Initial Version)

#### Introduction

Up till this point in our lab's, work we had focused our efforts exclusively on feature-data that could be extracted directly from a user's smartphone. We opted for this modality owing to its comparative ubiquity and ease of adoption.

However there was also no question that, particularly as our focus shifted towards detecting and monitoring stress and anxiety, relying solely on smartphone data, which is at-best incidental to stress, was limiting. Ultimately a more comprehensive look at the problem would require monitoring the physiological state of our users.

This coincided with a external partnership with Naval Special Warfare, where the request was

made to help design hardware systems to enable the biometric monitoring of active-duty service-members and their associated mental health.

This platform would ultimately represent the most mature and expansive build-out undertaken, validating a number of critical technical capabilities but also underwent considerable evolution based on user and stakeholder feedback.

### Hardware

In developing a hardware system for remote monitoring purposes, there were critical criteria we deemed crucial for the system to be useful in the real-world setting. In selecting the hardware components of our platform, we were mindful of the following:

**Ease of Use:** The proposed system needed to result in nothing more than minimal changes to individuals' daily lives. In the sense of the hardware component, to achieve this objective, we chose to limit the scope to a simple smartwatch that most people use everyday. While there are existing specialized sensors for monitoring mental health and especially for recognition of perceived stress using more specialized wearable components (e.g., ECG readings from chest [64]), nonetheless, one of our aims in this work is to study to what extent such systems would be necessary.

**Affordability:** The hardware components in a wearable health platform play the primary role in determining the overall costs, and the more affordable they are determines the levels of accessibility for the general public. There are advanced smartwatches offering sensory readings for less-available physiological signals, such as electrodermal and so on [47]. Nonetheless, such devices lie on the side of more expensive and less affordable devices, rendering it difficult to deploy them at scale. The smartwatch that we opted for was the Garmin Vivo Active 4s, whose price lies in the range of more affordable options for individuals. Additionally, it provided a comprehensive developer API for us to access the information to enable our analytics.

**Energy Efficiency:** The hardware and software components had to be in harmony to optimize the energy consumption associated with our data acquisition system. The primary *observation*s in this work are short-term episodes, therefore, we chose a low-frequency but continuous monitoring

of main signals. This reduces the computational load, while at the same time providing us with data crucial for accurate recognition of perceived stress.

**Sensors and Inter-device Compatibility:** Smartwatches often provide a range of sensory readings, from basic signals such as heart-rate to more advanced ones such as electrodermal response. In this work, we focused on the four modalities of physiological signals shown to be critically related to the body's arousal status. These are:

1. heart-rate and aggregate activity information,
2. normalized heart-rate variability levels,
3. pulse oximeter,
4. respiration rate.

To meet the goal of this research, the selected smartwatch needed to be equipped with standard tri-axial accelerometers, gyroscopes, wireless signal receiver, and power-efficient battery. Additionally, from a developer's perspective, we opted to work with watches that have well documented SDK and API supports.

Based on these requirements, the Garmin Vivoactive 4s was selected for this research. Although not as active as the developer communities of Samsung and Apple watches, Garmin does have documentation and support as well. the brand was also deemed more price-competitive to prospective users and was a preferred vendor for active-duty service members for whom this product was developed.

Although the system was built out with a specific commercial of the shelf (COTS) product, this data collection framework was deliberately built in a modular analytics pipeline with the intention of allowing a sort of *plug and play* functionality with future alternative hardware products, in addition to the fact that we focused on these widely available modalities, making it easy to deploy the same platform leveraging other smartwatches as well with minimal change.

### Platform Overview

We developed the Technology-Assisted Stress Control (TASC), a smartwatch and smartphone application that employs the embedded smartwatch sensors to collect feature data for this platform.

Given our platform's intimate link to the Garmin data ecosystem, our platform was constructed to work in parallel with Garmin's existing tech infrastructure.

This project consisted of 3 components:

1. a smartwatch app that connects with the end user directly and resided on the Garmin watch providing a direct interface with the user for soliciting wellness-checks and providing point of care interventions

2. an accompanying dashboard website that helps user to visualize and access their emotional health report with more detailed information as well as an enhanced set of communication, and intervention features.

3. a remote server that stores and processes data for the platform

### Data Collection Architecture

For our feature data collection, we relied on Garmin's existing architecture to collect and pre-process the sensor data, before connecting to its API to retrieve and incorporate the data into our stack.

The smartwatch and smartphone applications sync data and trigger activities via a Bluetooth connection between the devices, and separately, Garmin's own mobile application plays an integral role in syncing the watch with the server. Figure 1. demonstrates the data transfers. 1) Raw sensor values, along with user responses to Ecological Momentary Assessments (EMA) are recorded by the smartwatch device, cached on the device's memory, and synced to the mobile application via a bluetooth connection established by the user's smartphone when prompted by the Garmin Connect Mobile app.


Figure 2.4: Information Flow in TASC Model

2) The Garmin Connect mobile application packages smartwatch data and uploads it to Garmin servers for processing.

3) Launching the TASC mobile application, in-turn, triggers a cloud function in our TASC cloud to query the Garmin Health API for all updated information from the user. Our cloud server then processes the raw sensor values into a series of analytic outputs and then writes them to user-specific fields that the dashboard website reads when rendering.

4) If the server detects a high-stress event and triggers an EMA, the TASC mobile application will send an activation prompt to the smartwatch app, via Bluetooth, to trigger the EMA, which is captured and relayed back to the mobile app, and in-turn written to our server.

### Privacy

Offering a high-level of patient privacy is crucial in any health-related applications. For an increased level of privacy, we chose to focus our attention in recognition and monitoring of stress only to the physiological fluctuations and did not condition our predictions on users' background information. We did not gather patient identifying information at recruitment as well. While it is intuitively clear that doing so causes us to lose valuable information in understanding stress, given the trade-off of leveraging such data and providing highest levels of privacy, we chose to go with the latter and work with anonymized data.

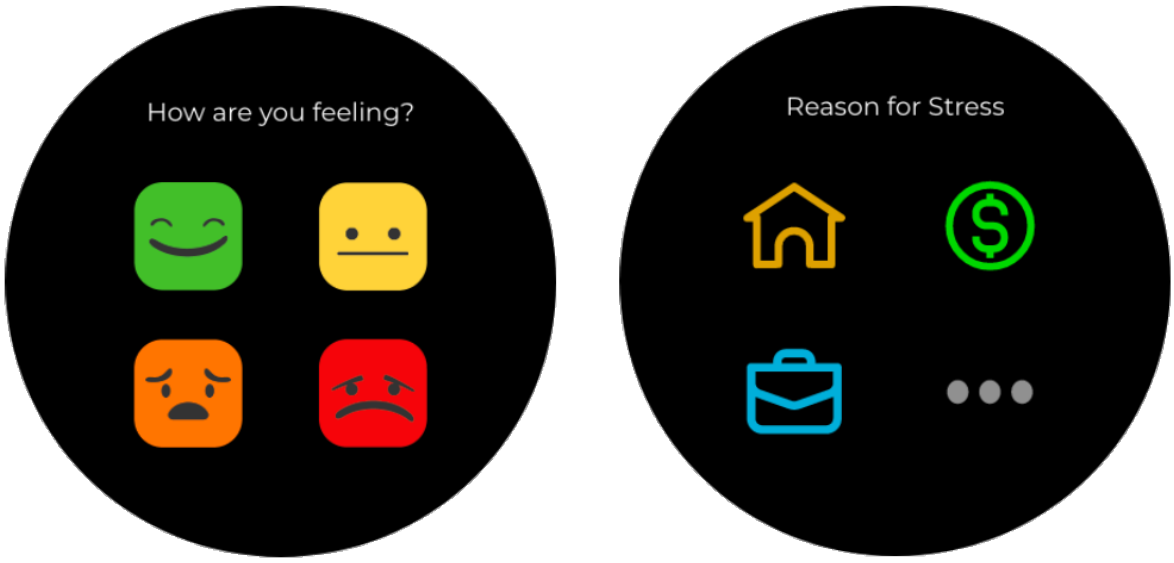



Figure 2.5: TASC Iconographic Labeling Interface

### Smartwatch Application

The smartwatch application was coded natively using Garmin's SDK development tool ConnectIQ. It was encoded in Monkey C, a Garmin-specific language that is derived from C++.

The watch included an iconographic Ecological Momentary Assessment designed to elicit the magnitude and type of stress a user was feeling. This design was adopted to optimize data collection on the smartwatch device, whose screen size limits the ability to input more than a limited set of button responses.

A point-of-care intervention, in the form of a guided deep-breathing exercise was also embedded in response to a user indicating a mild degree of distress in response to an EMA.

### Server and Dashboard

The architecture developed for VetThrive, with a google cloud based static data store, and React.JS encoded hosted web application, were migrated over and adapted to TASC's updated analytics.

In this case an 'OnLoad' page load would trigger a cloud function to query Garmin's API for any updated biometric data. The data would be read-in, and a sequence of python scripts would

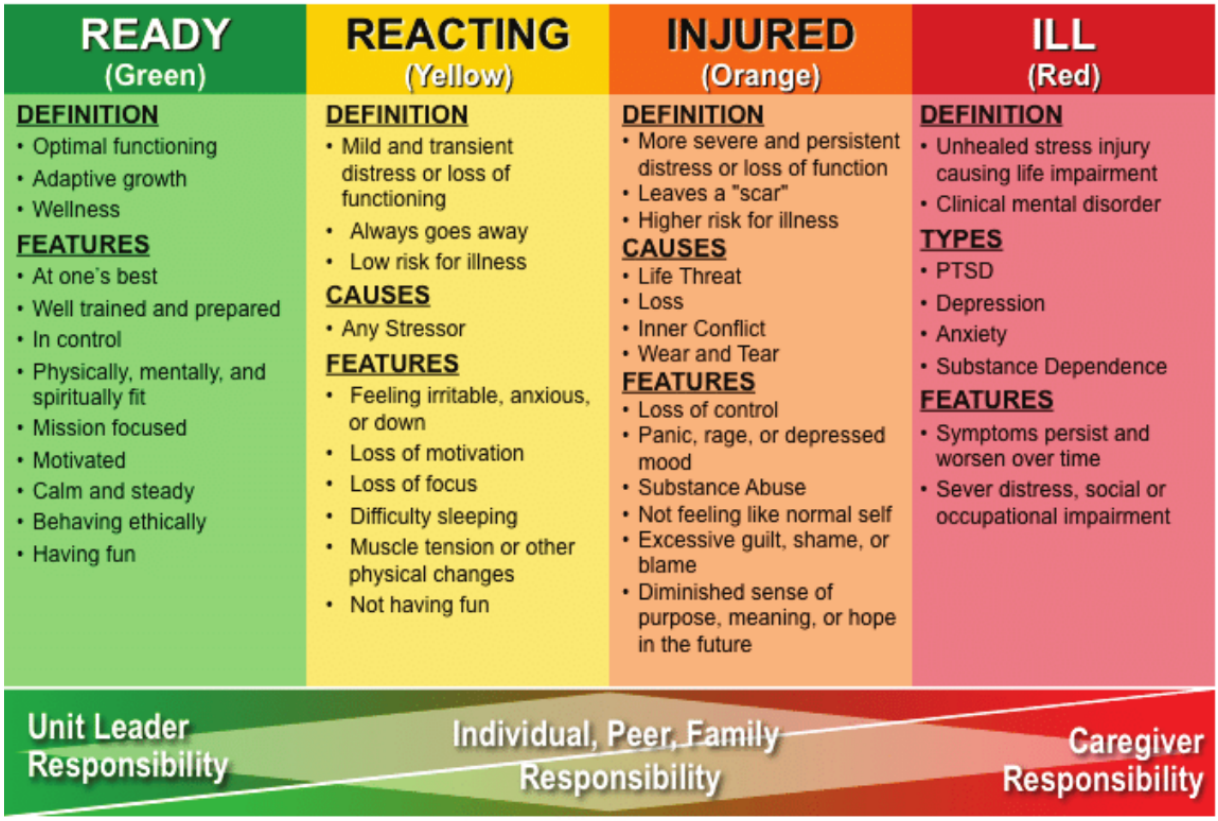



Figure 2.6: Likert Scale for Stress Continuum

process the data into aggregated fields and read them into fields the react dashboard would then use to populate the respective tables and graphs.

**PRISM (Promoting Resilience in Stress Management)**

The biggest functional expansion with this version of the platform was the introduction, not merely of data visualization, but of stress management techniques and tools, a platform we deemed PRISM (Promoting Resilience in Stress Management).

Under PRISM, user EMA responses were encoded to correspond to stress levels, they are established on a Likert scale of 4 and used the Stress Continuum Model. Figure 2.5 demonstrates this scale.

The PRISM model was designed to address the entire spectrum of stress responses that includes adaptive coping and wellness (represented by the color green for the ”Ready” zone); mild and

Figure 2.7: Original 'Ill' Icon

reversible distress or loss of function (represented by the color yellow for the "Reacting" zone); severe and persistent distress or loss of function (represented by the color orange for the "Injured" zone); and mental disorders arising from stress and unhealed stress injuries (represented by the color red for the "Ill" zone).

Because we designed platform for users to receive push notifications throughout the day, it was critical to solicit information in the most intuitive manner to achieve accurate data collection. Therefore each individual label of emotions has to be communicated unmistakably to the end user. Simplifying the labeling task with only 4 different types to choose from also reduces the data collection cognitive load and time burden. Furthermore, the attempt to use categorical labels instead of abstract and overly granulated numerical one disambiguate the labels from one another better.

Much thoughtfulness and keen awareness of current global circumstances were considered in the design of this interface. Originally, the concept of "Ill", the most severe stress level, was represented by a coughing icon. However, there was concern that a user might mistake this icon as an illness related to an infectious disease rather than identifying it with a stress level considering the global pandemic. Thus, in the final product, the distressed icon is represented by the icon shown in Fig 2.7, and the final watch face of stress level indication is shown in Fig 2.5. If the user selected yellow (reacting), orange (injured) or red (ill) icon, the watch further prompted them with another ecological momentary assessment (EMA) to identify the source of stress (Fig 6). The stress type labels consist of three major stressors in the EMA: "Family and Household", "Finance", "Work", and for additional flexibility, the "Others" option. The last option allows the user to customize the

reason in greater detail via the health portal dashboard website.

After the user indicated their stress level, the app triggered an ecological momentary intervention (EMI) based on their response. Each stress level triggered a specific intervention to match the severity of the stress level. If the user is experiencing moderate but manageable stress (the yellow icon), a breathing exercise would be triggered. If the orange icon is selected, the application triggers a custom message asking the user if they would like to have pre-selected friends and family members contacted. The app can call or send a text to the persons identified by the user, providing direct social support. If the red icon was selected, the app will trigger a message asking the user if they would like the Suicide Prevention or crisis hotline to be called.

**Issues with Data Pipeline and Realtime Monitoring**

The initial version of the application was built with the aspirations of leveraging biometric sensors to conduct real-time monitoring of an individual, with targeted EMAs providing just-in-time therapeutic interventions.

Further, we sought to take advantage of the presence of the smartwatch device, not only to serve as a data collection modality, but also as a direct interface for assessments and interventions, a modality that is more direct and convenient for the user over interacting with their smartphone.

However practical limitations inhibited this version from ultimately being fully realized, and ultimately it was discontinued.

This mainly derived from the functional limitations working with the Garmin hardware ecosystem imposed.

Garmin did not allow watch app developers access to sensor data directly off the device. All sensor data had to be queried from their API once it had been uploaded to their cloud. This effectively eliminated the possibility of doing client-side direct sensor monitoring (as had been originally intended with a basic expert-systems set of analytics hardcoded to trigger an EMA), or of using our own mobile application as a mechanism to upload data directly to our cloud.

Further, since the Garmin smartwatch connectivity was limited to only a bluetooth connection

it uses to tether itself to a smartphone device, the only way to offload cached sensor data was for the user to launch the Garmin Connect mobile application which would initialize the data upload to its cloud.

This effectively negated any possibility of real-time monitoring of the user, as we would have no access to any sensor data until the user manually launched their Garmin Connect app and then queried by our cloud API call.

Batched cloud processing of data was also not a meaningful option, although it is one we experimented with, (as it would have dramatically improved load times for our dashboard), as there was almost never new data to be processed given the aforementioned issue.

In fact, a dearth of updated sensor data was a persistent issue. As users had little motivation to manually upload their data, if left unprompted, we would often go weeks without updated sensor data (which further complicated API calls as we would have to dynamically modify the target window, and Garmin's API typically limited the retrospective data query to one-week).

We found that we had to prompt users to upload their data, and implemented a push notification to encourage them to upload data if no new data was available after a 48 hour window.

A similar consideration would occur within the TASC mobile app, where users would occassionally launch our application, having failed to first update their Garmin data. Consequently updated data for dashboards would not be available resulting in null-values that would be reflected in empty charts and tables. To combat this we similarly implemented an in-app banner that would notify a user if their sensor data was out of date and encourage them to load that data (by launching the Garmin Connect app) prior to relaunching our application.

The need for just-in-time data processing, proved to be a considerable UX issue. The need to query the Garmin API, read in data, then parse it into dashboard metrics and write them to their respective fields proved to be a lengthy one, typically lasting in excess of ten seconds. Early versions of the dashboard would simply have the dashboard load with blank fields, only to render suddenly after the user had already navigated away from the dashboard.

With batched pre-processing not an option, an interim solution of simply displaying an 'updat-

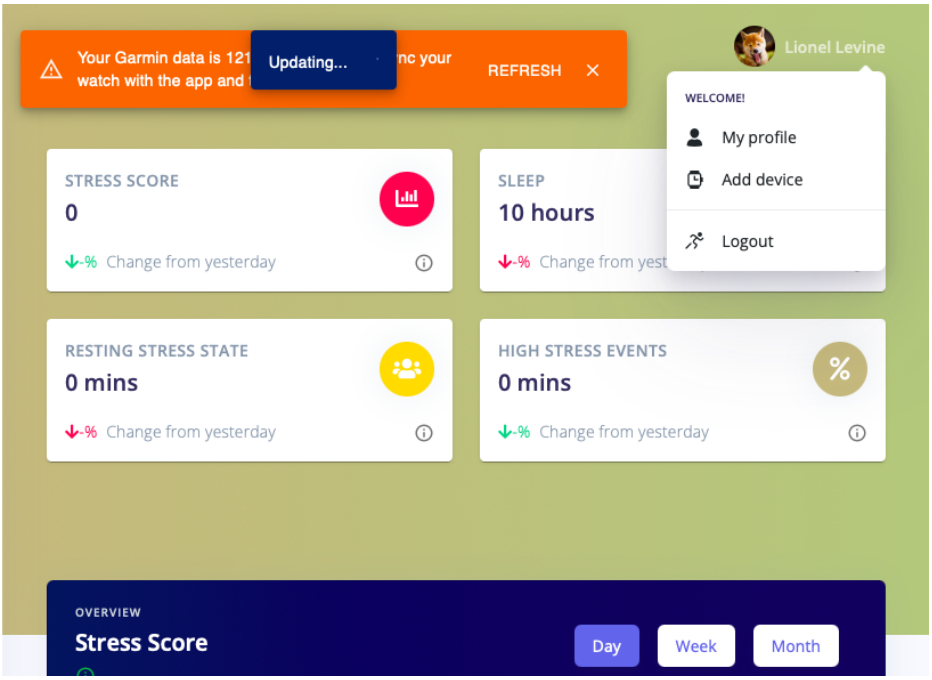



Figure 2.8: TASC Status Updating and Data Update notification statuses

ing' status to indicate to the user. Figure 2.8 demonstrates both the data out of date notification and the updating status.

Additional backend optimizations, including a prioritized queue for processing data, parallel processing, or even rewriting the base code in a compiled language to optimize runtime were explored but never implemented.

Further compounding matters, while Garmin smartwatch apps could be launched directly from the watch interface, or triggered by an accompanying smartphone application initializing a bluetooth connection with the smartwatch device, the watch itself was unable to trigger the smartphone app (a feature Garmin had previously supported but depreciated prior to us starting development).

Consequently, although the user could launch the TASC smartwatch app and record an EMA whenever they wanted to document a moment of intense stress, that action had to be cached on the watch, and could not be written to our server until the smartphone application was launched.

We were therefore in a scenario where users would be forced to launch their TASC mobile app, which, if circumstances warranted, would then trigger the smartwatch to prompt the user to complete an EMA. This was an unjustifiably complex and needlessly inconvenient user-experience,

one that effectively negated any potential benefits to using the smartwatch interface, which is what ultimately led us to abandon that interface, and instead migrate all TASC related functionality to the mobile application.

### 2.2.5 TASC (Updated Version)

The decision to migrate all end-user UX functionality off the smartwatch device and onto the dashboard, resulted in a bevvy of new features and use-cases.

A critical result of our delayed feature-data collection was we would have to jettison our Just-in-Time framework for label prompting and interventions, and instead adopt a more longitudinal approach for data collection and monitoring.

This would translate into a more extensive and systematic approach to mental health monitoring, as well as a more holistic and integrated approach to treatment.

#### Modified Labeling and enhanced data input capabilities

The EMA that was formerly on the smartwatch app, was now migrated to the user dashboard.

The same information was collected, and the same set of EMIs returned, however, now this was done in the form of an input modal on the mobile app, and disparate views on the app were returned in response to specific stress level indications.

Additionally, however a table view of stress instances was inserted into the dashboard, and additional visualizations were incorporated to visualize both the type and magnitude of stress.

The tabulated stress instances gave users a chance to review and edit prior EMAs, as well as provided an additional field for free-text input providing for additional context.

We also introduced a separate, journaling, functionality. This provided our users with a more long-form form of entry for documenting elements of their mental health.

### Communication and Interactivity

With the shift in platform focus towards a more holistic and longitudinal approach to mental health management, focus turned towards broader stakeholder engagement. Our earlier round of EMI incorporated both a 'phone-a-friend', and Crisis-Hotline Connection functionality, however there was the growing consensus that additional stakeholders needed to be a part of the framework, in what was termed a 'high-tech, high-touch' framework, one where the technology systems and associated analytics complimented and augmented in-person efforts.

Two specific stakeholder groups were identified and ultimately specific functionality built out for:

- **Healthcare Providers** Mental health professionals, who are seeing patients in therapeutic settings. The ability to augment their therapeutic sessions with additional insights, continuous monitoring, and proactive outreach was deemed a considerable value-add.

- **Commanding Officers** Given that the TASC app was built for originally for Special Operations Command, and subsequently for Marine Force Recon, and users were all anticipated to be active-duty service members. Commanding Officers have a vested interest in the wellbeing of those under their command. Additionally, maintaining and overseeing a unit's force readiness (or the capability of the unit to deploy to fulfill its stated mission), is an area of active concern for any commanding officer, and mental health, though a critical component of force readiness, is a hitherto un-monitored domain.

Sharing highly sensitive information with additional stakeholders, however, necessitates an additional range of concerns beyond standard issues around encryption and privacy.

For Mental Health providers, patients generally have less expectation of privacy, and a broader willingness to divulge highly sensitive topics in the scope of therapy. In that context more direct and personalized sharing of data was generally considered acceptable so long as users had explicit say in what was shared and with whom. therefore the design elements focused on explicit permissions, and ensuring users had full control and awareness over what data was being shared.

For commanding officers, in contrast, there were considerably more concerns around disclosing potentially compromising data. Here a confluence of overlapping concerns had to be factored into our approach.

There is generalized stigma towards mental health, a challenge particularly acute in the military, where it could be perceived as a sign of weakness, inferiority, or a lack of masculinity. Cultural concerns were exacerbated by professional ones, where there was considerable unease among users that mental health concerns, if reported, could result in career-related setbacks and demotions.

Conversely, though, there was the feeling among individual users that, overall, if commanding officers had a better understanding of the wellbeing of the service members under their charge, they would be able to do a better job tending to them.

This resulted in an analytic and design schema focused on fully anonymizing individual user data. In contrast to merely de-identifying data, where the data would have key identifiers removed or hidden, but there is still a risk that the data could be re-identified, anonymized data is data that has been permanently changed so that it can't be linked to any individual.

In practical terms that effectively precluded any individualized reporting, as it would be too easy to potentially map certain patterns in the data to publicly knowable schedules or life events.

Instead commanders would be provided with fully aggregated, summary statistics for units under their charge. This data would show the overall averages, and in some cases distributions of values, but no individual data, or outlier data would be made available.

While we were not able to extensively test this implementation in the field, so its actual utility isn't validated, it reflects a familiar set of compromises that constituted much of our data collection efforts. Neither the ability to collect, nor the utility of, data was often the limiting factor in its ultimate inclusion in our data schema. Rather it was the ever-present balancing act between wanting to maximize our feature data acquisition to improve out monitoring and predictive capabilities, against user discomfort and resistance to an overly aggressive and invasive monitoring regime.

Figuring out an effective balance of the two was a delicate balancing act that was an ongoing challenge that would mold our approach across systems and designs.

In this case the result was two additional dashboards. Users with authorized credentials could toggle between a personal user dashboard, or a **Provider** and **Commander** dashboard.

### Provider Dashboard

The Provider Dashboard was modeled after the individual user dashboard, with the same layout and content. Augmenting it however, was a provider level summary view, where a set of cards, one reflecting each patient, was rendered. The card also included a couple of high-level summary stats that would afford the provider with an 'at-a-glance' review of how their patient-population was doing, flagging any individual whose stats were worrisome for additional attention.

If a provider opted to take a closer look, they could then click on the individual's card, and underneath it their personal dashboard would load, offering the provider an identical view to what the patient sees.

In user account pages, they would be given the option to choose a provider to subscribe to, as well as a series of toggles where they could approve or block specific dashboards from being shared. Defaults shared all data, so it was an opt-out mechanism once a user subscribed to a provider.

### Commander Dashboard

The Commander Dashboard was designed in a similar manner to the Provider Dashboard. Commanders assigned to oversee certain units and companies (the hierarchy was flexible but was encoded for a battalion-level command, with unit and company-level sub-ordinate groups.

Like the provider dashboard, the homescreen displayed a series of summary cards for each unit under their command. Clicking on the unit would display a series of charts that were largely equivalent to the individual charts, however this time all data was aggregated into a unit-level average.

Unlike the provider/patient relationship, where the patient can opt in to monitoring and select a provider, for the commander dashboard, units were assigned directly on the backend and it was

envisioned that they would be updated administratively.

## 2.3 TASC (Final Version)

While the elimination of the Smartwatch application and the consolidation and expansion of the mobile application did dramatically improve usability, ultimately the need to manually launch the Garmin Connect mobile app in order to then launch and use TASC proved to be too frustrating a process for users, and alternative schemas were explored.

Of those Apple's Healthkit quickly emerged as a superior alternative.

Apple HealthKit is a framework that stores and synchronizes health and fitness data across devices, and allows apps to access and share that data. By virtue of it be a central enabling component of Apple Health's suite of features, most wearable devices (including Garmin) are healthkit compatible, and already store their data.

HealthKit stores all health data in one place, the healthkitstore, and users can access it from their iPhone, Apple Watch, and iCloud. This data, is stored locally on the user's device, and so long as the user grants an application access, a local app is able to pull the data directly.

This dramatically altered our data flow. With modifications to our native iOS app, we were now able to to access any data the moment it was updated in the HealthKit Store, and make asynchronous local updates to the app dashboard prior to writing updates to our server. This resulted in a dramatic improvement in UX upon initial load.

Perhaps most dramatically, by accessing HealthKit, we were also able to access all data available on it, and by extension any data modality the user was writing to it.

This meant that we were now able to access iPhone health related data, including steps taken and activity-level data and incorporate that into our data schema. We also experimented with incorporating additional wearable data by ingesting heart rate variability data from an Apple Watch.

This resulted in a dramatically improved user-experience, and a truly multi-modal platform for health data, but notably came at the cost of jettisoning our android users. Initial work looked

into Android equivalent platforms to Healthkit, however the work was ended once project funding concluded.

### 2.3.1 Conclusion

This work, spanning multiple applications, across different modalities, software and hardware applications, attempted to develop systems for the discrete monitoring of factors associated with mental health, along with providing novel platforms for individuals to engage directly with insights into their mental health, manage symptoms, and engage with key external partners in their care.

We continuously engaged stakeholders in the development and validation phases. There was a constant effort to balance user concerns about privacy and unobtrusiveness of data and label collection, with the very real set of hardware and system restraints, against the ultimate goal of accurate and timely detection of mental health episodes that may require additional attention.

While efforts to release this platform more broadly did not fully come to fruition, they nevertheless represent a significant body of work, providing insights on best practices for complex multi-modal system engineering, user engagement, and data collection mechanisms.

## 2.4 Dynamic User Querying and Opportunistic Sampling

**Introduction**

While all labeling work to-date was premised on a persistent labeling regime, one that could be enforced through research studies protocols, such a labeling system would not be practical in a production environment.

Further, given that the vast majority of episodic intervals would not result in notable levels of stress, labeling efforts should be limited to correctly identifying instances of acute stress and their accompanying severity.

This would be needed in the context of a 'high-touch' platform TASC was envisioned to be,

where interventions tailored to the user's preferences and needs would be curated and presented to them in a responsive, just-in-time manner.

Ensuring that the user was only offered up an appropriate intervention would necessitate first prompting the user to complete an EMA. This would have the dual benefit of ensuring that the user is not being inundated with unwanted and intrusive interventions, as well as providing us with a mechanism for ongoing data collection that could further improve upon our modeling accuracy. The question then became when to query the user about their wellbeing. It could not be at regular intervals, as compelling users to complete surveys, at even infrequent intervals, was a non-starter from a User-Experience standpoint. Instead the platform would only query users when it was suspected that they were recently suffering from an acute stress episode, and could use an intervention.

For this to work, a user's data would have to be continuously monitored, and the likelihood of a high-stress incident predicted. When the likelihood of a high-stress event crossed a critical threshold, the system would prompt the user to complete a wellness-check EMA.

Overtime the users' responses to this query would help to reinforce and improve the model.

**Expert-Systems Approach**

While it was ultimately hoped that our modeling would mature to the point where our predictive models were sufficiently robust so that they could be relied upon to generate these EMAs to users with an appropriate degree of accuracy, the paucity of data at the stage of system development necessitated a more algorithmic approach to stress detection.

To that end, in collaboration with our research partners, we devised an expert systems approach, leveraging both the Sleep, and Stress, metrics collected off Garmin to flag potentially worrisome longitudinal trends.

Figure 3.8 demonstrates the devised decision flow for this approach. An ongoing longitudinal approach was adopted, comparing aggregated stress and sleep quantity over a period of days, and then mapping them to the four stress continuum levels.

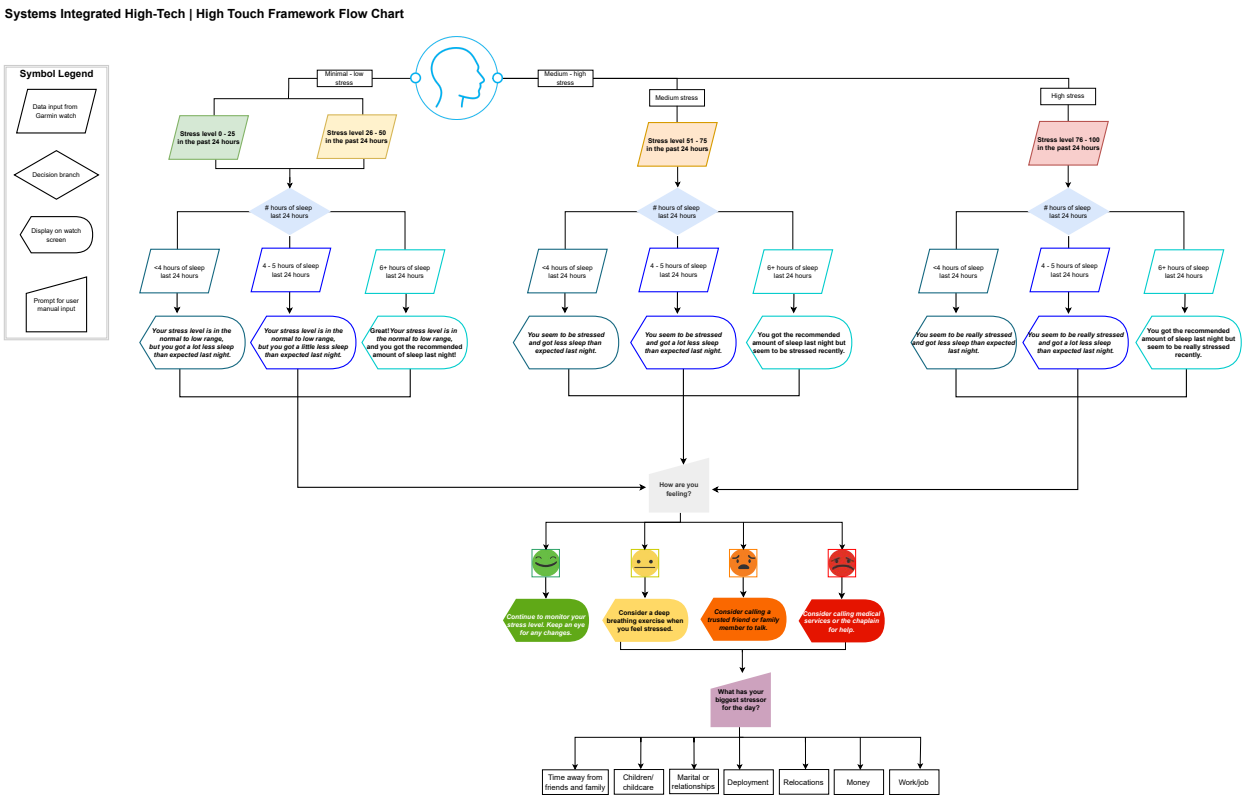



Figure 2.9: Stress/Sleep Longitudinal Monitoring Flowchart Schema

A customized series of user prompts were then devised to provide tailored feedback to the user.

Additionally, if the user was predetermined to fall into either the Orange (Injured) or Red (Ill) zones, an EMA prompt was then triggered to query the user for whether or not additional assistance was desired.

## Methods

As these metrics relied upon daily summary statistics, our first requirement was to create a user's daily aggregated summary of the biomarkers of sleep and stress and have the ability to access them without using the Garmin API again. As Garmin provided this data in a continuous stream, we would have to generate the series of aggregations internally.

Leveraging our backend database, and associated Google Firestore built-in automated scripting functionality, we encoded a sequence of Python scripts, designed to run at regular time intervals, and retrieve data from the Garmin cloud. All biomarkers relevant to this analysis, of all the users,

were queried at regular intervals and the results dumped to a collection. This collection is linked to each user for quick retrieval of data from the front end.

Given that users could not be expected to sync their Garmin data at regular intervals, the technical decision to retrieve data for all users at once was made to ensure any late data synced by the user after the previous time the script had run can now be consolidated to keep an accurate report of the user. Additionally the algorithm had to be adapted to prospectively search for any additional synched data by first identifying each user's most-recently synched data, and then prospectively querying for any updated data since, resulting in unique query windows for each user.

The collection of this data was then stored by date by biomarker for each user. We maintained collections for sleep and stress and stored them daily. For each day, we opted for a calendar day, synched to PST (GMT+8).

The algorithm was designed to bucket any data received during these intervals, and calculate a mean value for stress, and a sum for sleep. Daily aggregates were designed to be updated in the event that backfilled data for any new user was received.

Our second requirement was to then create an automated monitoring and alert system.

We segmented our framework by day and by week and for individual sleep and stress along with combined biomarkers. We named our individual biomarker framework the commander dashboard. The algorithm for this dashboard looks at the average stress levels in the past seven days for sleep and stress using a sliding average. We calculate the average of these biomarkers by keeping track of the days we received data in a week. As the data is manually synced from the smartwatch to our database, it was likely that users skip certain days. Thus our algorithm takes the average from all the days we received in a week and does not always average by seven days (by dropping null values). This ensured our average is kept accurate by the week. As previously noted, we have four zones within our algorithm under which users' health is flagged. Each of these zones has their own respective messages and actions that users can take to improve.

Our second framework is called the high-touch framework system. This framework is responsible for generating alerts based on a combination of sleep and stress biomarkers. This monitoring

system works daily.

The algorithm first looks at the stress of the user. Based on the day's stress, we then query sleep by the day to finalize the message for the monitoring system. If the user's stress is predetermined to be elevated, the algorithm will then trigger an EMA asking the user how they felt during the day. Based on the response of the user, we provide an additional layer of messages to help the user improve their health. Figure 3.8 explains the threshold for the stress and sleep levels of the user for which we decide what alerts to present.

Our last requirement was to generate a messaging system that can be seen on the UI by the user. For each of the frameworks described above, we have set up a messaging system that is created on a daily and weekly basis as explained by the framework. We created a collection for each user which keeps track of each message for the framework. This collection has the subject and message along with the date created and the date read. Figure 3.9 demonstrates the backend data storage component for these messages.

In the front-end UI, this system was realized by incorporating a status alert on the top of the application when the user logged in, and a push notification prompt, and in-app alert to query the user to complete an EMA. This has the added benefit of potentially triggering an EMA outside of user-app use, should the cloud determine that there is sufficient need to prompt the user.

## 2.5 Subsequent Work

We had initially proposed an ambitious follow-on project to this work, one designed to merge prior efforts from my colleagues around opportunistic learning and cost-aware information retrieval, with our mental health data collection pipeline with the ultimate integration into TASC.

Unfortunately due to logistical and financial constraints, continued work on the project proved untenable. We were unable to secure the funding necessary to collect additional data necessary for robust modeling, or conduct any validation studies. Attempts were made to secure external datasets to augment our efforts, but no publically-available datasets proved complimentary to our efforts.

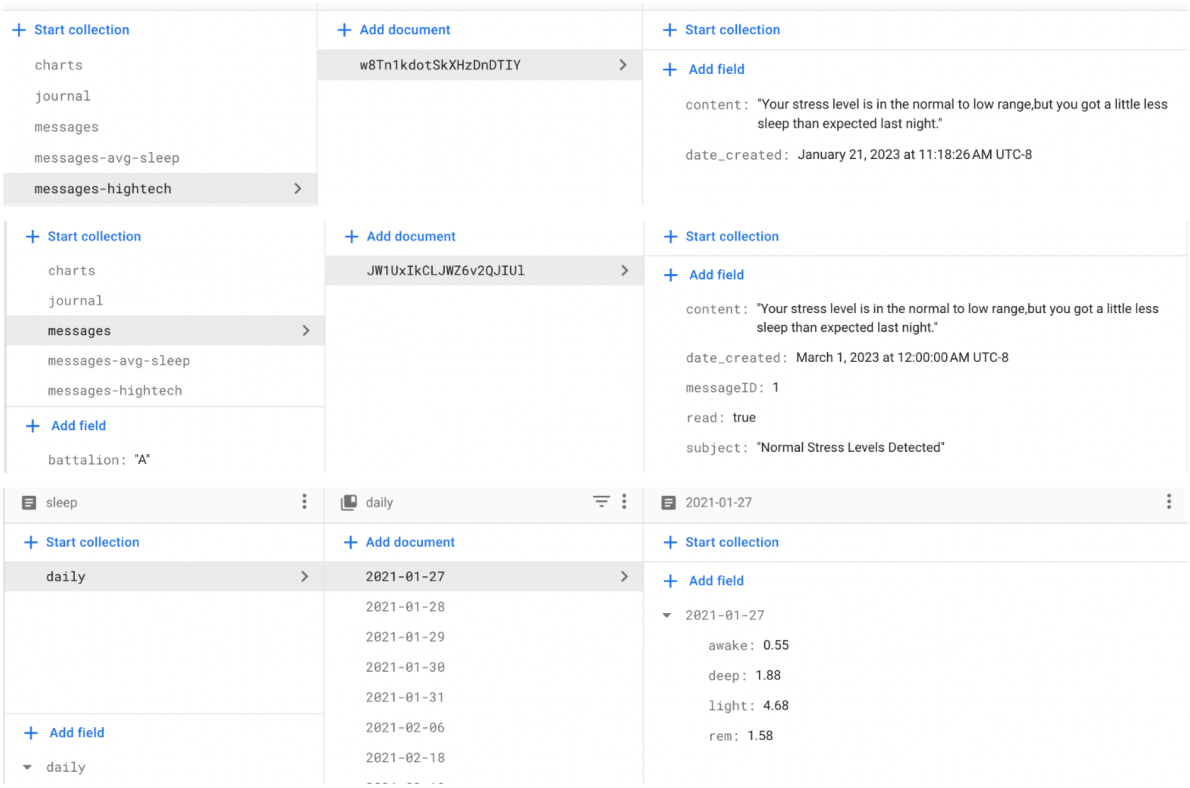



Figure 2.10: A sample of cloud-generated alerts to be relayed to the user when launching the App

Ultimately we were forced to pivot away from this work and towards efforts where open-sourced data could be leveraged for experimental purposes.

# Chapter 3

# Labeling Intractable Classes: A Mental Health Case Study

## 3.1 Introduction

Accompanying our efforts to build out systems dedicated to the treatment of mental health, we concurrently sought to develop robust models around classifying and predicting the presence of mental distress. The goal, from a platform perspective was to better anticipate user needs and potentially flag individuals in crisis for more intensive and targeted interventions.

Any Machine Learning system has inherent challenges of trying to develop an effective and robust predictive capacity for a label that may have, at best, a tenuous relationship to the feature data being provided for it. However this is a doubly complex issue in the domain of mental health, where even the concept of health is so amorphous and poorly articulated that applying discrete classifiers to the issue can seem laughably crude to the challenge.

Further confounding the issue of accurate labeling, is that even in the event that an accurate schema can be developed, employing an effective mechanism for producing the associated labels is a notoriously challenging endeavor. Any labeling schema dependent on human input is wrought with challenges obtaining consistent and reliable labeling, but this is further challenged in the case

of mental health, where evasive and deceptive labeling is an expect occurrence.

Nevertheless this vexing challenge is one we have spend considerable effort working, developing theoretical models and conducting multiple experiments. Ultimately we have devised several unique contributions to the domain and brought significant insight to the challenges in this, and similar areas of data curation.

## 3.2 Stress Management Analytic Challenges

There are several generalized analytic approaches to the classification of mental illness. One that has been central to our approach to the problem, is to look at mental health as a progressive and evolving state, with a tendency to evolve overtime. This longitudinal approach to monitoring aligns with a growing consensus in the field around approaches to monitoring and care. In contemplating improved means to address mental health challenges generally, and anxiety-related disorders specifically, it is notable that a critical part of modern healthcare involves accurate and efficient tracking of individuals' well-being through time.

Examples include tracking athletes and their training trajectories, and patients' rehabilitation exercises [104, 42, 115, 41, 129]. Compared to physiological health, the mental health domain is less investigated in the context of remote health monitoring.

This is largely due to a confluence of reasons. For one, the statistical sufficiency of observations obtained via data-driven approaches is not often intuitively clear (e.g., can one draw a conclusion regarding depression from the number of phone calls?). Another reason is that the data required for enabling the use of artificial intelligence (AI) is often not readily available or exclusive due to privacy and regulatory concerns.

The works in this domain, therefore, have mostly focused on longer-term patient phenotyping (e.g., classifying patients into bipolar disorder vs healthy) [47].

While these high-level labels are useful and can have critical importance in healthcare provision, helping to, for instance, identify individuals with potentially undiagnosed conditions, and

redirect resources towards at-risk individuals that might otherwise be forgoing care.

Side-stepping a formal diagnosis, another approach is to instead attempt to classify the presence of acute and debilitating stress. Stress often manifests as an emotional and physiological response of an individual to a triggering event, and can occur to anyone regardless of a formal diagnosis. For example, arguing with someone and being anxious about a deadline are instances of interpersonal and work-related stress that are liable to occur to anyone regardless of the existence of a pre-existing mental health disorder. Furthermore, the highly localized and temporary nature of these shorter-term episodes, which given the scarcity of data, makes making the most out of the available observations critical.

Nevertheless, psychological stress is known as one of the most widespread mental health problems. Everyone, regardless of their background and the possible presence of any clinically diagnosed mental illness, experiences significant bouts of stress that induce significant emotional responses at some point throughout their lives. There are various kinds of stress, including financial challenges, relationship issues, work-related problems, traumatic events, life changes, and loss, as prominent examples. Individuals' background, such as race/ethnicity, age, place they live, marital status, social life, and number of children, puts them at different risks of suffering from bouts of stress. One of the important factors in this domain, which we have taken into consideration in this work, is their occupation. College students are a group of individuals at high risks of stress, anxiety, and depression[84]. The members of the military are also a community at particular risk for various stress-related disorders. It is estimated that one in five veterans are experiencing stress-related mental health problems, including stress, anxiety, post-traumatic stress disorder (PTSD), and major depression [29].

The use of smart systems in enhancing and improving mental health treatments and patient monitoring, nevertheless, is considerably less investigated compared to its physical health counterpart. Passive sensing is a special subdomain that pertains to proposing non-invasive solutions that can be seamlessly integrated into the daily lives of individuals and, with no or minimal intervention, help paint a more complete picture of the patterns crucial to helping them make progress

towards a healthier and happier lifestyle.

Considering the complexity of the relationship between physiological signals and perceived stress, we aimed to design and conduct a thorough study and empirically investigate the statistical sufficiency and semantic capability of learned representations for stressful episodes resulting from processing temporal physiological signals.

## 3.3 Experiments

### 3.3.1 eWellness Data Pilot

#### Methodology

An IRB-approved pilot study was conducted on a dozen individuals who are using the Android version $5.0$ and above from the university community, including students and staff. Study participants did not have a reported history of mental illness. Participants were asked to download and install the eWellness application (Figure **??**), and then run it on their phone for a month. Passive data was collected continuously by the application throughout the month. Participants were asked to answer EMA daily through the eWellness app, but did not provide any other personal information, such as name, gender, age, during participation.

The Kessler Psychological Distress Scale (K10) [3] is a validated measure of psychological distress over the past 30 days, which is used for clinical and epidemiological purposes. It has a notable success in measuring feelings of anxiety along with depression. For this pilot, the K10 was modified to assess criteria over the previous 24 hour period. The modified K10 prompted the users as daily EMA to measure their feelings of anxiety and depression. The K10 is composed of ten questions, structured on the following standardized template, "Over the past 24 hours, how often have you...", to which users can provide one of five standardized responses: All of the time, Most of the time, Some of the time, A little of the time, and None of the time). These responses are scored on a range from five (All of the time) through one (None of the time). The minimum

Table 3.1: Categorization of K10 Scores [10]

| K10 Score | Level | Samples (N=146) |
|---|---|---|
| 10-15 | Low distress | 91 |
| 16-21 | Moderate distress | 29 |
| 22-29 | High distress | 21 |
| 30-50 | Very high distress | 5 |

possible score of K10 is $10$, and the maximum possible score is $50$. K10 results are categorized into four levels of psychological distress: low distress, moderate distress, high distress, and very high distress (Table 3.2). Table 1 details the stress threshold scores. These results were leveraged as a label for the classification of supervised learning.

**Results**

Only 10 participants answered at least seven days of EMAs and provided successful passive sensing data throughout the month. Our analysis focused on a fully supervised learning approach, and only labeled samples were included. For this pilot study, we used 146 daily samples to identify daily anxiety and depression levels. The Z-Score normalization was applied to the features to reach normalized values from different participants.

**Discussion**

**Relevant Features** There are some notable and counter-intuitive findings regarding what data elements proved to be most-highly correlated to mental health. It is not surprising to note the presence of features closely related to physical activities (e.g., Duration of time spent biking or walking) as such activities have been definitively linked to mental health [108].

What is somewhat less intuitive is the presence of multiple audio and light sensing features. Audio sensing was included in the protocol under the hypothesis that a moderate level of sound could be indicative of pro-social activities like being outdoors or in group settings. Conversely, overly loud or quiet noise profiles could be indicative of stressful environments or isolated conditions that could be deleterious to mental health. But while interesting in theory, there are many

confounding causes of noise that, by limiting ourselves to solely capturing the frequency and decibel levels of the sound, we would fail to distinguish. (intuitively, someone watching TV at home alone could register the same noise profile as someone out to dinner with friends).

Similarly, it was hypothesized that light sensing could be indicative of an individual being outside, which has been shown to positively correlate to mental health [78], however here too, many confounding factors would impact light readings, foremost among them, that the user would actually have to have their phones out and exposed when outside for the light sensor to register it.

The authors note that Sound and Light sensing is notable in that both were the most frequently sampled of all features. It is possible that the high degree of granularity of readings afforded to these particular sensing modalities explains their relevance. Regardless, the authors suggest additional work is needed to understand whether or not these features are indeed more universally indicative of mental health, and explore why that is potentially the case.

**Limitations of the Study** While 10 subjects completing one-months worth of continuous data represents a critical validation of the technology and its potential utility, the dataset is too small to achieve statistically significant results. Additionally, this pilot was scoped to only include individuals without a clinical diagnosis of Anxiety. Consequently, there were insufficient cases of user-reported mental distress, particularly moderate or severe cases, in order to classify them effectively. Additional studies are planned to enlarge our dataset and include a cohort of individuals with diagnosed mental health conditions.

**Accuracy of Labeling** The authors feel there is significant concern about the veracity of user self-reported labeling of mental health that was leveraged in this study. When constructing the experimental design, focus was placed on maximizing user participation in the study. At the time, the primary concern the authors had was that participants would fail to submit a sufficient number of survey responses. Therefore the protocol was designed to combat this, by prompting users to fill out a daily EMA in the application via push-notification, with manual outreach to users who failed to complete an EMA within 48 hours, as well as designing the K10 to be a simple to complete multiple-choice assessment. This combination resulted in successfully encouraging

active participation in the study; however, there was no mechanism designed to confirm or validate that the resulting inputs were an accurate reflection of a user's actual wellbeing.

It is highly likely, therefore, that at least some users were motivated to respond quickly, and not necessarily accurately. This would result in users simply selecting the default answer of no reported anxiety to each question.

Furthermore, there may have been a reluctance among users to accurately report out mental health issues given perceived embarrassment or stigma associated with poor mental health. Under-reporting of mental health issues is a persistent issue that plagues the domain more generally, and isn't limited to this study [30], however failing to account for under-reporting is a notable issue.

Finally, even well-meaning participants may have failed to accurately represent their mental health state due to their either overlooking, or mischaracterizing, stresses they encountered. This is particularly true when comparing responses across users, where baseline expectations of stress may vary wildly among participants, with prior work for instance demonstrating a clear association between gender and reported wellbeing [97], the result being that one participant's perception of a 'normal' day, might easily be classified as a low or moderately-stressed day by another.

Solving this challenge is essential for ultimately achieving the intended goal of accurate classification of mental illness, for unlike alternative labeling exercises, where quantifiable metrics are possible, here the labeling of an objective state, mental illness, particularly when physiological monitoring is not available, is entirely reliant on subjective inputs, ones that are difficult to accurately capture, and even more difficult to standardize across users.

The authors recommend that future studies will have to address these concerns by better anticipating and correcting for challenges with accurate labeling of mental health.

There are a number of possible remedies to this. In the questionnaire itself, careful structuring of the questions can engage users to provide more thought-out results [61]. Cross-validating questions designed to ensure internal consistency are also an effective means of ensuring user accuracy [71].

Consideration should also be given to alternative methods for collecting labels. Interviewing

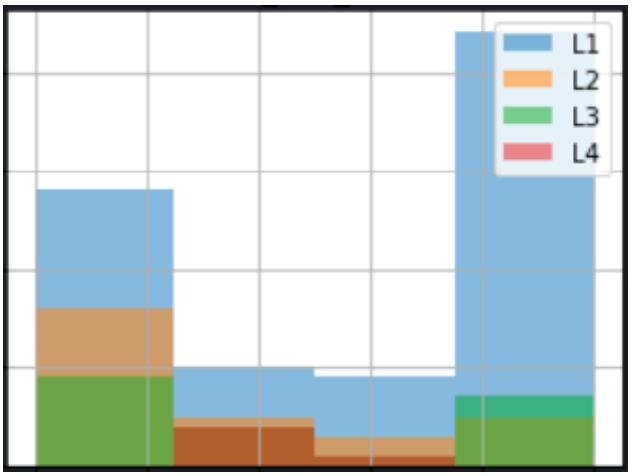



Figure 3.1: Histogram of normalized values of Duration on Foot for the 4 labels of stress

subjects to determine their mental health, for instance, would likely produce more accurate results, although would have attendant tradeoffs of its own, such as reducing the number of labels that could effectively be captured.

Educating participants on the presentations of Anxiety could also be key towards a more accurate and consistent recall of symptoms. Finally, developing the trust of participants through engagement and transparency, could help to solicit more honest engagement.

**Subject Heterogeneity** The activities tracked by the VetThrive app showcase significant heterogeneity across subjects in-terms of usage-patterns. Variables like distance-traveled, number of texts and calls, and physical activity levels, are all far more likely to be impacted by the individual's lifestyle, than their mental health on any given day.

Fig. 3.1 showcases a fairly typical distribution, in this case the duration spent on foot in a given day, bucketed into quartiles, with the 4 labels of interest (with L1 or Level-1 corresponding to Low-distress, L2 to Moderate Distress, L3 to High-Distress, and L4 to Very high Distress), in this classification superimposed. While intuitively more time spent on foot may be associated with better mental health, here we observe no clear pattern.

It was therefore assumed that primary-success would be achieved by classifying mental health within users across time, once their baselines for normative behavior were established, rather than across users. The limitations of this initial dataset did not allow for adequate classifying by indi-

vidual; however, the fact that classification success was achieved by bundling samples across all subjects is remarkable in its indication that cross-subject learning in this domain could be possible. The authors suspect that part of this result likely stems from normalization performed on the data to account for habitual differences in subject usage. By normalizing the data in this manner, the absolute number is rendered largely moot, and instead variances in user patterns are highlighted, as it is likely the day-to-day variations that are more reflective of shifts in mental health. Additional data collection is necessary to validate this finding.

**Usability** Attempting to gauge the viability of the concept, participants in the pilot were asked to submit a voluntary anonymized post-study questionnaire regarding their perceptions about the application and its data collection practices. All participants responded. A significant majority described the application as somewhat (40%) or mostly (40%) useful. Likewise, all users endorsed feeling comfortable with the application, and only one user expressed reservations about the data being collected.

All participants obtained detailed accounting of the data that was collected as part of their onboarding process to the study. No individual declined to participate after learning the precise nature of what was being tracked. This sampling suggests that, particularly among the young adults who are more accustomed to digitized lives, there is less concern about data collection through their mobile devices. Limiting the collection of PII could be sufficient to assuage most privacy concerns.

The primary issue users had with the application was its battery consumption resulting from heavy over-sampling of the sensors. Future iterations of the application will seek to optimize battery usage by minimizing the sampling frequency.

T-distributed Stochastic Neighbor Embedding (t-SNE) [76] provides additional insights as to the nature of the collected data. t-SNE represents each high-dimensional object to a two-dimensional point in such a way that nearby points model similar objects and distant points with high probability model dissimilar objects [76]. In Figure 3.3, the vast majority of data points are red, which is low distress (level 0). While some of these level 0 data points are scattered, a distinct cluster of level 0 is perhaps segmented by gender type or mobile device usage type.

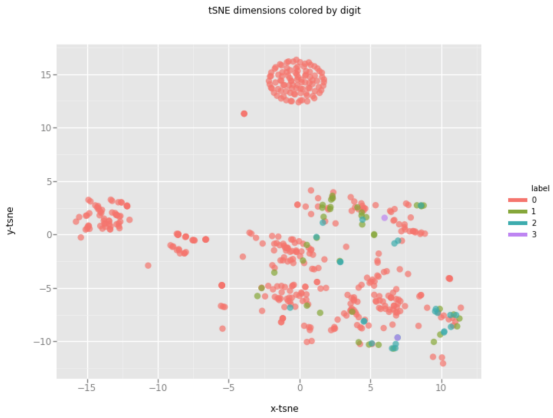



Figure 3.2: tSNE mapping of classes.

We selected 25 features that have a relatively higher correlation with the raw K10 score. Table 2 provides a detailed list of feature labels and associated descriptions.

For the 4-class classification, we used 5-fold Cross-Validation (CV) with four models: K-Nearest Neighbors (KNN), Extra-Trees (ET), Support Vector Machine (SVM), and Multilayer Perceptron (MLP). The class weight was automatically applied to the models inversely proportional to the class frequencies to train the imbalanced dataset. The highest classification accuracy achieved was around 76% with the extra-trees model. We also applied the under-sampling technique to improve the performance of an imbalanced dataset. Samples from the low distress class were removed randomly to make uniformly distributed class labels. Samples from the very high distress class were also ignored. A confusion-matrix demonstrates that the average score of classifying three classes is $0.65$.

### 3.3.2 TASC

#### Experimental Methods

We conducted a smaller data collection pilot at UCLA with the ultimate objective of our study being to present a working solution that can be used by the general population for continuous and passive monitoring of bouts of stress, we concentrated on the intersection of two groups of people particularly at risk for stress, namely, college students and military members.

Table 3.2: 25 Features Most Highly Correlated to K10 Scores

| Feature Name | Description |
|---|---|
| total-messages | Total # of text messages-received |
| is-silent-count | Number of instances no noise was detected |
| freq-std | Standard deviation for noise frequency |
| freq-25% | Noise frequency 25th percentile value |
| deci-std | Standard deviation for noise decibel |
| deci-50% | Noise decibel 50th percentile value |
| deci-75% | Noise decibel 75th percentile value |
| rms-max | Root mean squared measure of audio over time |
| act-transition | Activity tracking count when in transition |
| still-cnt | Total count of time user was still |
| tilting-cnt | Total count of instances the user was tilting |
| on-foot-cnt | Total count of the instances the user was still |
| on-bicycle-cnt | Total count of time user was riding a bike |
| on-foot-dur | Duration of time on foot |
| on-bicycle-dur | The duration the user spent on a bike |
| elapsed-device-on | Count of time the phone was active |
| elapsed-device-off | Count of time the phone was inactive |
| light-std | Standard deviation for luminescence value |
| light-25% | Luminescence 25th percentile value |
| light-50% | Luminescence 50th percentile value |
| loc-speed-mean | Average speed traveled in a day |
| loc-alt-mean | Mean Altitude Location |
| loc-alt-std | Standard deviation od the altitude |
| loc-alt-75% | Altitude 75th percentile value |
| loc-alt-max | Altitude max value |

Table 3.3: Classification Accuracy of 5-fold Cross-validation

| Model | Accuracy (SD) |
|---|---|
| KNN | 0.681 (0.085) |
| ET | 0.760 (0.097) |
| SVM | 0.673 (0.083) |
| MLP | 0.656 (0.088) |

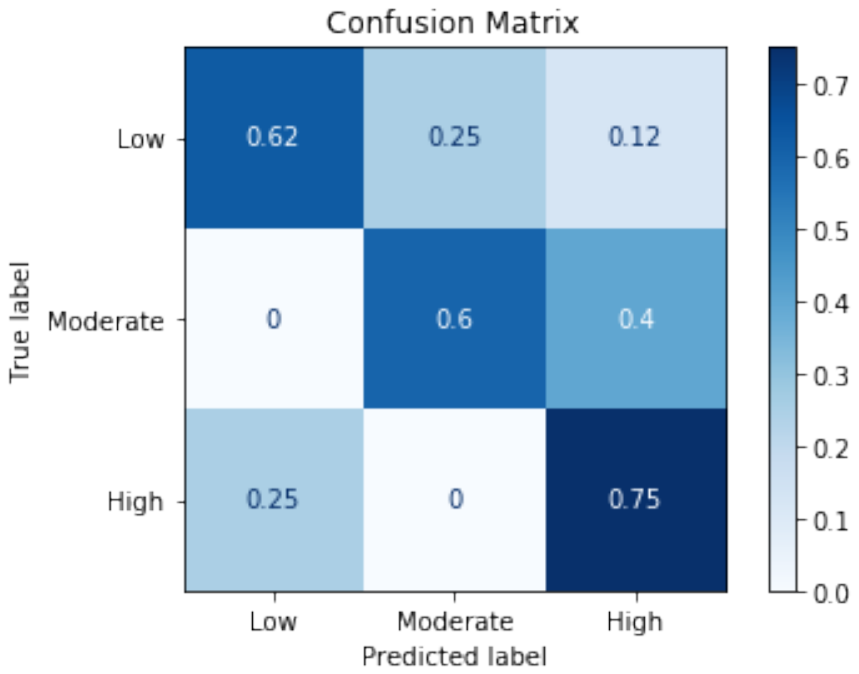



Figure 3.3: 3-class (Low, Moderate, and High distress) Classification Confusion Matrix.

To create our cohort, we recruited former and current members of the military within the University of California - Los Angeles (UCLA) community for participation in this study. We published our study's advertisement in social media channels and military-affiliated organizations.

Interested individuals were asked to complete an initial online screening survey. Those determined to be eligible participated in a virtual informational onboarding session to discuss the study protocol and determine their willingness to complete the required components of the study.

Bouts of stress in response to triggers or anxiety does happen to everyone regardless of the formal clinical diagnosis of other mental health disorders.

For this reason, and because our aim was to learn universal representations for the stress recognition task and evaluate the feasibility of such a system in the absence of any patient-identifying background information, there was no need for a full clinical assessment at the onboarding phase.

**Data Collection**

Participants were instructed to wear the watch continuously for 10 consecutive days. Toward the recognition of perceived stress, we also needed to gather supervision signals. To this end, we designed a labeling scheme to allow participants to submit self-reported details of their stressful episodes.

An online survey particularly designed for this purpose was sent to them via email, daily, in which they were asked to provide:

1. A rating of their overall perceived stress level for that day (`none`, `low`, `medium`, or `high`).

2. A recounting of any specific stressful incidents, including their description of the context and what they believe to be the cause or trigger, in addition to the approximate time or timespan at which these episodes occurred and perceived magnitude (`low`, `medium`, or `high`).

3. Any times during which the device, based on its solely HRV-based approximation of stress, triggered a high-stress indicator

4. Any times during which the watch was removed and for what reason (e.g., for charging or showering).

Our chosen smartwatch has an internal approximation of physiological stress based on sole analysis of the heart-rate variability (HRV), and once this value gets past a certain threshold, the watch will issue notifications to the individual so as to focus their attention on what could be a stressful episode.

As a basic intervention measure, the smartwatch then gives the user the option to perform a breathing exercise to help relieve the stress.

Additionally, to ensure that sufficient records of stressful episodes were present in our data, with their informed consent, the participants were asked to undergo two remote, stress-inducing laboratory sessions under the direct guidance of study personnel. These sessions were scheduled at least one day apart and around the same time of day.

Study personnel would guide the participant over a video call through a series of three exercises designed to induce a controlled amount of stress while the participant wore the smartwatch.

Each exercise has been clinically established as a method for induced stress control, invoking both cognitive, social, and physical stress.

The specific exercises were a cold pressor test [11], Stroop Test [82], and Maastricht Acute Stress Test (MAST) [105].

For the Stroop Test, participants were shown the name of a color in a colored ”ink.” They were asked to say aloud the ink color and press the corresponding key on their computer.

For example, the word ”purple” appears as a green color.

Participants have to say “green” and not purple.

The color names were shown one at a time for a total test time of about one minute. The cold pressor test involved the participant first submerging their hand without the watch in room temperature water (around 38°C) for two minutes, followed by submerging their hand in ice cold water (0°C- 4°C) for 2 minutes.

Lastly, the MAST exercise consisted of a two-minute timed arithmetic test while they submerged their hand in ice-cold water in the same fashion as the cold pressor test. The participant’s self-assessed level of stress (measured as low, medium, or high) was recorded after each test.

Participants were also asked about their activities before and after the stress test session to give context for their emotional state before and after the laboratory sessions.

**User Self-reported Discrepancies between biometric stress detection and emotional perceptions of stress**

To further validate existing literature on the divergence between emotional perceptions of stress, and biometric stress readings, subjects were further asked to document instances when the smartwatch signalled that they were experiencing a high-stress episode, and to report their concurrent perceptions of their own stress.

Cumulatively our sample test cohort reported a total of 15 high-stress notifications from their

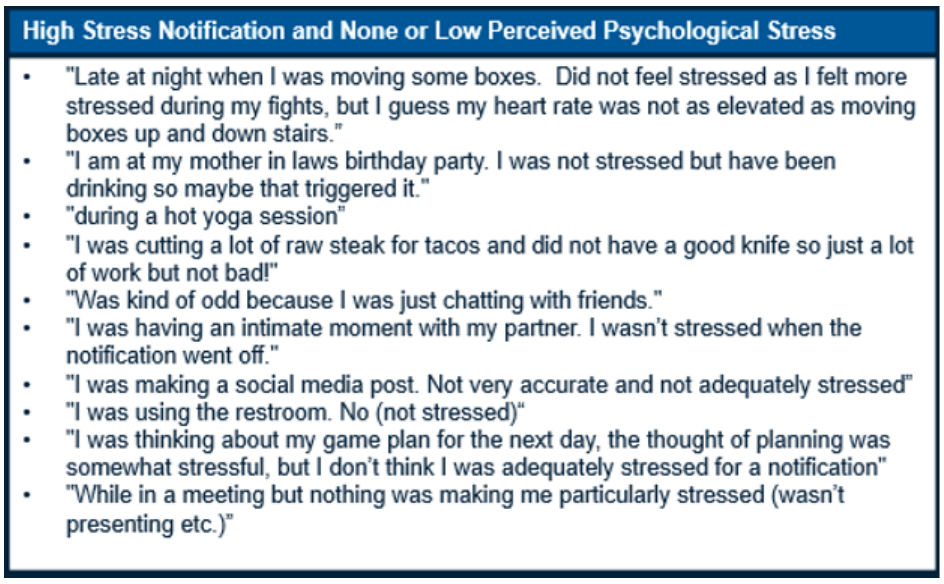



Figure 3.4: Anecdotal Recounting of False Positive Triggers

smartwatches. Of the 15 instances, participants self-assessed 5 of them to be True Positives- indicating the stress they felt warranted a high- stress notification, and 10 False Positives, indicating that their perceived stress did not align with their physiological stress response. This suggests an anecdotal Precision of 0.33 for accurately classifying moments of significant emotional stress.

Conversely, participants self-reported a total of 10 incidents in which they reported feeling high stress. Of those 10, 5 had a concurrent high-stress indication from the smartwatch, indicating a Recall of 0.5, and an estimated F1-score of 0.4.

Users also provided accompanying descriptions of their activities associated with a high- stress trigger. As expected, physical activity, alcohol consumption, social interaction, and work-related activities can all serve as triggers for physiological stress responses. False negatives, when users self-reported high-levels of emotional stress that did not coincide with a high- stress trigger, tended to take the form of social stress (in the form of arguments), physical duress, or work- related stress.

While this data isn't statistically robust enough to be reportable in its own right, it does serve as a performance baseline for current commercially available stress modeling and helps to better contextualize our own model performance in that light.

| Sensor | Rate |
|---|---|
| Pulse-Oxometer | every 60 seconds |
| Heart Rate | every 60 seconds |
| Respiration Rate | every 60 seconds |
| Stress/HRV | every 180 seconds |
| Cumulative Sleep | 1 per-day |

Table 3.4: Monitored Sensors and Associated Monitoring Rates

### Preprocessing and Initial Data Analyses

Data were collected in a continuous manner over regular intervals as predetermined by the smartwatch manufacturer. For the purposes of this study, both raw sensor data, as well as a selection of features produced by the smartwatch, were collected. These included the pulse oximeter/oxygen saturation, respiration rate, and heart-rate data. The watch also produced a Stress level indicator derived from the measure of heart rate variability while the individual was sufficiently inactive.

Finally, the user's sleep quantity from the previous night was tallied and included.

Table 3.5: Number of participants in our cohort per category based on their duty status and service branch

| | **Duty Status** |
|---|---|
| National Guard | 2 |
| Active Duty | 3 |
| Veteran | 9 |
| | **Service Branch** |
| Airforce | 2 |
| Marine | 3 |
| Navy | 4 |
| Army | 5 |

Significant null values were observed in the data we collected.

This is an acknowledged issue with wearable sensors, particularly smartwatches that are not typically affixed tightly to the skin [25], when consistent skin contact is required to accurately take a reading. Figu demonstrates one such example where a user's respiration rate is riddled with null values throughout the high-stress incident of a car breaking down.

In attempting to address these gaps, various data imputation strategies were attempted. Ulti-

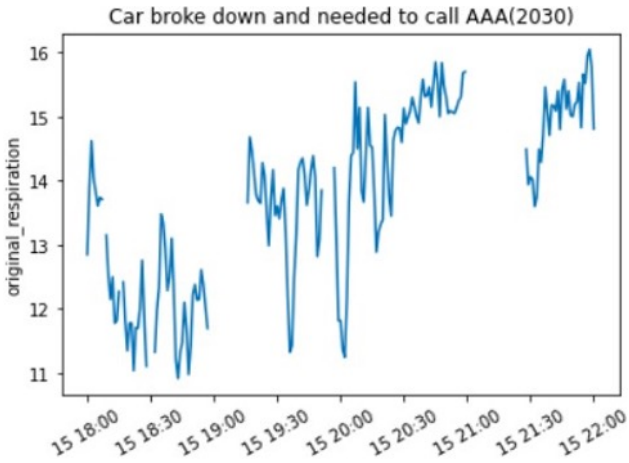



Figure 3.5: Visual Example of Null Values Observed in Measured Metrics

mately, they were not included in the final analysis due to the relatively small quantity of data and the challenges inherent to conducting accurate imputation under those circumstances. Instead, data was down-selected only to include intervals when complete data was available, and for episodes with a low degree of missing information, we leveraged a causal interpolation scheme to utilize the last known value and complete the available episode information by that. Stressful events in the data were labeled using the daily journal surveys completed by participants, in which they specified approximate periods of high stress, the stress magnitude, and the type of stress. The survey included pertaining to the approximate occurrence time associated with the stressful episode as well, and that data as well as dates were obtained and processed accordingly.

### Label Smoothing and Window Augmentation

Given the limited availability of data for enabling efficient training of our pipelines, we needed to generate as many training examples as possible using our corpus. To this end, we considered a label smoothing strategy to allow proposing an estimate of the stress level at each point in time, leveraging a non-causal smoothened version of the events obtained from the self-reports. For each individual, we aimed to create a continuous and real-valued *teacher function* $f(\cdot)$ enabling us to

```
{
    "subject_id": "12345",
    "stress_type": "general",
    "perceived_rate": "low",
    "stress_rate": "low",
    "stress_description": """
    I was stressed because of a deadline
    """,
    "probe_datetime": (
        datetime(
            2021, 5, 10, 9, 0, 0,
            tzinfo=timezone.utc),
        datetime(
            2021, 5, 10, 18, 0, 0,
            tzinfo=timezone.utc))
}
```

Figure 3.6: A sample stress probe datapoint, showing the record structure

map a timestamp $t$ (in seconds) to an estimation of the severity of stressful response.

$$f(\cdot) \leftarrow f(\cdot) + \lambda(d) \cdot \exp(-\frac{t - \mu(d)}{2\sigma(d)^2})$$

The iterative process by which we created these functions is as follows: We start from a constant function set to zero, $f(t) = 0$, which follows the intuitive assumption of *default state is non-stressed*. Note that the individuals in our study are not patients, and they were confirmed not to have any diagnosed condition affecting the prevalence of stressful episodes. then, we sweep the recorded entries and iteratively update the teacher function by adding functional components to it. Consider each entry similar to the sample depicted in Figure **??** as a record $d$. Each record contains an attribute titled `probe_datetime`, which can take two types of values:

1. *timestamp*: a single specific timestamp assigned to a reported episode of stressful response. Note that in our study, the individual's report in 15-minute units (e.g., 4:15pm).

2. *timespan*: a pair of timestamps, indicating participant's reported start and end-times.

The update step involves the operation below:

$$f(\cdot) \leftarrow f(\cdot) + \lambda(d) \cdot \exp(-\frac{t - \mu(d)}{2\sigma(d)^2}) \tag{3.1}$$

The parameters associated with each update functional, namely, $\mu$ and $\lambda$, are selected such that the resulting signals be qualitatively reasonable to a human reviewer. Figures 3.7. and 3.8 show case an example data window and its corresponding teacher function.

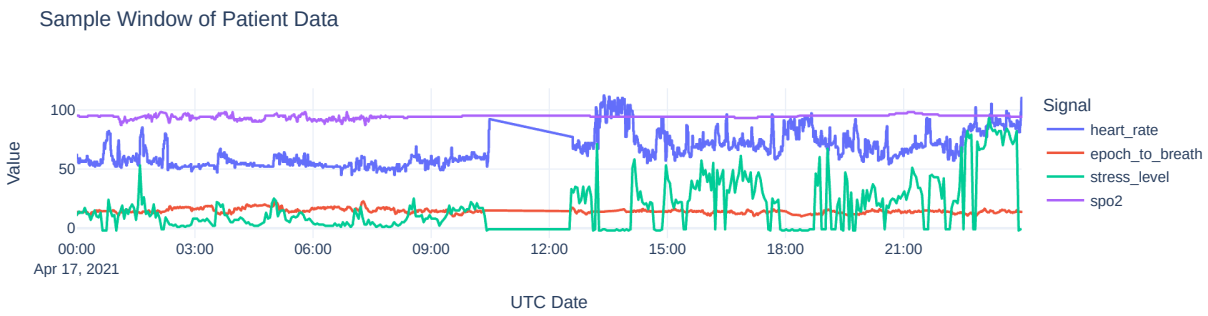



Figure 3.7: Example: A slice of patient physiological signals recorded via smartwatch system

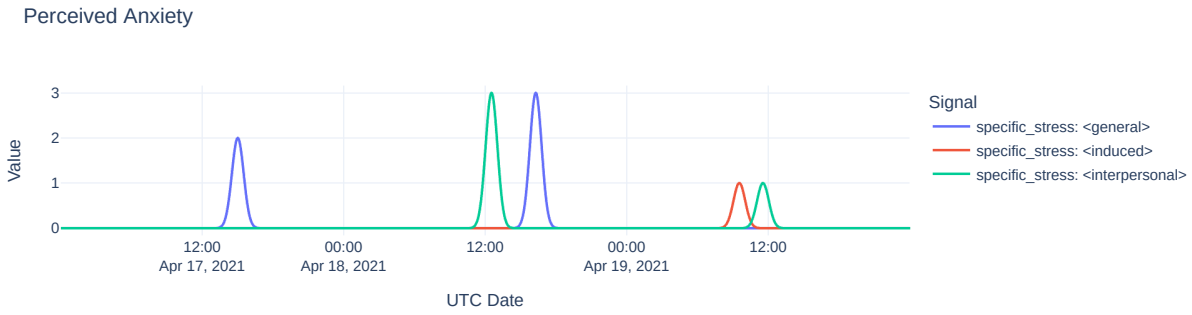



Figure 3.8: Example: Anxiety levels - Subject trajectories through time

### Representation Learning

**Event Observations** The first step towards preparing our inference pipeline is defining the *observation*s in this study, or, in other words, the input to our pipeline. We define our entity, the *episode*, as the one hour episodes, and our observation is the collection of sampled data corresponding to a 1-hour sensory reading episode, with data sampled from the aforementioned modalities. The sampled records contain longitudinal information on the aforementioned data modalities (heart-rate and activity aggregates, heart-rate variability, pulseox, and respiration) throughout the span of the

episode. To help with faster convergence and overall stability, we leveraged min-max normalization over the magnitudes and learned standardizers for reshaping the data. We define the task of predicting perceived stress below:

**Perceived Stress Task:** Given the information provided by 1-hour observation of user's physiological episode, determine whether or not the episode is one that ends in a stressful note, a.k.a, with $t_{\text{end}}$ denoting the episode's ending timestamp, determine whether or not the statement $f_{\text{u}}(t_{\text{end}}) > \tau$ is true.

### Fusion

When multiple data sources are involved in a single observation, one critical question would be the point at which the merging of information should take place. In this study, we offer two solutions in which we have early fusion for one representation pipeline and late fusion for the other.

**Early-fused Pipeline** The first approach to fuse the information is the simple choice of creating a *hybrid continuous token* by synchronizing and concatenating the information from all data sources. This approach is more intuitive, however, it is expected to render the system more prone to the impact of inaccurate data.

**Tackling Limited Data**: In our early-fusion pipeline, given that we are working with a more complex input and a single pipeline responsible to learn the corresponding joint distribution, we designed a targetted information removal objective to reduce the overfitting and help with the more efficient utilization of this data. To this end, we leverage gradient ascent to *unlearn* the information biasing the model towards the pecularities of data associated with individuals rather than the task. Consider a model acting as discriminator, denoted as $D(\cdot;\psi_1)$. The objective of this model is to leverage the latent representation resulting from processing the episode by our model, and try to distinguish between different participants given their IDs as the task. We consider our pipeline's stem as the core component of the $generator$ model, denoted as $G(\cdot;\psi_2)$, which is to challenge the discriminator in making the aforementioned subject classification task more cumbersome to carry

out. We implement $D(\cdot;\psi_2)$ as MLP-based multi-layer head applying on the latent representations:

$$D(\cdot;\psi_1) = f_{\text{adv}}(\cdot;\psi) \tag{3.2}$$

To build the optimization component, we simply add the loss term $\mathcal{L}_D - \mathcal{L}_G$ to the final loss, in which $\mathcal{L}_D$ and $\mathcal{L}_G$ are defined as below:

$$\mathcal{L}_G = E[l_{\text{ce}}(\text{fr}(f_{\text{adv}})(z(x)), y_{\text{sub}})] \tag{3.3}$$

$$\mathcal{L}_D = E[l_{\text{ce}}(f_{\text{adv}}(\text{sg}(z(x))), y_{\text{sub}})] \tag{3.4}$$

Note that in the above, $\text{fr}(\cdot)$ *freezes* the parameter set $\psi$ and sg is the stop-gradient operator, cutting off the gradient pathway to what comes before it.

**Late-fused Pipeline** While intuitive and simple to build, the early-fusion pipeline introduces its own problems in training and inference. In that setup, we need to make assumptions on regularity of the data sampling, no missingness[1], and the optimality of following a uniformly distributed importance weights over the input information coming from different sources. To explore an alternative solution, we propose a late-fusion pipeline, in which the data from each modality is represented by a dedicated neural network pipeline and mapped to a *shared* semantic space.

The observations representing our episodic one-hour entities are easier to build compared to the early-fusion pipeline. In this context, there is no need to additional transformations such as interpolation, upsampling, or synchronization. Each datapoint contains the information thoroughly representing the episode's timespan, denoted as $e = (t_{\text{start}}, t_{\text{end}})$, with the information we have in the corpus:

$$\boldsymbol{x}_{m,e}^{(p)} = \{x_{m,t}^{(p)} \in \mathbf{x}_m^{(p)} | t \in e\} \tag{3.5}$$

In this setup, we have a separate modality-specific dedicated encoder denoted as $f(\cdot;\theta_m)$. Given that our observation data comes from different modalities, for each modality $m \in [M]$,,

[1] In our case, we do causal interpolation to deal with that.

we have:

$$f(\cdot;\theta_m) : \mathcal{X}_m \to \mathcal{S} \quad \forall m \in [M] \tag{3.6}$$

Note that in the above equation, $\mathcal{S}$ is a shared semantic space allowing the encoders from different modalities to collaborate, teach each other, and contribute to the final learned representations. Finally, each encoder creates a representation $z_{m,e}^{(p)}$ for the corresponding data, computed as below:

$$z_{m,e}^{(p)} = f(\boldsymbol{x}_{m,e}^{(p)};\theta_m) \quad \forall m \in [M] \tag{3.7}$$

There are numerous reasons for which it is plausible to assume that the information from each modality and each episode can contribute differently to the final output, if considered by a bayesian optimal classifier. The quality of the information made available via smartwatch sensors can vary overtime (based on how the watch is worn, how subtle the experienced stressful episode was, and so on), which is a crucial matter to consider while deciding on the contribution of these modalities. To enable the model the make such decisions automatically and on the level of *instance* data, we designed a pooling pipeline based on attention pooling.

$$a_i^{(m)} \leftarrow g(\boldsymbol{z}_i^{(m)};\boldsymbol{\psi}) \quad \forall m \in [M] \tag{3.8}$$

Afterwards, a softmax operation is applied to make sure that the summation of the predicted contributions is $1.0$, in other words, the contribution matrix is right stochastic:

$$\alpha_i^{(m)} = \frac{\exp(a_i^{(m)})}{\sum_{j\in[M]} \exp(a_i^{(j)})} \tag{3.9}$$

The attention weights will then be used to tune the contribution of the semantics of each modality to the construction of the final aggregated representation embedding the entire episode. $z = \sum_{i=1}^{M} \alpha_i^{(m)} \cdot z_m$.

To observe the content similarity between the latent representation and the aggregated embed-

ding, we use cosine similarity which is a conventional choice in the contrastive learning domain.

$$\phi(\boldsymbol{u}, \boldsymbol{v}) = \frac{h(\boldsymbol{u})^T \cdot h(\boldsymbol{v})}{\|h(\boldsymbol{u})\|_2 \cdot \|h(\boldsymbol{v})\|} \tag{3.10}$$

The following objective which will be involved in the final optimization then injects the proper incentives and penalties, guiding the encoders to represent the data for each episode closer to each other and the aggregate representations. Note that our approach does not solely rely on bootstrapping and does leverage negative examples as well, therefore, training can take place without the use of stop-gradient operation. Given the difficulty of providing comprehensive data transformations that can apply on episodes without a guarantee that the semantic content will not be altered extensively, we chose this approach.

- *Pre-training*: Pre-training the model parameters by optimizing $\mathcal{L}_{\text{cl}}$ (defined below) through a longer pre-training sequence.

$$\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} -\log \frac{\exp(\phi(z_i^{(m)}, z_i)/\tau)}{\sum_{j \in \mathcal{B}, j \neq i} \exp(\phi(z_i^{(m)}, z_j)/\tau)} \tag{3.11}$$

  Afterward, start with the resulting weights as the initial point for the supervised fine-tuning of the model with the cross-entropy objective:

$$\mathcal{L}_{\text{cross-entropy}} = -\sum_{c \in \mathcal{C}} y_c \ln p_c \tag{3.12}$$

  In the equation above, $\mathcal{C}$ is the set of all classes (e.g., in our experiments, the two categories of stressed and non-stressed for each episode), and $p_c$ is the predicted probability of class $c$ for an observation, computed by passing representations through a final projection and Softmax layer.

- *Regularization*: Use $\lambda_{\text{reg}} \cdot \mathcal{L}_{\text{cl}}$ as a regularization term in the overall loss, and train the model

by optimizing this loss simultaneously as the supervised learning objective.

There are several points worth remarking upon with regard to the comparison of these two training schemes. To begin with, deciding whether pre-training is going to lead to better generalization performance versus the regularization-based approach depends on model complexity, availability of data, and the challenges of the specific task that one is targeting. That being said, the regularization approach is expected to be considerably faster than the two-stage pre-training and fine-tuning method, and in our experiments on the task of predicting stress labels, it led to better test performance as well.

We have detailed the main intuitive advantages of this approach below:

**Modularity:** Having a separate dedicated encoder for each modality and fusion via a shared semantic space allows a *modular* inference process. Our optimization scheme's reliance on the contrastive regularization focuses on optimizing individual branch's ability to generate representations optimal for the task.

**Knowledge Transfer:** The encapsulated view in the early phase of our pipeline allows one to leverage knowledge transfer and employ weights and models that serve as suitable initial points for training the pipelines. In addition, different optimizers and learning rate schedulers can be used for each branch per the requirements of the fine-tuning scheme.

**Plug-and-Play:** The shared semantic space also has the empirical advantage of allowing researchers and developers to apply the same pipeline with new modalities added or some removed, assuming a corresponding expert model is either trained or fine-tuned to allow the new modalities to work with the shared semantic space[2].

### Experimental Results

**Label Smoothing** In our experiments, we followed the strategy below in parameterizing the teacher function for every subject:

[2]Our analyses do show that the empirical contribution of data from different modalities is not a uniform distribution. Therefore, further research on larger corpora can shed more light on whether one modality can be used as the *central* data source in our observations, similar to the approach presented in [38] for self-supervised learning from multiple modalities.

$$\mu(d) = \begin{cases} d[\texttt{`probe\_datetime`}] & \text{if single} \\ \text{avg}(d[\texttt{`probe\_datetime`}]) & \text{if range} \end{cases}$$

$$\sigma(d) = \begin{cases} 30 \times 60 \text{ sec} & \text{if single} \\ \dfrac{\text{len}(d[\texttt{`probe\_datetime`}])}{(30 \times 60 \text{ sec})} & \text{if range} \end{cases}$$

In this work and as a proof of concept, we focused on predicting whether a window ends in a high-stress note. We considered the coefficient $\lambda(d)$ which adjusts the magnitude of each probe to be proportionate to the reported stress levels, and respectively defined the high-stress window as one that the corresponding teacher function returning a value larger than $0.5$ for its endpoint, which led to a high consistency between the resulting teacher functions and the reported episodes. The sensory data used in our work was based on heart-rate (every $15$ seconds), a measure of heart-rate variability (every $3$ minutes), pulse-oximeter (every $1$ minute), and respiration rate (every $2$ minutes). On the input side and to help stabilize the training further, we fit min-max normalizers on the features across the time slices in the train set.

**Modeling and Optimization** For the early-fusion pipeline, the recurrent neural network module we selected was a 4-layer bi-directional RNN in Long Short Term Memory (LSTM) configuration, leading to a $z \in R^{256}$ latent representation. To prepare the inputs for processing, we performed an early fusion of the sensory readings. We created a sequence of vectors representing physiological status at each timestamp, as it is a more intuitive approach for modeling the inputs in this case.

Our optimization protocol employed the Adam algorithm with a learning rate of 1e-3 and a weight decay of 1e-4 to help with overfitting. We also made use of cosine annealing scheduling, reducing the learning rate across our $100$ epoch experiments.

With regards to adversarial regularization for enhancing subject invariance, our $f_{\text{adv}}(\cdot; \psi)$ is a two-layer MLP mapping the latent representation to the subject identifier label. The results shown

Table 3.6: Performance overview for the task of recognizing high-stress windows (early fusion setup)

| | **Acc** |
|---|---|
| **Supervised Setup** | 62.0 |
| **Supervised Setup + Adversarial Subject Invariance** | 64.5 |

Table 3.7: Performance comparison for the trained pipeline under different learning setups

| **Method** | **Test Accuracy** |
|---|---|
| Early-fusion + Supervised Training + Adversarial | 64% |
| Late-fusion + Supervised Training | 66% |
| Late-fusion + Contrastive Pre-training + Fine-tuning | 70% |
| **Late-fusion + Supervised Training + Contrastive Regularization** | **73%** |

in Table 3.6 indicate an increase in the generalization performance.

In our late-fusion setup, focusing on our real-world perceived stress corpus, we conducted experiments under the main settings of 1) supervised training baseline, 2) pre-training the contrastive objective and fine-tuning via supervised objective, and 3) training the supervised objective and simultaneously optimizing a scaled version of the contrastive term as a regularizing loss.

We observed that leveraging more features and following a late-fusion protocol for combining modality representations did lead to an improved generalization performance over the supervised setup proposed in [35], which combined the features at the beginning of the pipeline. In the case of our cohort, training with contrastive regularization led to the best generalization on the unseen test data, and the results are shown in Table 3.7. Note that, in general, it is hard to say which self-supervised setup (pre-training versus regularization) is best, as it could depend on other factors, including model complexity, optimization, data availability, and task difficulty. That being said, our approach allows learning high-quality representations by optimizing the modality-contrastive objective via both of these setups.

Additionally, we investigated the interpretability and leveraged the task-specific attention mechanism in our pipeline, which pools the representations from different modalities, to study the utility and contribution of observations from each feature group. This enables the network to dynamically assign weights to each modality's latent representation (in the shared space) as it processes each in-

stance, allowing us to study their contribution both per instance and in expectation for performing the desired task. In Figure 3.9, we have shown the results on this matter for the contrastive regularization setup[3]. The results indicate that even though the contributions of the different modalities follow a non-uniform distribution as expected, none of them were ignored by the model and they all play a part in the final predictions.


Figure 3.9: The average contribution of the four modalities to the final episode embeddings.

## 3.4 Discussion

### TASC

In this work, we presented our results on a comprehensive end-to-end framework for recognition of episodic stressful response, distinguishing perceived stress from merely aroused physiological signals. The main application of this framework is in remote mental health monitoring, and is specifically suitable for short-term tasks. To deal with the challenges of remote health monitoring and representation learning, we designed specific optimization schemes leveraging techniques such as adversarial information removal and contrastive optimization so as to further enhance the learned representations. The multi-modal nature of the data raises the question of the optimal point to fuse the modality representations. To this end, we designed an architecture for each one of these design choices and showed that, empirically and on our cohort's data, late-fusion scheme worked

[3]The label `heart` in Figure 3.9 corresponds to the `daily` modality's information, given that its main focus is heart-rate.

better. In addition, it provided enhanced capabilities for inter-device compatibility and transfer learning. Nonetheless, our early-fused pipeline also demonstrated reasonable performance for a simple and intuitive model design. All in all, our framework provides a proof of concept for effective monitoring of mental health objectives happening in the short-term episodes, particularly when compared to the performance of current, commercially available analytic tools.

# Chapter 4

# AI Assurance

## 4.1 Introduction

In recent years, artificial intelligence (AI) has emerged as a transformative force, revolutionizing diverse sectors with its unparalleled capabilities. No longer confined to ancillary services, AI applications are increasingly being found in centralized roles of what might be deemed 'essential services' [40]. The integration of AI models into domains like healthcare, law enforcement, public works, and more, has brought about unprecedented opportunities for innovation, efficiency, personalization, and safety, but in-turn newfound urgency to the challenge of AI safety.

As AI capabilities continue to mature and models are incorporated into increasingly critical aspects of daily-life, it becomes imperative to establish robust techniques for the validation of these models [13]. This necessitates the development of a robust validation processes to ensure that AI models not only meet but exceed the stringent standards required for deployment in real-world scenarios [109].

This technical challenge goes hand-in-hand with the growing realization from regulatory and licensing agencies that increasing scrutiny over AI is needed [27], with a growing chorus of policy analysts advocating for a similar level of rigor applied to the review and approval of AI models that are currently applied to domains like hardware, pharmaceuticals, and medical devices that undergo

extensive, standardized, safety reviews prior to their release into public use [95].

Yet, despite the growing criticality of validating AI models, the task presents a formidable challenge. The current state of the art in AI model validation is characterized by complexities and uncertainties, stemming from the inherent nature of the training and validation of machine learning algorithms. A variety of efforts including trials, simulations, model-centered validation, and expert opinion, are employed during training to attempt to validate a model's performance [87]. However, what is notable about all these approaches is their dependency on internal developmental factors including a complete reliance on developer-generated training data and agents, to test and validate models for release. Post-market validation is also often present in best-practices, with various methods including failure monitors, safety channels, redundancy, voting, and input and output restrictions being employed to continuously validate the systems after deployment [87], but these methods, while crucial, have the drawback of being able to gather data only once a model is already in use, and failures have real-world consequences. More generally, in spite of these considerable efforts, the intricate interplay between data, algorithms, and the dynamic nature of the environments these models are deployed to, poses unique obstacles in ensuring the reliability and generalizability of AI models.

What is ultimately needed for a fully robust validation framework, is the addition of fully external and independent means of evaluating models, an analysis that can be effected independently by wholly unaffiliated researchers, laboring without the need for insights and resources from the model's originators.

In this paper, we delve into the intricacies of validating AI models, examining the challenges posed by an over-reliance on developer curated data sources and the lack of effective external mechanisms to validate models. We then introduce a novel mechanism for evaluating the performance of a model by contrasting the architectures of models. This paper aims to contribute to the discourse on the critical importance of AI models in society and the pressing need for advancing techniques to validate their efficacy, reliability, and ethical implications in the pursuit of enhancing the safety.

## 4.2 Background

### 4.2.1 Trustworthy AI

There are multiple definitions for AI. A widely accepted definition originates from IEEE-USA, who defines AI as follows: “Artificial Intelligence (AI) is the theory and development of computer systems that are able to perform tasks that normally require human intelligence such as visual perception, speech recognition, learning, decision-making, and natural language processing” [57].

It is particularly in the domains for which AI is called upon for decision-making, that trustworthiness of the models empowered to do so become of paramount importance. There has been considerable work done to-date to specify a precise definition for Trustworthy AI. While multiple definitions exist, for the purposes of this paper we will reference the National Institute of Standards and Technology (NIST) Artificial Intelligence Risk Management Framework, which defined the ”characteristics of trustworthy AI systems include: valid and reliable, safe, secure and resilient, accountable and transparent, explainable and interpretable, privacy-enhanced, and fair with harmful bias managed” [90]. From this definition we can extrapolate a working definition that Trustworthy AI refers to the development and deployment of artificial intelligence systems that exhibit reliability, transparency, fairness, and ethical considerations throughout their lifecycle.

While all the aforementioned elements of trustworthiness are critical, for the purposes of this paper, we will focus on the Accuracy/performance and robustness elements of a model’s trustworthiness.

#### Accuracy and Performance

The primary goal of any AI model is to accurately and effectively perform the tasks for which it was designed. This involves achieving high accuracy on relevant metrics, minimizing errors, and demonstrating consistent performance across diverse datasets. NIST speaks to this, formally defining accuracy as “closeness of results of observations, computations, or estimates to the true values or the values accepted as being true” [90].

**Robustness and Generalization**

An AI model's validity extends beyond its performance on the training data to its ability to generalize well to new, unseen data. Robust models maintain accuracy even in the presence of noise, outliers, or variations in the input data distribution. NIST defines robustness or generalizability more concisely as the "ability of a system to maintain its level of performance under a variety of circumstances" [90].

Robustness requires not only that the system perform exactly as it does under expected uses, but also that it should perform in ways that minimize potential harms to people if it is operating in an unexpected setting.

Validity and reliability for deployed AI systems are often assessed by ongoing testing or monitoring that confirms a system is performing as intended. Measurement of validity, accuracy, robustness, and reliability contribute to trustworthiness and should take into consideration that certain types of failures can cause greater harm.

### 4.2.2 Approaches to AI Model Validation

As previously established, having a mechanism to establish external validation of AI models is essential for ensuring reliability, robustness, and generalizability in real-world applications. Unlike internal validation processes that rely on developer-curated data and tools, external validation leverages independent datasets, benchmarks, and tools to provide an unbiased assessment of model performance. These tools can be categorized into several approaches, each addressing distinct facets of validation.

A fundamental method for external validation is dataset-based validation (more on that below), which utilizes standardized benchmarks to evaluate models. Popular datasets like ImageNet, COCO, GLUE, and OpenML are commonly used for a range of AI tasks, providing a well-recognized standard for performance comparisons.[33] These benchmarks are invaluable for assessing a model's accuracy and generalizability across tasks. However, such datasets are often domain-specific and may not fully capture the complexity of real-world conditions. Cross-

domain testing, which employs datasets from related but distinct domains, can further probe a model's ability to generalize. Additionally, adversarial datasets, such as those available through platforms like RobustBench, are increasingly used to test a model's robustness against intentional perturbations.[88] While these dataset-driven methods provide a strong foundation for external validation, their efficacy is inherently limited to domains where such benchmarks exist.

Another avenue for external validation lies in model auditing tools, which analyze specific aspects of model performance and behavior. Tools such as AI Fairness 360 (AIF360) assess fairness and identify biases within models[15], while explainability tools like SHAP, LIME, and Captum offer insights into how models make decisions[101]. For robustness assessments, tools like Foolbox and Cleverhans simulate adversarial attacks to test a model's resilience.[55] Comprehensive frameworks such as TensorFlow Model Analysis (TFMA) and MLflow support large-scale model validation workflows, allowing developers and external reviewers to examine key performance metrics in a structured manner.[98] While these tools provide actionable insights, interpreting the results often requires a deep understanding of both the model and the validation framework.

Synthetic and simulated testing environments represent a particularly powerful approach to external validation, especially for applications in controlled or high-risk environments. Synthetic data generation tools, such as Gretel and Mostly AI, allow researchers to test models under various hypothetical scenarios, while platforms like OpenAI Gym and CARLA offer simulation environments for reinforcement learning and autonomous driving, respectively.[98] These environments enable systematic evaluation of model performance under diverse conditions, offering control over variables that may be difficult to manipulate in real-world settings. Although these approaches provide valuable insights, the representativeness of synthetic or simulated data remains a concern, as they may fail to fully replicate the complexity of real-world scenarios.

Post-deployment monitoring tools are also essential for continuous validation of models in production. Platforms such as Datadog and Evidently AI monitor real-world performance, detecting issues like data drift and concept drift that can degrade model accuracy over time.[43] Drift detection tools like WhyLabs and Alibi Detect play a crucial role in identifying and responding to

changes in input data distributions.[43] These tools are particularly valuable for ensuring long-term reliability, though they are inherently reactive, identifying issues only after they have begun to impact performance.

Independent audits and certifications provide an additional layer of external validation by ensuring models meet industry standards and regulatory requirements. Frameworks such as the National Institute of Standards and Technology (NIST) AI Risk Management Framework and ISO/IEC 23053 establish guidelines for evaluating AI systems' safety, robustness, and fairness.[133] Organizations like AlgorithmWatch offer third-party audits that evaluate models independently of their developers.[107] These certifications lend credibility and transparency to AI systems but can be resource-intensive and time-consuming to obtain. Finally, regulatory and ethical validation frameworks are increasingly available to ensure compliance with legal and ethical standards but also build user trust, which is critical for widespread adoption of AI systems.

In summary, external validation of AI models employs a diverse set of tools and methodologies, ranging from dataset-based approaches and synthetic testing to independent audits and regulatory compliance. These methods address distinct aspects of validation, including robustness, fairness, generalizability, and ethical considerations. While each approach has its limitations, together they form a comprehensive framework for assessing the reliability and safety of AI models in real-world applications.

## 4.3 Model Validity Experiment

Despite the recognized importance of establishing the validity of AI models in generlized contexts (i.e., that a model will continue to perform to a predetermined level of quality outside it's training domain), defining an objective, universally applicable measure is a challenging task.

Even setting as an assumption that different stakeholders who may prioritize various aspects of model performance differently can settle upon a universally accepted set of validation metrics that encapsulate all relevant dimensions of validity, the validity of an AI model is also context-

dependent, varying based on the application domain and the specific requirements of users. A one-size-fits-all approach to validation may simply not be feasible, given the diverse applications of AI across sectors [96].

More intrinsically, a critical limitation when determining the universal applicability of a model is its general reliance on the dataset it was trained on [111]. These data, often collected and curated by the model-developer themselves, is generally comprised of a highly specific set of features and transformations, collected under unique circumstances, that are virtually impossible in many instances to recreate independently. This makes it difficult to replicate and reproduce the model independently, a hallmark of the peer-review process in most disciplines.

Furthermore, owing to Machine Learning's overriding goal of maximizing predictive performance, and not establishing any statistically rigorous assertion of an underlying relationship between features and labels, we do not typically have the capacity to leverage statistical hypothesis testing to assert the generalizability of our models. Validation efforts can try to some statistical rigor through simulations [87] but such simulations remain dependent on the initial dataset to define the distribution of feature-values in an augmented dataset.

To be able to assert that performance observed on a (typically) small subset of training data will continue to bear true outside of the training environment, researchers typically default to the classic Train/Test split where data is purposefully withheld from the model training in order to test it's robustness against data it hasn't previously encountered as a mechanism to infer its generalizability [111]. While this is often an effective proxy, and tools like cross-validation, or model manipulations like regularization, are employed to further help reinforce this approach, its underlying validation's dependence on the dataset it was trained on, cannot be fully eliminated, nor can the broader effort to achieve independent verifiability or reproducibility be achieved.

In summary, while there is some consensus on what the critical elements of a robust AI model validation might be, the inherent subjectivity, context dependence, and dependence on limited datasets, pose formidable challenges in implementing an objective and universally applicable measure of validity.

One potential approach to addressing this challenge could emerge, not from the data used to train the model, but rather the resulting model architecture itself. While the formation of a model architecture is almost entirely data dependent, we have some inclination that the structure of the model itself (e.g., an overtrained model with high variance), could yield valuable insights into the performance of that model.

This Cross-model validation could offer another innovative approach to external evaluation. By comparing the activations or architectures of a newly trained model against a well-validated baseline, we can potentially infer robustness and generalizability. We explore the potential of studying one model, a relatively unknown one, against the context of another, known and validated model to ascertain whether similarities in neuronal-level correlation, can be sufficient to determine aspects of robustness.

### 4.3.1 Model Robustness

Neural networks have been widely applied in security applications including spam and phishing detection, intrusion prevention, and malware detection. This black-box method, however, where models are deployed without an understanding for their underlying mechanisms of discriminant classification, is shown to be vulnerable to adversarial attacks [110]. There are two types of adversarial attacks: white-box and black-box. In white-box attacks, the attacker has complete knowledge of the target model and data. Among the existing white-box attack techniques, Fast Gradient Sign Method (FGSM) [73], DeepFool [86], and Projected Gradient Descent (PGD) [77] are the most prominent and widely adopted methods. These techniques primarily leverage the gradient of the loss function in their algorithm [31]. Conversely, in black-box attacks, only the model weights and data are accessible. Hence, some black-box methods opt to utilize features extracted from images instead of relying on model properties [122].

Many of the proposed adversarial attacks including per-sample and universal have demonstrated the ability to transfer and generalize effectively to unknown models targeting either classification or interpretation [31, 94]. This fact is notable as these models are trained on completely

independent data, and often focus on entirely unique domains. That this transferability is possible is largely attributed to the inherent underlying structural similarities among various neural networks, which arise from the common use of gradient descent-based optimization techniques during the training process [31].

The apparent success of cross-model attack techniques implies that the potential exists to establish a correlation score between neural networks, and more specifically, between well-established neural networks with more extensively audited and validated performance, and newly trained models lacking a similarly robust track record. Were such a correlation established, it could provide us with meaningful insights into the potential robustness and generalizability of a newly trained neural network prior to taking it into production.

To that end, in this paper, we introduce a novel methodology to accurately estimate this metric. We detail the technical approach for establishing a cross-model neuronal correlation. Next, we explore the implications of this score in validating an unknown model by leveraging the validation of another established model.

### 4.3.2 Methods

The correlation between the output vectors of two neurons $N_i$ and $N_j$ with respect to the input vector $X$, is determined by the Pearson correlation coefficient $\rho(.,.)$ computed by

$$\rho(A, B) = \frac{cov(A, B)}{\sigma_A \sigma_B}, \tag{4.1}$$

where $cov(.,.)$ denotes the covariance function and $\sigma_.$ is the standard deviation.

As suggested by [62], the neuronal correlation of the $l$-th layer's representation $R_l$ is

$$Corr(R_l) = \frac{1}{n(n-1)} \sum_{i,j=1;\ i \neq j}^{n} |\rho(R_{li}, R_{lj})|, \tag{4.2}$$

where $i$ shows the activation of the $i$-th neuron and $n$ is the dimension of $R_l$.

This existing methodology is limited to comparing correlations within a neural network and cannot provide a representation of two distinct neural networks. To extend the definition to allow for correlations across networks, we iterate through each neuron $N_i$ in the first network and determine the most correlated neuron $N_j$ in the second network. The correlation score of $N_i$ denoted as $S_i$ is derived by

$$S_i = \frac{\rho(N_i, N_j)}{|\text{layer}(N_i) - \text{layer}(N_j)| + 1}. \tag{4.3}$$

The inclusion of the denominator in the formula is crucial, as it accounts for the expected decrease in correlation between neurons that are more distant from each other in the respective architectures. The score is estimated for all neuron pairs. Following that, the average over all $S_i$ and $S_j$ values determines the correlation score between the two neural networks in question. Given that $\rho$ falls within the interval of -1 to 1, the resultant correlation score similarly spans from -1 to 1. It should be noted that we calculate the correlation function specifically for neurons in the hidden layer, as the last linear layer does not contribute significantly to understanding the network's behavior and interpretability.

Computing the correlation between all neurons in two networks can be extremely time-intensive, especially for larger architectures. To address this challenge, we also propose a more efficient approach which involves calculating partial correlations, particularly for networks that share some structural similarities. For instance, when comparing two networks with differing layer counts but matching output sizes in certain layers, the correlation analysis can be confined to these comparable layers. While this method sacrifices some accuracy, it offers a significant improvement in computational efficiency, making it more practical for analyzing larger and more complex neural networks.

Additionally, we explore the vulnerability of neural networks to black-box adversarial attacks and examine how this susceptibility relates to our proposed metric. To investigate the architectural relationships between different neural networks, we adapted the attack method introduced by blackbox, with slight modifications to their original approach. This black-box method facilitates model comparison without necessitating prior knowledge of their architectures or training datasets.

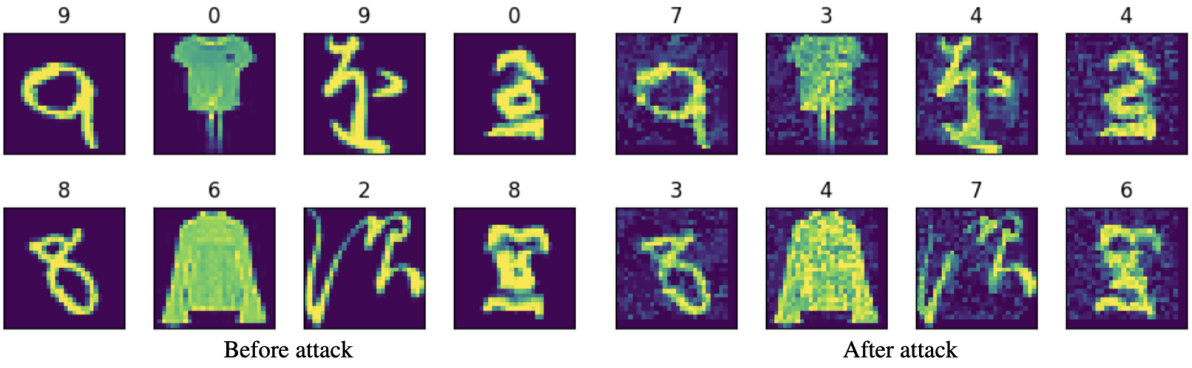



Figure 4.1: Test images and their predicted classes before and after attack.

As described in Algorithm 1, at each iteration, we train a simple model like a two-layer CNN that differs from the other models used in the experiments on the dataset. Subsequently, we adjust the data using the perturbation computed via the Fast Gradient Sign Method (FGSM) [73]. We iterate through the process multiple times to generate our ultimate adversaries. In our method, we solely utilize test data instead of custom-crafted data, as the former remains untouched during training. Furthermore, we refrain from altering labels based on the oracle's feedback and discard intermediate adversaries generated throughout the process, retaining only the final iteration to reduce the time and space complexity.

**Algorithm 1** Substitute DNN Training

**Input:** Training data $D = (x_i, y_i)_{i=1}^{n}$, simple two-layer CNN $C$, perturbation norm $\epsilon$

**Initialize** perturbed data $P_0$ with $X$

**for** epoch $\in \{1, \ldots, N_{ep}\}$ **do**

**Train** $C$ on $P_{epoch-1}$

**Update** $P_{epoch} \leftarrow \{\big(\tilde{x} + \epsilon \cdot \text{sgn}(J_C(\tilde{x}), y\big) : (\tilde{x}, y) \in P_{epoch-1}\}$

**end for**

**Return** $P_{N_{ep}}$

### 4.3.3 Results

Our numerical experiments were designed to investigate the performance of our proposed correlation method across two scenarios: small neural networks and large neural networks. This approach allowed us to comprehensively evaluate the method's efficacy and scalability.

#### A. Small neural networks

We evaluated our method on five simple fully connected networks (FCNs) and a two-layer convolutional neural network (CNN). The number of neurons considered for each hidden layer is described in Table 4.1. These networks were trained on four datasets including MNIST [66], Fashion-MNIST [126], Kuzushiji-MNIST [26], and Afro-MNIST [125]. All these datasets contain 60000 training examples and 10000 test examples. Each example comprises a $28 \times 28$ image paired with a classification label from a set of 10 categories. MNIST, Fashion-MNIST, Kuzushiji-MNIST, and Afro-MNIST contain images of handwritten numerals, fashion, Hiragana characters, and African numbers respectively. The selection of datasets with similar attributes was deliberate as our focus was on understanding the interplay between networks, irrespective of data size or category.

Table 4.3 illustrates the robustness of FCNs against the black-box attack detailed in Algorithm 1. Effect of the attack on some test images is illustrated in Figure **??**. Results for CNN are also provided, showcasing its inferior robustness, given its role in generating the adversaries. In our

Table 4.1: Description of FCNs.

| Network | Number of neurons in hidden layer |
|---|---|
| Net1 | [100] |
| Net2 | [60, 40] |
| Net3 | [40, 60] |
| Net4 | [196, 392] |
| Net5 | [20, 20, 20, 20, 20] |

Table 4.2: Correlation between FCNs.

(a) MNIST

| | Net1 | Net2 | Net3 | Net4 | Net5 |
|---|---|---|---|---|---|
| Net1 | 1 | 0.644 | 0.646 | 0.642 | 0.534 |
| Net2 | X | 1 | 0.697 | 0.699 | 0.564 |
| Net3 | X | X | 1 | 0.730 | 0.551 |
| Net4 | X | X | X | 1 | 0.541 |
| Net5 | X | X | X | X | 1 |

(b) Fashion-MNIST

| | Net1 | Net2 | Net3 | Net4 | Net5 |
|---|---|---|---|---|---|
| Net1 | 1 | 0.684 | 0.666 | 0.671 | 0.614 |
| Net2 | X | 1 | 0.724 | 0.724 | 0.664 |
| Net3 | X | X | 1 | 0.747 | 0.639 |
| Net4 | X | X | X | 1 | 0.604 |
| Net5 | X | X | X | X | 1 |

(c) Kuzushiji-MNIST

| | Net1 | Net2 | Net3 | Net4 | Net5 |
|---|---|---|---|---|---|
| Net1 | 1 | 0.572 | 0.556 | 0.581 | 0.510 |
| Net2 | X | 1 | 0.623 | 0.619 | 0.512 |
| Net3 | X | X | 1 | 0.679 | 0.538 |
| Net4 | X | X | X | 1 | 0.530 |
| Net5 | X | X | X | X | 1 |

(d) Afro-MNIST

| | Net1 | Net2 | Net3 | Net4 | Net5 |
|---|---|---|---|---|---|
| Net1 | 1 | 0.576 | 0.573 | 0.588 | 0.373 |
| Net2 | X | 1 | 0.695 | 0.705 | 0.500 |
| Net3 | X | X | 1 | 0.699 | 0.485 |
| Net4 | X | X | X | 1 | 0.458 |
| Net5 | X | X | X | X | 1 |

specific scenario, we selected a norm value of $\epsilon = \frac{10}{255}$ and $N_{ep} = 10$. The results indicate that, on average, Net5, Net2, Net4, Net3, and Net1 ranked from vulnerable to robust. Among the FCNs, Net5 exhibited the lowest overall accuracy. Notably, according to Table 4.2, Net5 displayed the weakest correlation with the other networks, aligning with its subpar performance. Conversely, Net3 demonstrated the highest average correlation score with the other networks, potentially attributed to its architectural similarity with the other FCNs.

Table 4.3: Accuracy of FCNs before and after attack.

| Network | Metric | MNIST | Fashion-MNIST | Kuzushiji-MNIST | Afro-MNIST |
|---|---|---|---|---|---|
| Net1 | Accuracy before attack | 96.2% | 86.8% | 84.4% | 99.2% |
| | Accuracy after attack | 73.7% | 59.5% | 50.4% | 45.1% |
| | Relative degradation | 23.4% | 31.5% | 40.3% | 54.5% |
| Net2 | Accuracy before attack | 96.6% | 86.1% | 84.4% | 99.4% |
| | Accuracy after attack | 65.7% | 56.9% | 47.4% | 44.3% |
| | Relative degradation | 32.0% | 33.9% | 43.8% | 55.4% |
| Net3 | Accuracy before attack | 96.0% | 84.2% | 83.8% | 99.1% |
| | Accuracy after attack | 69.4% | 53.3% | 48.2% | 44.5% |
| | Relative degradation | 27.7% | 36.7% | 42.5% | 55.1% |
| Net4 | Accuracy before attack | 96.1% | 86.8% | 84.5% | 99.2% |
| | Accuracy after attack | 67.4% | 58.1% | 47.2% | 45.0% |
| | Relative degradation | 29.9% | 33.1% | 44.1% | 54.6% |
| Net5 | Accuracy before attack | 92.9% | 77.5% | 74.3% | 98.5% |
| | Accuracy after attack | 53.5% | 59.5% | 39.6% | 40.3% |
| | Relative degradation | 42.4% | 23.2% | 46.7% | 59.1% |
| CNN | Accuracy before attack | 98.9% | 89.9% | 91.7% | 100.0% |
| | Accuracy after attack | 70.8% | 59.5% | 33.9% | 27.1% |
| | Relative degradation | 28.4% | 33.8% | 63.0% | 72.9% |

In Table 4.4, the correlation between FCNs and CNN is investigated. Here, due to the large number of neurons involved, we had to limit the correlation analysis to only $10$ test data points. Looking at the correlation values between CNN and the different networks, Net3 consistently demonstrates the highest correlation values across multiple datasets. This high correlation between Net3 and CNN suggests that their predictive patterns and feature extraction align closely across various datasets. In addition, it's noteworthy that Net5 exhibits the lowest correlation among the networks, which aligns with expectations due to its inferior performance and robustness.

Table 4.4: Correlation between CNN and other networks.

| Dataset | Net1 | Net2 | Net3 | Net4 | Net5 |
|---|---|---|---|---|---|
| MNIST | 0.880 | 0.856 | 0.908 | 0.831 | 0.416 |
| Fashion-MNIST | 0.756 | 0.822 | 0.820 | 0.755 | 0.317 |
| Kuzushiji-MNIST | 0.900 | 0.769 | 0.910 | 0.894 | 0.519 |
| Afro-MNIST | 0.588 | 0.629 | 0.730 | 0.718 | 0.059 |

Table 4.5: Partial correlation between ResNets.

| | ResNet-18 | ResNet-34 | ResNet-50 | ResNet-101 | ResNet-152 |
|---|---|---|---|---|---|
| ResNet-18 | 1 | 0.639 | 0.203 | -0.263 | 0.054 |
| ResNet-34 | X | 1 | 0.267 | -0.349 | 0.015 |
| ResNet-50 | X | X | 1 | -0.534 | -0.407 |
| ResNet-101 | X | X | X | 1 | -0.688 |
| ResNet-152 | X | X | X | X | 1 |

**B. Large neural networks**

The proposed correlation metric was also investigated on different variants of ResNet, including ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152 [49]. We used the pretrained weights on ImageNet [33] as our baselines. Since investigating the correlation on all of the neurons in all layers was not possible we only tested the method on the first output of the fourth layer on all ResNets. Also, we used only 10 test data points from the validation set of ImageNet. Considering the absolute values of the correlations, results suggest that the most similar networks to ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152 are ResNet-34, ResNet-18, ResNet-101, ResNet-152, and ResNet-101. This pattern illustrates that networks with a similar number of layers exhibit stronger partial correlations, serving as a validation of the effectiveness of our proposed correlation metric.

### 4.3.4 Discussion

These experimental results indicate that our proposed correlation score can effectively demonstrate connections and relationships across diverse neural networks. While these findings are still preliminary, we assert that this score can potentially offer insights into various factors including:

- **Architectural size impact:** Smaller architectures tend to exhibit higher correlation with other networks due to their simplicity and shared properties with most neural networks. As networks increase in complexity, this correlation diminishes, reflecting the nuanced decision-making processes that diverge from standard neural network behaviors. Calculating partial correlations, especially for the final layers of the networks, can yield more interpretable

insights.

- **Performance implications:** Lower correlations may correlate with poorer model performance, indicating that neuron activations deviate from the expected patterns necessary for accurate decision-making.
- **Robustness:** A strong correlation with a robust model may suggest a level of robustness within another model as well.

We further note that despite the valuable insights provided by the proposed correlation method, a principle drawback lies in the suboptimal performance in terms of time complexity for obtaining this score. Moreover, when dealing with larger models, a more efficient approach for calculating correlation with improved time complexity and precision may be necessary. Another limitation of our current methodology is the inability to pinpoint the exact reasons behind a low correlation score. However, if a high correlation is observed between two networks, where one is recognized for its accuracy and robustness, there is a strong indication that the other network is also performing well and likely to generalize. Nevertheless, this inference is not entirely precise.

## 4.4 Data Sufficiency Experiments

### 4.4.1 Introduction

The sufficiency of a dataset, both in terms of size and representativeness, is critical to training an effective Machine Learning model. Failure to train on data of adequate quality is likely to result in models that dramatically under-perform in production environments.

Training machine learning models on inadequate data presents several challenges that can significantly impact model performance and generalizability. While the problem of 'inadequacy' of data is a multifaceted one, one significant component of it is a lack of sufficient data volume, which in-turn likely impacts the representativeness of the data sample towards the overall population undermining the ability of a model to learn meaningful patterns, resulting in overfitting, where the

model performs well on the training data but struggles to generalize to unseen data leading to poor outcomes when deployed in real-world scenarios . This owes to the fact that if the dataset used for training does not accurately reflect the conditions and distributions the model will encounter in real-world applications, the model will struggle to generalize beyond the narrow scope of its training. Consequently, the model's assumptions about the domain may be incorrect, leading to suboptimal decision-making in practical applications.

In cases where the dataset is particularly small or inadequate, machine learning models—especially complex ones like deep learning architectures—fail to learn the intricate relationships within the data. Complex models require large amounts of data to effectively capture subtle patterns. Without sufficient data, the model may not converge, or it may underfit, meaning it performs poorly on both the training and test sets due to an inability to learn the underlying patterns.

Moreover, the use of inadequate data increases variability in model performance. Small datasets are more sensitive to changes in initialization values, training procedures, or even small differences in the data samples used during training. As a result, models trained on inadequate data exhibit inconsistent and unpredictable behavior, making them unreliable in different testing or production environments. Issues such as overfitting, bias toward dominant classes, poor handling of noisy data, and a lack of generalizability can result from insufficient, non-representative, or poor-quality data. Ensuring access to high-quality, diverse, and sufficiently large datasets is essential for developing models that are both robust and reliable in their predictions.

Having a reliable means of determining the quantity of training data required to effectively train a model would be an incredibly useful tool for researchers to have. This owes both to the upfront challenges and associated costs of data collection, and, with the advent of increasingly costly training runs, the investment costs associated with training.

This is a widely recognized need across research domains, with many experimental protocols requiring power analyses, or equivalent statistical studies to prospectively determine the requisite amount of data collection required to run studies effectively.

However in spite of the obvious utility of such an assessment, it remains elusive for Machine

Learning model developers. As already noted, many methods for external dataset-based validation but are post-hoc model dependent.

### 4.4.2 Background

#### Prospective Analysis of Data Sufficiency

In most academic disciplines, conducting a sufficiently robust study design is arguably more important than the statistical analysis that succeeds it. After-all, a poorly designed study may be unsalvageable after the fact, whereas a poorly analyzed study can simply be re-analyzed once errors in methodology are determined.

A critical component in study design is the determination of the appropriate sample size. The sample size must be large enough to establish statistical significance of any potential findings, yet not so large as to unnecessarily burden researchers with the (often costly) acquisition of data.

Attempting to determine data sufficiency for Machine Learning (ML) models is a particularly vexing challenge given the nature of Machine Learning. Unlike statistics, ML does not attempt to asset any factual relationship between label and feature set. Rather ML models are almost mercenary in nature. Charged with best accomplishing a given prediction or classification task, irrespective of any actual underlying statistical relationship.

This relaxed focus allows more a more expansive inclusion of features for whom a statistically significant relationship to the label may not exist

#### Current approaches to determining sufficient sample size

#### Statistical Effect size

#### Hypothesis

This research attempts to ascertain whether or not certain descriptive statistical features of a dataset can be indicative of prospective model performance, and can provide a heuristic for data volume required to achieve certain generalizable performance benchmarks.

Corollary: The magnitude of Feature Effect size, and data volume, both impact model performance and generalizability, with a higher effect-size offsetting the need for large datasets, while a smaller effect size generally requiring a greater volume of data to achieve similar results.

Corollary: There is an upper-bound to performance irrespective of data size, and possibly effect size (i.e., as data size goes to infinite model performance plateaus).

The goal of this research: If a researcher has advanced knowledge the size of the dataset employable for training, and can calculate the Effect Size of features with respect to the labels, one can make a reasonable assumption on projected model performance.

To best explore this overall objective, we conducted two specifics experiments to determine whether or not such a relationship exists.

Hypothesis 1: There is a correlation between statistical effect-size and model performance up to a point.

Hypothesis 2: There is a correlation between statistical effect-size, data size and model performance up to a point

These trends are universal and applicable across datasets.

### 4.4.3 Methods

For this study, a dataset containing a sampling of adult census data was employed. The dataset was sourced from the UCI Machine Learning Repository and includes 48,842 observations across 14 features.

The following were the features along with data types:

Opting for a binary classification problem, the label for this study was income, which was segmented into a binary categorical segmented as either greater than 50k or less than or equal to 50k per year.

Standard preprocessing of the data was undertaken which included removing rows with null values, encoding categorical variables to ensure compatibility with specific models, and normalized numerical features. Categorical features were encoded using LabelEncoder, converting each

Table 4.6: Features in the Census Dataset

| Feature Name | Description | Type |
|---|---|---|
| Age | Age of the individual | Numerical |
| Workclass | Type of employment | Categorical |
| Fnlwgt | Final weight; a census measure | Numerical |
| Education | Highest level of education completed | Categorical |
| Education-Num | Number of years of education | Numerical |
| Marital-Status | Marital status | Categorical |
| Occupation | Type of occupation | Categorical |
| Relationship | Relationship to head of household | Categorical |
| Race | Race of the individual | Categorical |
| Sex | Gender of the individual | Categorical |
| Capital-Gain | Income from investment | Numerical |
| Capital-Loss | Losses from investment | Numerical |
| Hours-Per-Week | Average hours worked per week | Numerical |
| Native-Country | Country of origin | Categorical |
| Income | Income level (target variable) | Categorical |

category into a numerical value. All numerical features were standardized using StandardScaler. The standardized numerical features and encoded categorical features are combined into a single feature matrix, which is then used for training and evaluating the machine learning models. Following the preprocessing step, a total of 33,000 samples remained.

The following set of experiments was then performed:

**Experiment 1: Determine if Effect Size Correlates to Performance Across datasets and feature mixes**

**Experiment 1.1: Different subsets of data, with the same set of Features, (i.e., different mix of rows within the same dataset).**

For this experiment the dataset was segmented into 66 subsets, each with 500 rows. All features remained identical across subsets, with only individual rows varying across datasets.

**Experiment 1.1.1:** Model performance was compared across the same set of features/different data.

For this experiment the **"Income Greater Than 50k"** binary categorical feature served as

label, with the rest of the features serving as the feature set

For each data subset: the averaged effect size was calculated for the binary label we will aim to classify.

**Effect Size calculation** The effect size for numerical features was calculated using **Cohen's d**, which measures the standardized difference between the means of two groups (e.g., individuals with income greater than $50K and those with less than or equal to $50K). For categorical features, the effect size is calculated using the **Odds Ratio** derived from a contingency table that compares the frequency distribution of the categories across the target variable (income).

After calculating the effect size for each individual feature, the average effect size across all features within a subset is computed. This average provides a single summary metric representing the overall effect size of the dataset in that particular subset.

An identical sequence of classifiers was trained on the models, employing different classification techniques. This was done to try and generalize the approach across learning model paradigms. Specifically, for every experiment, we employed the same four machine-learning models:

1. Logistic Regression
2. Decision Tree
3. Random Forest
4. Neural Network.

This yielded 66 data points of model performance vs. averaged effect size, allowing us to compare those two elements by calculating a direct correlation.

**Experiment 1.1.2:** Attempted to employ a Different Feature mix on the same rows, (i.e., utilizing the same hundred subsets of data but with a different set of features and labels.) Specifically the income label was reintroduced into the feature-mix and this time the "sex" binary feature was employed as the label. The same set of models were run and performance metrics calculated, yielding another 66 datapoints.

**Experiment 1.2: Compare model performance across a mix features with same underlying data Experiment 1.2.1:** For this experiment, model performance across a differing set of features with the same underlying data was compared. In this experiment the **income greater than 50k** binary was retained as a label, while the rest of the features were retained as a feature set. For this experiment, a series of subsets were devised. For each subset one of the features was dropped, and then the averaged effect size for the binary label was recalculated on the reduced feature space. An identical sequence of classifiers was then trained, and model performance and associated averaged effect size were calculated.

This produced 14 data points of model performance vs. effect size.

**Experiment 1.2.2**: In a parallel fashion to experiment 1.1.2, we reran 1.2.1, but this time used the **sex** binary label as a feature. An additional 14 data points were generated.

**Experiment 2. Compare the relationship between effect size and learning curve slope**

This experiment sought to determine if there a relationship between the slope of a learning curve (the rate at which a model's error rate decreases with respect to training data size) and the effect size, irrespective of overall model performance. The hypothesis being tested was whether or not a more dramatic effect size would indicate a cleaner dataset, requiring fewer samples for a model to 'bottom out' in terms of error. The critical distinction here was that it wouldn't necessarily predict model performance, but rather how large a dataset was required for convergence on an optimal model.

For each model in experiment 1 where a previously calculated effect size was obtained, a learning curve/error rate for both training and validation sets was plotted.

**Experiment 2.1 Correlation of Effect Size to Slope of Learning Curve** For each model an associated learning curve on a validation set was plotted. This model generally adhered to a logrithmic function with an associated coefficient, beginning elevated, but converging on an optimal error rate with the addition of more training data. This slope was then plotted against the associated Effect Size and a correlation computed.

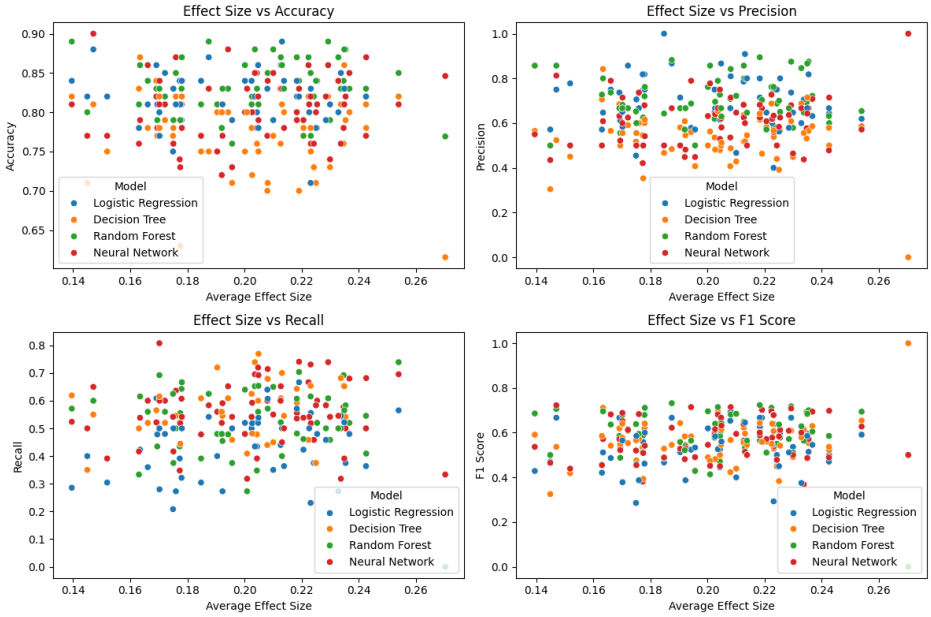



Figure 4.2: Experiment 1.1.1 (income as label) R2 Value: 0.0132

**Experiment 2.2 Correlation of Effect Size to Ratio of Error between Training and Validation Sets** For each model, in addition to learning curve on the validation set, a learning curve on the training set was also plotted. These models tended to start with virtually no error, as a small amount of training data allowed for significant overfitting. As more training data is introduced, the error rate increases until it converges with the validation set's error rate. Pairing both sets of plots, for each point, the magnitude of the difference in error between the training and validation sets was calculated. This resulted in a linearly decreasing amount as error rates converge. Then take slope of this linear line representing the magnitude of error difference was then extracted and correlated against Effect Size.

### 4.4.4 Results

**Experiment 1:** The results indicate an exceptionally weak correlation between effect size and model performance.

Table 4.7 details the experimental results.

The data indicates that effect size does not predict the outcome of model performance well.

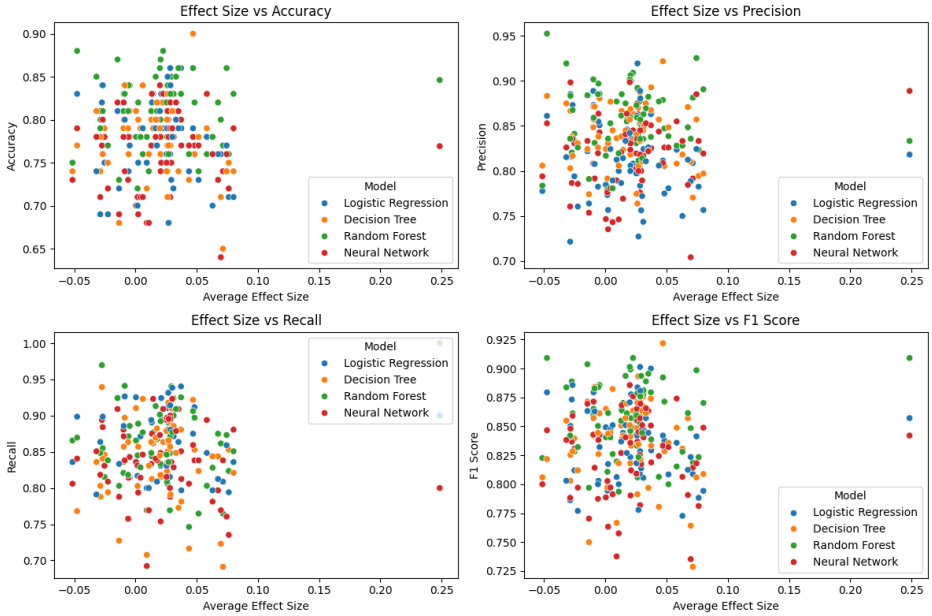



Figure 4.3: Experiment 1.1.2 (gender as label) R2 Value: 0.0011

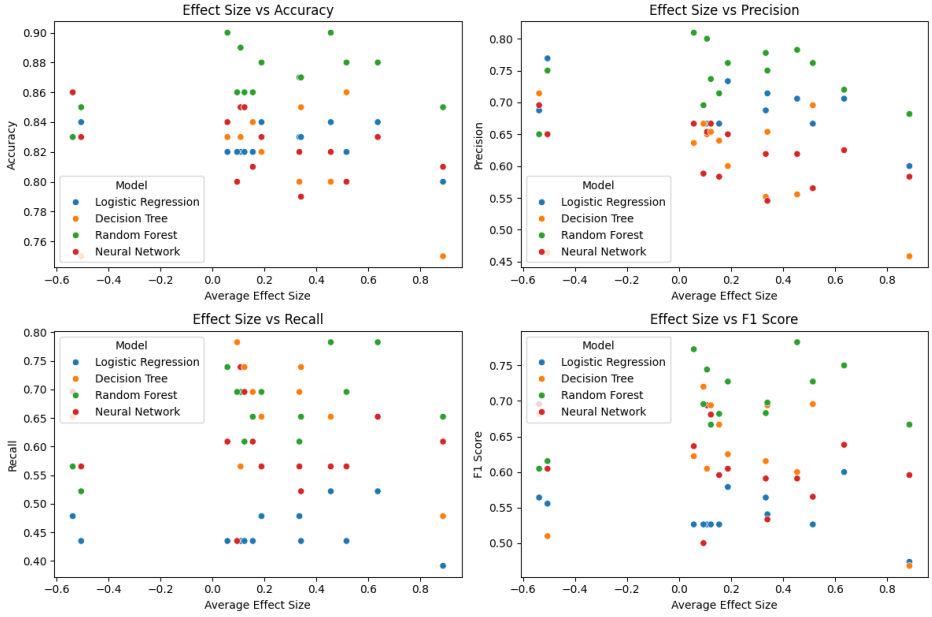



Figure 4.4: Experiment 1.2.1 (income as label) R2 Value: 0.0142

| Exp. No. | Experiment Name | R2 Value |
|---|---|---|
| 1.1.1 | Income label, identical features, different subsets | 0.0132 |
| 1.1.2 | Sex label, identical features, different subsets | 0.0011 |
| 1.2.1 | Income label, different feature mix | 0.0142 |
| 1.2.2 | Sex label, different feature mix | 0.091 |

Table 4.7: Summary of Results for Experiment 1

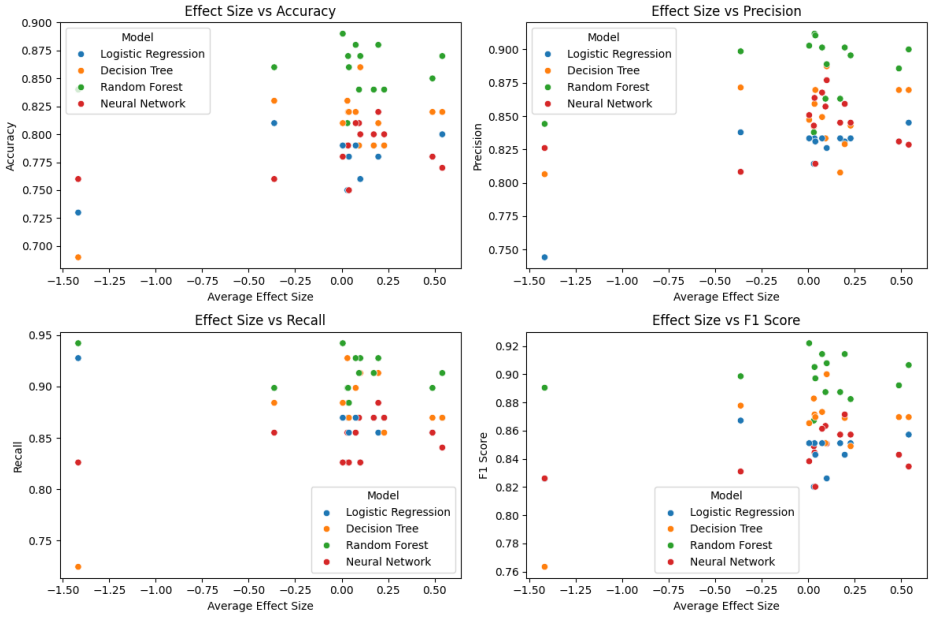



Figure 4.5: Experiment 1.2.2 (gender as label) R2 Value: 0.091

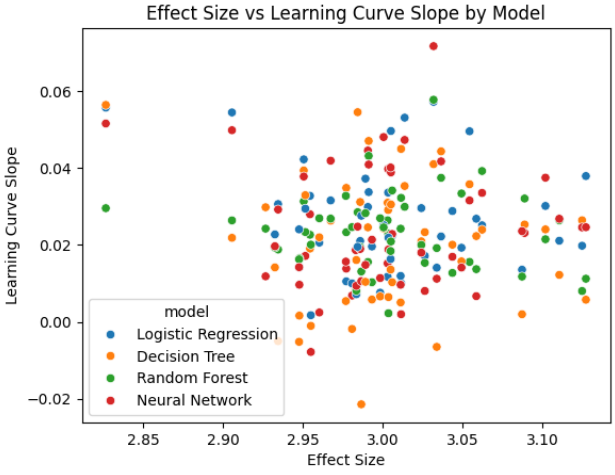



Figure 4.6: Experiment 2.1 R2 Value: 0.0054

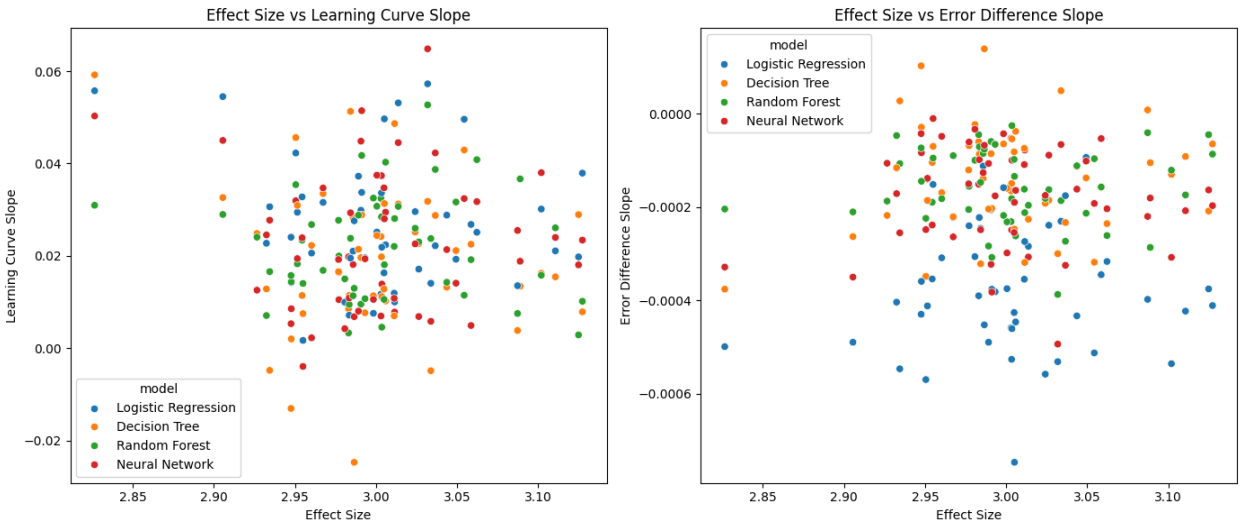



Figure 4.7: Experiment 2.2 R2 Value: 0.0007

Experiment 2: Again there is almost no correlation, but this time between effect size and data sufficiency. Experiment 2.1 suggests there is no correlation between the effect size and slope of the validation learning curve. Experiment 2.2 indicates no correlation between effect size and slope of the magnitude of the difference in error between the training and validation sets. The data points and R2 values suggest that determining a sufficient sample size to train your data is dependent on some other factor of our dataset along with the specific model utilized.

## 4.5 Conclusion

In the ongoing effort to establish a robust framework for AI trustworthiness, a critical component of any eventual solution will be analyses that can be performed by objective and independent researchers without the need for the data the model was trained with initially. One possible approach may be to examine the architecture of the model in question in reference to another established model, in the hopes of gleaning insights into robustness and generalizability. We proposed an approach of taking an aggregated correlation score of neuronal activation across two independent neural networks. Our resulting correlation score effectively reveals connections across diverse neural networks. This score highlights performance implications, suggesting that lower corre-

lations may signal deviations from expected activation patterns, and further indicates that strong correlations with robust models imply similar expectations for performance in other networks. Additionally, our method addresses a critical gap in responsible AI development by providing a robust approach to evaluate model quality and generalizability, independent of internal developer datasets. In spite of these promising results, this approach faces limitations in precision with larger models and cannot pinpoint particular reasons for low correlation scores, making inferences less specific. Future research should focus on resolving the deficiencies, optimizing efficiency, and enhancing diagnostic capabilities to deepen insights into neural network behaviors, ultimately supporting the safe and effective deployment of AI systems.

## 4.6 Subsequent Work: From Neuronal Correlation to Information-Centric Compression

### 4.6.1 Limits of cross-model neuronal correlation.

While we attempted to expand upon this work, ultimately correlation-based assurance presumes that different networks expose comparable activation spaces for the *same* inputs. In practice, this approach breaks down along several axes: (i) **dataset heterogeneity** (proprietary, domain-specific corpora with distinct label spaces and priors), (ii) **feature-map incongruence** (depth, stride, dilation, tokenization, and normalization producing non-isomorphic layers), (iii) **pipeline drift** (preprocessing/augmentation differences that change input distributions), and (iv) **IP/privacy constraints** (inability to share raw data or internal states). As a result, correlating activations across models trained on *different* datasets yields apples-to-oranges scores that neither scale nor retain objective meaning beyond the narrow setting in which they were computed.

**Abstractions we explored (briefly).** We considered representation-level normalization and alignment—e.g., projecting to common subspaces (CCA/CKA-like ideas), probing tasks, and distributional surrogates—to "standardize" activations across systems. These routes proved brittle, costly to

compute at scale, and still dependent on latent assumptions about dataset overlap and feature geometry. The core insight was that any robust, comparable score must be *grounded in the same data distribution*.

### 4.6.2 Information-centric reframing.

We therefore pivoted from cross-model correlation to an *information-centric* view on a fixed dataset: measure how much class-relevant information each computational motif carries and retain pieces that maximize label-aware information flow. Once framed this way, the signal is more valuable as a *constructor* than as a *score*: it tells us what to keep and what to cut and served as a basis for beginning our work on model compression. The resulting work: M2M-DC operationalizes this approach, and is explored fully in the next chapter.

# Chapter 5

# Information-Guided Model Pruning: MI-to-Mid Distilled Compression (M2M-DC)

## 5.1 Introduction

As modern deep neural networks (DNNs) continue to expand in depth, width, and architectural complexity, their deployment cost, in terms of latency, energy, and memory footprint, has become a central bottleneck for real-world use. At the same time, the need to ensure that deployed models are robust and trustworthy has only grown more urgent as AI systems are integrated into high-stakes settings including healthcare, public infrastructure, and security-sensitive applications [23]. Models in these settings are expected not only to perform well in-distribution, but also to generalize reliably, maintain consistent behavior under perturbations, and exhibit predictable failure modes.
Multiple current methods exist by which large neural networks can be pruned substantially while preserving task performance, provided that removals target parameters or blocks with limited task relevance and that the surviving network is properly adapted, however such methods vary widely in compression efficacy, and training cost, and there is a broad ongoing effort to continue to push

the boundaries of what is possible at the frontiers of model pruning.

In hopes of furthering this broader effort, in this work we introduce **MI-to-Mid Distilled Compression (M2M-DC)** a novel, *combinatorial information guided* pruning strategy combined with a short, staged knowledge distillation (KD) schedule. Our approach combines two information-centric mechanisms that operate at different scales. Prior work on model compression via pruning and quantization has established effective efficiency-accuracy tradeoffs [45, 23], while staged knowledge distillation (KD) provides a powerful teacher signal during compression [51]. Our approach composes these ideas with information-theoretic guidance. First, a *block-level* mutual-information (MI) criterion removes residual units that contribute little task signal subject to stage-survival constraints. Second, a *layer-level, planes-aware* channel selection ranks and keeps the top-$K$ residual planes per stage using information-guided scores (conditional MI when available; otherwise BN scale magnitudes and/or residual filter norms), while co-slicing the downsample path and the next stage's input to preserve residual geometry. Interleaving brief KD phases between steps stabilizes targets and recovers accuracy.

We evaluate this pipeline across multiple architectures (ResNet-18/34 and MobileNetV2) and random seeds. Empirically, Info-Guided Block Pruning with Progressive Inner Slicing produces students that *match or surpass* the teacher while reducing FLOPs and parameters; even at aggressive prune ratios, KD substantially closes the gap between pre- and post-pruning accuracy at levels that meet or exceed current SOTA levels. These results suggest that a purely Information-centered criterion, paired with lightweight KD, is a practical, cost-effective, and reproducible basis for compute–accuracy Pareto improvements.

In summary, our contribution is to introduce a two-scale, information-guided compression strategy that pairs mutual-information (MI) block pruning with distilled inner mid-channel slicing to deliver compact, accurate ResNet-class students. By first removing globally uninformative residual blocks and then reshaping capacity within surviving layers—while interleaving short knowledge-distillation phases—we achieve substantial reductions in parameters and FLOPs without sacrificing

(and often improving) accuracy. The approach is "residual-safe" by construction, maintains stable optimization via BN recalibration and KD, and cleanly extends to planes-aware/channel-ranking refinements. Practically, it targets deployment realities: lower latency and smaller memory footprints on modest hardware, with promising calibration characteristics critical for decision support. Conceptually, it unifies information-theoretic selection with staged repair, yielding a cleaner accuracy–compute Pareto frontier than either technique alone and setting a template for sequence/next-action models used in clinical workflows.

## 5.2 Background & Related Work

### 5.2.1 Classical Approaches to Model Compression

**Unstructured pruning and sparsification.** One of the earliest and most influential approaches removes individual weights with small magnitude, yielding sparse weight matrices that can be stored and (in principle) executed more efficiently [46]. After pruning, the model is typically fine-tuned to recover accuracy. This can dramatically reduce parameter count [46]. However, unstructured sparsity often requires custom runtimes or specialized hardware to realize inference speedups in practice.

**Structured channel/filter pruning.** Channel pruning and block pruning remove complete channels, filters, or even whole residual units. These methods typically use importance scores based on weight norms, Taylor expansion of the loss, or gradient-based saliency to rank structures. For example, [72] propose a straightforward and effective heuristic: prune filters with the smallest **L1-norms**. Others, like [50], use a LASSO regression-based channel selection to identify unimportant channels and iteratively prune them. In their "Network Slimming" approach, [74] introduce **L1-regularization on the scaling factors ($\gamma$) of Batch Normalization layers**; channels whose $\gamma$ factors are pushed to zero during training are identified as unimportant and subsequently pruned. This produces slimmer feature maps and shallower effective depth.

**Quantization.** Quantization reduces numerical precision (e.g., 32-bit floating point $\rightarrow$ 8-bit or lower) for weights and/or activations [60]. This can dramatically cut memory bandwidth and improve throughput on modern accelerators. Quantization-aware training or post-training calibration is often required to avoid large accuracy drops, especially at very low bit-widths.

**Low-rank and factorized decompositions.** Another common strategy is to approximate weight tensors with low-rank factorizations, depthwise separable convolutions, or group convolutions [53]. These architectural motifs are often designed *a priori* for efficiency, rather than carved out of an existing large model.

**Knowledge distillation.** Knowledge Distillation (KD) trains a smaller "student" network to match the outputs (and sometimes intermediate representations) of a larger "teacher" [51]. KD is often used after a hand-designed or pruned student architecture is chosen. It can recover surprising amounts of accuracy and stabilize aggressively compressed students.

### 5.2.2 More Recent Pruning Approaches

Recent surveys organize pruning along three orthogonal axes: (i) *what* speedup is realized (unstructured/sparse vs. structured, and universal vs. hardware-specific), (ii) *when* pruning occurs (before, during, or after training), and (iii) *how* saliency is determined (hand-crafted criteria vs. learned sparsity) [23]. In practice, *structured* pruning (channels, filters, blocks) is the most deployment-friendly route to wall-clock gains on commodity hardware, whereas unstructured/semistructured sparsity often requires specialized kernels (e.g., $N{:}M$ patterns). Representative exemplars include PBT methods (SNIP, GraSP) that preserve gradient flow at initialization [67, 118]; PDT methods such as SET/RigL and movement pruning that update sparsity online [83, 34, 102]; and PAT structured pruning via norms or Taylor saliency [72, 85, 50]. Compression is frequently combined with quantization or low-rank factorization [45, 37] and followed by knowledge distillation (KD) [51]. Semistructured $N{:}M$ patterns further trade flexibility for accelerator efficiency [131].

**Information-theoretic pruning: from global relevance to flow preservation.** Information-theoretic criteria offer task-aware signals by estimating the mutual information (MI) between internal features and labels. *High-Relevance (HRel)* ranks *filters* by MI with class labels using a non-parametric estimator, prunes the lowest-relevance filters *layer-wise*, and retrains between rounds; it also analyzes post-pruning behavior on the information plane [1]. Complementing label-centric MI, *Mutual Information Preserving Pruning (MIPP)* reframes the objective around *inter-layer* information flow: it selects upstream nodes so that the MI between upstream and downstream activations is preserved after pruning, guaranteeing the existence of a mapping from pruned upstream features to the downstream layer and improving re-trainability with minimal fine-tuning [121]. A parallel line uses *conditional* MI (CMI) to rank channels by the *unique* information they contribute conditioned on other channels or labels, enabling parallel layer-wise pruning and aggressive compression while limiting accuracy loss [22, 100].

**Explainability-aware pruning.** A contemporary trend ties pruning to attribution: XAI methods (e.g., Integrated Gradients, LRP) provide task-aligned saliency, and recent work shows that *optimizing* attribution hyperparameters specifically for pruning improves compression–accuracy trade-offs across CNNs and Transformers, including few-shot and transfer regimes [103].

### 5.2.3 How Our Approach Differs Conceptually

**Positioning of our approach.** We adopt an information–theoretic, data-driven strategy, but differ in both *granularity* and *what is preserved*. At a coarse scale, we compute a label-aware MI score at the *block* level and *remove entire residual (or inverted-residual) blocks* with the lowest MI (structured PAT), followed by BN recalibration and a staged KD schedule to stabilize and recover accuracy. At a finer scale, we introduce a *residual-safe inner slicing* step that prunes only the *mid* channels inside surviving blocks (conv1 out / BN1 / conv2 in) while *keeping conv2 out (the residual planes) unchanged*, thereby preserving residual geometry by construction. The two scales are complementary: block-level MI targets global representational redundancy, while inner slicing

reallocates capacity within stages at low risk; short KD phases interleaved between steps maintain optimization stability and recover performance.

**Our approach vs. prior MI methods.** Relative to HRel's *filter*-level, *layer-wise* thinning [1], our block criterion acts across layers and maps directly to universal FLOP/latency reductions; the subsequent inner slice then operates *within* those surviving stages, but in a residual-safe manner (conv2 out fixed), avoiding add-mismatch pathologies. Compared to MIPP [121], which preserves *adjacent-layer* MI to guarantee recoverability, we explicitly optimize label-aligned MI at the block level and rely on staged KD for recovery rather than enforcing inter-layer MI constraints during pruning. In contrast to conditional-MI channel selection [22, 100], our coarse unit aggregates per-block redundancy for hardware-friendly speedups, while our inner slice provides a controlled, fine-grained complement; these lines are compatible and could be combined in future work. Finally, XAI-driven pruning [103] offers another task-aligned saliency signal that is orthogonal to our staged KD recovery mechanism and could be integrated as an alternative ranking source.

**Other contemporary directions.** Beyond MI, our method is generally complementary to methods like structured channel/filter and block pruning frequently use BN-$\gamma$ and magnitude/Taylor/-gradient criteria [74, 72, 50, 85]; semi-structured patterns (e.g., 2:4) exploit accelerator sparsity; and learned sparsity (regularization/dynamic sparse training) provides alternative routes. Pruning also composes naturally with quantization and low-rank decompositions, and is increasingly studied for large architectures (ViTs, diffusion models, LLMs).

**Design invariants and training protocol.** (i) *Residual safety*: inner slicing never changes conv2 out, so the residual add remains dimensionally valid; (ii) *Post-slice BN recalibration*: prevents spurious covariate shift from masked channels; (iii) *Staged KD with weight ramp*: briefly distill after each structural change to recover accuracy and smooth optimization; (iv) *Hardware alignment*: block-level removals yield platform-agnostic FLOP/latency gains, while inner slices reduce mid-layer compute with minimal structural disruption.

**Algorithm 2** Two-scale information-guided compression with residual-safe layer splicing

**Input:** Teacher $\mathcal{T}$, target keep budget; data $\mathcal{D}$
**Block saliency:** compute $s_b = \hat{I}(Z_b; Y)$ for each block $b$
**Constrained block pruning:** prune lowest $s_b$ subject to stage/transition safety
**Student init:** copy $\mathcal{T}$, delete pruned blocks $\rightarrow$ student $\mathcal{S}$
BN recalibration on $\mathcal{S}$; staged KD (ramped CE+KL) against $\mathcal{T}$
**for** each stage $l$ with planes $C_l$ **do**
    compute $s^{(l)}$ via conditional MI or BN-$\gamma$ + $\ell_1$ proxy
    select $\mathcal{K}_l = \mathrm{TopK}(s^{(l)}, K_l)$
    **Planes co-slice:** slice `conv2`/`bn2` out, downsample, and next `conv1` in to $\mathcal{K}_l$
    **(Optional) Mid slice:** slice `conv1` out / `bn1` / `conv2` in; keep `conv2` out fixed
    BN recalibration; short KD repair (CE+KL ramp)
**end for**

## 5.3 Methodology

We propose a two–scale, information–guided compression pipeline that first removes globally uninformative computation at the *block* level, then reshapes capacity *within* surviving stages using residual-safe *layer splicing*. After every structural change we perform BatchNorm (BN) recalibration and run a short, staged knowledge–distillation (KD) phase against the unpruned teacher to stabilize optimization and recover accuracy. The procedure is architecture-stable and applies to residual CNNs (e.g., ResNet-18/34 [48]) and mobile-style CNNs (e.g., MobileNetV2 [54]) by abstracting blocks and stages behind a minimal interface. In what follows, we elaborate on each stage, including implementation notes and practical considerations; the complete pipeline is summarized in Algorithm 2.

### 5.3.1 Block-Level MI Pruning (with Constraints) and Repair

Let $b$ index a semantic block (e.g., a residual or inverted-residual unit). For an input minibatch $x$, record the block activation $A_b(x) \in R^{C_b \times H_b \times W_b}$, and obtain a channel descriptor by spatial mean pooling,

$$Z_b(x) = \mathrm{mean}_{h,w}\big(A_b(x)\big) \in R^{C_b}.$$

Score each block with a label-aware mutual information (MI) estimator,

$$s_b = \hat{I}(Z_b; Y) \;=\; \frac{1}{C_b} \sum_{c=1}^{C_b} I\big(\,\mathrm{bin}(Z_{b,c}),\, Y\big),$$

where $\mathrm{bin}(\cdot)$ applies per-channel quantile binning (empirical default: `10` bins). We compute $\hat{I}$ on a fixed probe loader to stabilize rankings. Intuitively, blocks with low $\hat{I}$ contribute little unique information about $Y$ and are prime pruning candidates.

**Constrained pruning.** Given scores $\{s_b\}$ and a target prune ratio $r \in [0, 1)$, let $\mathcal{B}_{\text{free}}$ be the set of prunable blocks (i.e., not protected by architecture rules). We prune

$$k = \Big\lfloor r \cdot \big|\mathcal{B}_{\text{free}}\big| \Big\rfloor$$

blocks subject to two constraints:

**Architectural safety** (never remove resolution/channel-transition units).

- **ResNets:** protect `layer2.0`, `layer3.0`, `layer4.0` (downsampling shortcuts).
- **MobileNetV2:** protect any `InvertedResidual` with stride $\neq 1$ or where `in_channels` $\Rightarrow$ `out_channels`, and protect the non-inverted-residual stem/-tail ops.

**Stage survival** (retain representational depth): each stage (e.g., `layer1`, `layer2`, ...) must keep at least one block.

Algorithmically, we sort blocks by $s_b$ in ascending order and we greedily mark them for pruning while skipping protected blocks and any choice that would empty a stage. The survivors define a

compact keep configuration:

$$\texttt{keep\_cfg} = \{\ \texttt{layer1} : [0],\ \texttt{layer2} : [0],\ \texttt{layer3} : [0, 1],\ \texttt{layer4} : [0, 1]\ \}.$$

We form the student by deleting the pruned blocks (sharing weights elsewhere where applicable), recalibrating BatchNorm, and running staged knowledge distillation (KD) with a linear ramp on the distillation weight. Relative to HRel's filter-level, layer-wise thinning [1], we target a coarser block unit that maps directly to FLOP/latency reductions across diverse backbones; relative to MIPP [121], we optimize a label-aligned mutual information objective and rely on staged KD rather than inter-layer MI constraints.

### 5.3.2 Layer-Level Planes-Aware Pruning (Complementary to Block MI)

Even after block pruning, surviving stages still harbor channel-level redundancy. To harvest it without violating residual constraints, we act on each stage's *planes* (the out-channels of `conv2` that are added to the skip connection).

**Ranking signals.** For stage $l \in \{\texttt{layer2}, \texttt{layer3}, \texttt{layer4}\}$ with $C_l$ planes, we form a per-channel score vector $s^{(l)} \in R^{C_l}$. When available, we prefer conditional MI; otherwise we use strong proxies correlated with information flow, BatchNorm scale magnitudes ($|\gamma|$) and residual filter norms (e.g., $\ell_1$ of `conv2`) aggregated across blocks, building on [1, 74, 72, 50, 85]. We then select the top-$K_l$ planes per stage via

$$\mathcal{K}_l = \mathrm{TopK}\big(s^{(l)},\ K_l\big).$$

**Shape-safe co-slicing (planes).** Given $\mathcal{K}_l$, we *co-slice*: (i) each block's `conv2`/`bn2` *out-channels* to $\mathcal{K}_l$; (ii) the first block's downsample path (if present) to $\mathcal{K}_l$; and (iii) the *next* stage's `conv1` *in-channels* to $\mathcal{K}_l$. This enforces the residual invariant $\texttt{conv2\_out\_C} = \texttt{identity\_C}$ for both transition and non-transition blocks, guaranteeing that $x + \mathrm{F}(x)$ remains dimensionally valid.

**Mid-channel inner slice (residual-safe).** Orthogonally to planes selection, we optionally prune only the *mid* channels within each block—`conv1` out / `bn1` / `conv2` in—while keeping `conv2` *out* fixed to the stage planes. This yields additional FLOP reductions at low risk and composes cleanly with planes co-slicing.

**Staged KD between steps.** After each planes and/or mid-channel slice, we recalibrate BN and run a short KD phase with a CE+KL loss (teacher logits), using a linear ramp of the distillation weight [51]. This reliably recovers performance lost to capacity changes and keeps the student close to the teacher's decision boundary as stages shrink.

**Complexity and safety.** Planes co-slicing reduces parameters and MACs roughly in proportion to $K_l/C_l$ for stage $l$, while preserving residual tensor shapes by construction. Inner mid-slicing further reduces MACs without touching residual planes. Empirically, composing block-level MI with residual-safe layer splicing yields a cleaner accuracy–compute Pareto frontier than either component alone.

**Implementation notes.** Any plug-in MI estimator can drive the per plane scores, and in low data regimes the BatchNorm scale and $\ell_1$ filter norms provide a resilient fallback. After every slice we verify the residual invariants by checking that non transition blocks satisfy `conv2_out_C` = `conv1_in_C` and that transition blocks satisfy `conv2_out_C` = `downsample_out_C`. In our experiments we instantiate the layer step with uniform per-layer-keep fractions for clarity and clean ablations; replacing this policy with per channel ranking by conditional mutual information or its proxies is fully compatible and left as an orthogonal refinement.

### 5.3.3 Student Reconstruction and BatchNorm Recalibration

We materialize the student by copying the teacher and replacing each stage with the subsequence specified by `keep_cfg`. Because all kept modules are weight-identical to the teacher, the student initialization is strong. We then perform BatchNorm (BN) recalibration: a forward-only pass over $\sim$50 batches updates BN running means/variances, which substantially reduces the immediate accuracy drop post-pruning.

### 5.3.4 Staged Knowledge Distillation (KD) Repair

Let $f_s$ and $f_t$ denote student and teacher. For input $x$ with label $y$, we minimize

$$\mathcal{L} = \underbrace{\mathrm{CE}\big(f_s(x), y\big)}_{\text{supervised}} + \alpha \underbrace{\big(1 - \cos\angle(f_s(x),\, f_t(x))\big)}_{\text{logits align}} + \beta \underbrace{\big(1 - \cos\angle(\phi_s(x),\, \phi_t(x))\big)}_{\text{feature align}},$$

where $\phi(\cdot)$ extracts penultimate features (e.g., pooled output of `layer4` in ResNet or the last `features` block in MobileNetV2), and $\cos\angle(u, v) = \frac{\langle u, v\rangle}{\|u\|\,\|v\|}$ is cosine similarity.

We use a simple linear schedule over epochs $t = 1, \ldots, T$ (with $T = 5$):

$$\alpha_t = \alpha^\star \cdot \frac{t-1}{T-1}, \qquad \beta_t = \beta^\star \cdot \frac{t-1}{T-1},$$

with $(\alpha^\star, \beta^\star) = (0.1, 0.1)$. Thus epoch 1 sets $(\alpha, \beta) = (0, 0)$ (pure CE "surgery repair"), while epochs 2–5 gradually introduce logits and feature alignment. We train with Adam (base learning rate $10^{-4}$ for KD; $10^{-3}$ for the teacher fine-tune), gradient clipping at 1.0, and standard image augmentations.

## 5.4 Experimental Results

### 5.4.1 Setup

**Data and preprocessing.** Following common practice, we train and evaluate on CIFAR-100 with images resized to $224 \times 224$, per-channel normalization (ImageNet mean/std), and random horizontal flips for the training set.

**Teacher fine-tune.** For each architecture (ResNet-18, ResNet-34, MobileNetV2) and each random seed, we take ImageNet-pretrained weights, replace the classifier for 10 classes, unfreeze the final third of the backbone (ResNet: `layer3`, `layer4`, and `fc`; MobileNetV2: last $\sim$1/3 of `features` and `classifier`), and train for 5 epochs with Adam at $10^{-3}$.

**MI scoring.** We collect pooled activations on a fixed probe iterator (deterministic order; batch size 64; $\leq$5000 samples) and compute $\hat{I}(Z_b; Y)$ per block via channel-wise quantile binning and mutual information. Blocks are ranked descending by MI for reporting and ascending for pruning.

**Pruning schedule.** We sweep prune ratios $r \in \{0.10, 0.25, 0.40, 0.50\}$ and also test *frontier* hand-crafted `keep_cfg` variants (e.g., making late stages single-block) to probe the collapse boundary. For MobileNetV2, pruning is confined to repeat inverted residuals within a channel group (safe by construction).

**Student evaluation (pre-distill).** After reconstruction and BN recalibration, we compute pre-distillation accuracy, parameters, and FLOPs.

**KD schedule and logging.** We run 5 KD epochs with the staged schedule above, logging per-epoch CE, logits-alignment, feature-alignment losses, and accuracy. This makes visible the characteristic "epoch-1 rescue" (large accuracy jump with CE-only) and the subsequent KD polishing (+1–2% absolute).

Table 5.1: ResNet-18 on CIFAR-100: per-seed Top-1 accuracy (%).

| Seed | Teacher | Block-KD | Inner-slice |
|---|---|---|---|
| 42 | 82.82 | 85.37 | 85.04 |
| 99 | 82.71 | 85.21 | 85.46 |
| 123 | 83.93 | 85.07 | 85.37 |
| Mean $\pm$ Std | $83.15 \pm 0.55$ | $85.22 \pm 0.12$ | $85.29 \pm 0.18$ |

Table 5.2: ResNet-18: compute and parameter profile.

| Model | Acc (mean) | Params (M) | GMacs |
|---|---|---|---|
| Teacher | 83.15 | 11.18 | 0.0372 |
| Block-KD | 85.22 | 9.63 | 0.0230 |
| Inner-slice | 85.29 | 3.09 | 0.0139 |
| Inner-slice reductions vs. Teacher: $-72.4\%$ Params, $-62.6\%$ GMacs. | | | |

**Cross-seed and cross-architecture.** To assess robustness, we repeat the full pipeline for seeds $\{42, 123, 999\}$ on ResNet-18 and ResNet-34. We also run MobileNetV2 under the same regime to demonstrate architectural generality. All runs report mean$\pm$std of accuracy and FLOPs, with per-seed curves available in the supplement.

**Frontier experiments.** We additionally construct minimal `keep_cfgs` that keep only the stage-initial/downsampling blocks (e.g., `layer`$\{1, 2, 3, 4\} : [0]$ in ResNet-18) and push pruning until KD can no longer restore above-teacher accuracy, empirically mapping the collapse frontier and the Pareto knee of accuracy vs. compute.

**Hyperparameters.** Unless noted: Adam optimizer, teacher LR $10^{-3}$, student LR $10^{-4}$ for KD, batch sizes 128 (train) / 256 (test), 5 epochs for teacher fine-tune and 5 epochs for KD, gradient clipping at 1.0. We profile FLOPs on a single $1 \times 3 \times 224 \times 224$ dummy tensor.

**Comparators.** Our focus is on ablating *saliency source* (functional MI vs. conventional magnitude/gradient heuristics) and *repair schedule* (staged KD). Baselines include: (i) no pruning; (ii) pruning by MI without KD (to isolate the contribution of the repair); and (iii) staged KD without MI pruning (to show that KD alone does not create the compute savings).

### 5.4.2 Results on ResNet-18

**Takeaways (R-18).** Inner slicing delivers a stronger Pareto point than both the teacher and the block-KD student: *higher* accuracy at *much lower* compute. Across seeds, the final inner model

Table 5.3: ResNet-34 on CIFAR-100: per-seed Top-1 accuracy (%).

| Seed | Teacher | Block-KD | Inner-slice |
|---|---|---|---|
| 42 | 83.50 | 83.56 | 84.25 |
| 99 | 82.61 | 83.58 | 84.58 |
| 123 | 81.63 | 83.86 | 85.02 |
| Mean $\pm$ Std | $82.58 \pm 0.76$ | $83.67 \pm 0.14$ | $84.62 \pm 0.32$ |

Table 5.4: ResNet-34: compute and parameter profile.

| Model | Acc (mean) | Params (M) | GMacs |
|---|---|---|---|
| Teacher | 82.58 | 21.29 | 0.0751 |
| Block-KD | 83.67 | 16.71 | 0.0372 |
| Inner-slice | 84.62 | 5.46 | 0.0195 |
| Inner-slice reductions vs. Teacher: $-74.4\%$ Params, $-74.0\%$ GMacs. | | | |

averages $85.29\%$ Top-1 with only $3.09$M params/$0.0139$ GMacs ( $-72.4\%$/$-62.6\%$ vs. teacher).

### 5.4.3 Results on ResNet-34

**Takeaways (R-34).** The final inner model improves average accuracy by $+2.04$ points over the teacher while removing $\sim 74\%$ of both parameters and multiply–adds.

### 5.4.4 MobileNetV2 (3 seeds)

Table 5.5: MobileNetV2 on CIFAR-100: per-seed Top-1 accuracy (%).

| Seed | Teacher | Block-KD | Inner-slice |
|---|---|---|---|
| 42 | 64.76 | 65.39 | 66.27 |
| 99 | 65.44 | 68.62 | 69.29 |
| 123 | 67.88 | 69.65 | 70.05 |
| Mean $\pm$ Std | $66.03 \pm 1.34$ | $67.89 \pm 1.81$ | $68.54 \pm 1.63$ |

Table 5.6: MobileNetV2: compute and parameter profile (means across seeds).

| Model | Acc (mean) | Params (M) | ConvGMACs |
|---|---|---|---|
| Teacher | 66.03 | 2.35 | 0.0244 |
| Block-KD | 67.89 | 1.71 | 0.0186 |
| Inner-slice | 68.54 | 1.71 | 0.0186 |
| Inner-slice reductions vs. Teacher: $-27.4\%$ Params, $-23.8\%$ ConvGMACs. | | | |

**Takeaways (MobileNetV2).** M2M-DC lifts Top-1 to **68.54%** (+2.51 over the teacher) while trimming compute to $\sim$1.71M params / $\sim$0.0186 ConvGMACs ($-27.4\%$ / $-23.8\%$). Most gains come from MI-guided block pruning with KD; the current inner-slice keeps that budget and adds a small, consistent bump.

## 5.5 Conclusion

We introduced *M2M-DC*, a residual-safe compression recipe that couples label-aware, block-level MI pruning with planes-aware inner slicing and brief staged KD after each edit. The method is simple to implement, hardware-friendly, and transfers across residual and inverted-residual families. On CIFAR-100, it delivers strong accuracy–compute trade-offs: for ResNet-18, **85.29%** mean Top-1 at **3.09M** params/**0.0139** GMACs; for ResNet-34, **84.62%** mean at **5.46M/0.0195**; and for MobileNetV2, **68.54%** mean at $\sim$**1.71M/**$\sim$**0.0186** ConvGMACs,+2.51 points vs. its 66.03% teacher while cutting parameters/compute by 27%. KD is key to recovering (and sometimes surpassing) teacher accuracy after aggressive structure edits. Limitations include MI-estimator variance at small sample sizes, reliance on proxy ranks for inner slicing in low-data regimes, and current focus on classification. Future work will tighten estimators (conditional MI/calibration), add per-layer frontier search, and report on ImageNet-scale accuracy, on-device latency/energy, and transfer to detection, segmentation, and control-loop policies.

# Chapter 6

# Foundation Model for Clinical Diagnostic Pathways

## 6.1 Introduction

Healthcare is, at its core, a human enterprise—dynamic, contingent, and shaped by judgment under uncertainty. Yet over the last two decades the practice of care has been progressively codified into standardized pathways: order sets, triage protocols, clinical guidelines, templated documentation, and decision-support nudges embedded in the EHR. This shift hasn't diminished clinician agency; it has reframed it. Clinicians still improvise and prioritize, but they do so within workflows that are increasingly observable, structured, and machine-readable. That interplay creates new opportunities for computational models capable of better predicting it.

Traditional clinical AI has mostly aimed at disease: predicting diagnoses, risks, or outcomes from patient data. Valuable as those models are, they treat care delivery as the backdrop rather than the subject. Meanwhile, the delivery of care has itself become an increasingly standardized and algorithmic process—composed of discrete, timestamped actions such as ordering tests, starting therapies, consulting services, documenting assessments, and planning disposition. Concurrently the increased digitalization of clinical events in structuered EHR records, provides a unique op-

portunity to reframe the question around clincal care. When we reframe the problem from "What disease does this patient have?" to *"What is the next clinically appropriate action in this context?"* we align modeling targets with the operational reality clinicians inhabit, and the data to support it.

Large Language Models (LLMs) make this reframing tractable. Because they naturally model sequences and context, LLMs can be trained to predict **next-in-sequence actions**-not just words—over event streams that encode clinical workflow (orders, notes, vitals, results, handoffs). In this view, a patient encounter becomes a tokenized trajectory; prompts condition on patient state and recent actions; outputs propose likely next steps with rationales. The same mechanism that predicts the next token in a sentence can anticipate the next order, assessment, or disposition—adapting as new information arrives and as clinician behavior diverges from canonical pathways.

Modeling of human actions opens a new lens on both behavior and outcomes. It enables decision support that is situational (responsive to what the clinician just did), anticipatory (surfacing downstream needs before delays occur), and evaluable in counterfactual terms (what if we had ordered imaging before antibiotics?). It also allows learning from natural variation across clinicians and sites: rather than enforcing a single pathway, the model can surface multiple plausible next steps, calibrated by context and uncertainty, while tracking which action sequences correlate with safer, faster, or more equitable outcomes.

Framed this way, LLMs become not prescriptive authorities but adaptive collaborators—absorbing the structure of modern care while leaving room for the craft of medicine, and, in doing so, helping us model not only *what clinicians know* but *what clinicians do* and how those choices shape patient outcomes.

## 6.2 PRISM Baseline Model

### 6.2.1 Background

#### Limits of Probabilistic Diagnostic Testing

While Bayesian methods provide a mathematically rigorous framework for updating disease probabilities based on diagnostic tests, they struggle with higher order correlations between tests and the specific diagnoses often rely on pairwise likelihoods—probabilities that relate specific tests individually to specific diagnoses. Although individually these probabilities can be empirically accurate and useful, real-world diagnostic situations frequently involve multiple overlapping comorbidities, complex interactions between test outcomes, and numerous patient-specific confounding variables and specific histories.

Early computerized diagnostic aids such as QMR-DT relied on expert rules and Bayesian reasoning, but they covered fewer than 600 diseases and required years of manual curation.[59]. Yet even painstakingly curated models like the QMR-DT, their performance degrades when the model is deployed in a new hospital whose disease prevalence and practice patterns differ from the original cohort.[127] Moreover, classical Bayesian networks are brittle at run time: they do not adapt gracefully when previously unseen evidence types or updated laboratory ranges are introduced.

More fundamentally, when multiple factors coexist, the assumption of conditional independence—often a critical prerequisite for practical Bayesian calculations—breaks down. The intricate interdependencies among symptoms, test results, and diseases in an individual patient's unique clinical context make it computationally infeasible, and often conceptually inaccurate, to comprehensively enumerate and correctly weigh all conditional probabilities. Consequently, Bayesian approaches become limited in their ability to faithfully represent realistic uncertainty, since accurately calculating a truly reflective posterior probability distribution across all potential combinations of conditions and outcomes quickly becomes impossible.

Therefore, relying solely on Bayesian statistical modeling to drive diagnostic decision-making risks oversimplifying patient-level complexity.

### Elaborating the Analogy Between Diagnostic Systems and LLM Token Prediction

In language models (LLMs), each token—representing a word or sub-word unit—has clear pairwise probabilities relative to other tokens. For example, given a token such as ”New,” the subsequent token might have high probability for ”York,” ”Zealand,” or ”Jersey.” Individually, these pairwise probabilities are straightforward. However, as the sequence of tokens expands—considering an entire preceding sentence or paragraph—the combinatorial complexity and contextual dependencies make it impossible to precisely calculate probabilities using first order correlations alone. Hence, deep learning and attention mechanisms in transformer architectures (like GPT) approximate these complex conditional dependencies, capturing long-range contexts and intricate relationships between tokens.

By analogy, diagnostic medicine faces a similar complexity. Individual diagnostic tests may exhibit well-defined pairwise probabilities for specific diseases—just as pairwise token probabilities can be well-established. However, when attempting to predict the next best test or disease diagnosis given the entire preceding sequence (the patient's complete history, including medical tests, symptoms, prior diagnoses, lifestyle factors, and comorbidities), the scenario becomes profoundly more complex. The interactions between these elements are intricate, context-dependent, and frequently nonlinear, making exact probabilistic inference practically infeasible.

Thus, Bayesian statistical approaches quickly reach limitations. Transformer-based diagnostic models, analogous to language models, can approximate the complex contextual dependencies inherent in patient histories, making them potentially more capable of accurately predicting subsequent medical events (tests, results, or diagnoses). These models implicitly learn from data to reflect patient-level complexity more naturally, offering both predictive capability and interpretable attention scores to highlight influential factors in diagnostic reasoning.

### Prior Work in Transformer and Deep Learning Models for Sequential Clinical Diagnosis

In contrast to high-level summations of pre-existing efforts, which are considerable and highly varied in nature, I focus in on prior work of immediate relevance to our proposed approach, by

zeroing in on a methodology-specific approaches that have informed our own experimental design, and serve as a basis for which our work attempts to build on.

**Tokenization and Representation of Medical Data** The first step in adapting transformer models to diagnostic-sequence prediction is to convert a patient's richly structured history into a sequence of discrete *tokens*. Early work relied on standardised vocabularies such as ICD-9/ICD-10, CPT and Read codes, treating each code as an individual token [6, 75]. A coarser alternative aggregates thousands of raw codes into a few hundred clinically coherent groups, reducing sequence length while retaining clinical meaning [6].

More granular pipelines tokenise free-text concepts, symptoms and history fragments extracted from clinical narratives [5]. The ETHOS framework extend this idea further by encompassing: admissions, diagnoses (ICD-10-CM), procedures (ICD-10-PCS), medications (ATC), laboratory results (LOINC) and even inter-event time gaps. Under the ETHOS framework, these events are mapped to 1–7 tokens each, then ordered chronologically into a patient-health timeline (PHT) [7].

Because multiple events can occur simultaneously during a visit, several authors represent a patient record as a *sequence of sets*, forcing the model to be order-invariant within each set. DPSS is the canonical example of this strategy [128]. Finally, semantic enrichment with external knowledge graphs (or hierarchical ontologies such as CCS) can be injected by concatenating concept embeddings with ontology embeddings, as done in the SETOR framework [4].

**Transformer Model Architectures in Diagnostic Prediction** Traditionally model choice varied depending on the prediction task. Encoder-only families (BERT, ALBERT, MedAlbert) excel at classification problems such as early disease detection from longitudinal code streams; MedAlbert, for instance, uses a 6-layer ALBERT encoder to flag incipient lung cancer three years in advance [6]. Decoder-only designs (GPT-like) suit generative forecasting: ETHOS, for instance, employs a causal decoder to roll out future PHTs in a zero-shot setting [7]. Hybrid encoder–decoder stacks (e.g., TransformEHR) translate past encounters into predicted future ICD sequences [75].

Domain-aware modifications are common. SETOR feeds visit-level embeddings— augmented

with CCS-derived ontology vectors and continuous-time positional encodings—into a multi-layer transformer encoder dubbed the *Patient Journey Transformer* [4]. Such customisations can improve both performance and interpretability without abandoning the core attention machinery of the original architecture.

**Datasets and Evaluation Strategies** Most studies train on large, publicly available EHR corpora—chiefly MIMIC-III/IV, which contain in excess of 200k ICU and ward admissions with time-stamped diagnoses, procedures, labs and medications. Task-specific cohorts are also common: WSIC North-West London primary-care records underpin MedAlbert's lung cancer work [6], while the DDXPlus collection provides 49-disease differential labels for multi-label diagnosis models [5]. Limited access to longitudinal, high-quality datasets remains a bottleneck [4].

Evaluation metrics mirror task formulations. For single- or multi-label classification, researchers report accuracy, precision, recall, $F_1$ and AUROC; rank-based measures such as accuracy@$k$ gauge whether the true diagnosis lies in the top-$k$ suggestions [4]. Generative timelines are judged by downstream outcomes (e.g. mortality AUC, LOS MAE) or exact-match token accuracy [7]. Standard practice splits data into train/validation/test partitions, sometimes augmented with cross-validation or zero-shot evaluations on unseen institutions to probe generalisability [128].

## Methods

For this framework, we employed data from the publicly available MIMIC-IV database, an extensive electronic health records repository collected from patients admitted to the Beth Israel Deaconess Medical Center (BIDMC) between 2008 and 2019. The database encompasses detailed clinical information, including demographics, vital signs, laboratory test results, diagnostic procedures, and discharge diagnoses captured during hospital stays [63].

**Patient Selection and Diagnostic Trajectory Filtering**

To investigate diagnostic trajectories, we chose a specific clinical use case to train the model on. Specifically, we opted for patients with initial presentations of undiagnosed chest pain followed by confirmed cardiac conditions. To obtain this subset, we constructed a filtering pipeline using structured data from the MIMIC-IV database. We selected patients whose first hospital encounter included a diagnosis of unspecified or non-specific chest pain, and who were subsequently diagnosed with a more definitive cardiac condition during their clinical course.

Patients were first filtered based on their earliest recorded ICD-9-CM diagnosis, specifically for unspecified or atypical chest pain. The following 5-digit ICD-9 codes were used to define initial inclusion:

- **Initial Admission for Unspecified Chest Pain**
  - `786.50` – Chest pain, unspecified
  - `786.51` – Precordial pain
  - `786.52` – Painful respiration
  - `786.59` – Other chest pain

Following this, patients were retained in the analysis if they subsequently received any diagnosis corresponding to a defined cardiac condition. The cardiac categories and their associated ICD-9-CM codes are listed below in Table 1.

A total of 14,536 unique patients were hospitalized with one of these initial diagnoses, and 3,164 patients were subsequently diagnosed with one of these 26 terminal diagnoses. Temporal ordering of events was verified using timestamped diagnosis records to ensure that cardiac diagnoses occurred after the initial chest pain admission. This filtered cohort was used for downstream analysis involving diagnostic test patterns, time-to-diagnosis evaluation, and treatment initiation windows.

| Category | ICD Code | Description |
|---|---|---|
| Myocardial Infarction (MI) | 410.00 | AMI of anterolateral wall, episode of care unspecified |
| | 410.10 | AMI of other anterior wall |
| | 410.20 | AMI of inferolateral wall |
| | 410.40 | AMI of other inferior wall |
| | 410.70 | Subendocardial infarction |
| Angina Pectoris | 413.00 | Angina, unspecified |
| | 413.10 | Prinzmetal angina |
| | 413.90 | Other and unspecified angina |
| Heart Failure | 428.00 | Congestive heart failure, unspecified |
| | 428.10 | Left heart failure |
| | 428.20 | Systolic heart failure, unspecified |
| | 428.30 | Diastolic heart failure, unspecified |
| | 428.40 | Combined systolic and diastolic heart failure |
| Cardiac Arrhythmias | 427.31 | Atrial fibrillation |
| | 427.32 | Atrial flutter |
| | 427.60 | Premature beats |
| | 427.81 | Sinoatrial node dysfunction |
| | 427.89 | Other specified cardiac dysrhythmias |
| Aortic and Pericardial Conditions | 441.01 | Dissection of aorta, thoracic |
| | 420.9 | Acute pericarditis, unspecified |
| | 423.3 | Cardiac tamponade (pericardial effusion) |
| | 429.0 | Myocarditis, unspecified |
| Pulmonary Embolism | 415.19 | Other pulmonary embolism and infarction |
| Endocarditis | 421.0 | Acute and subacute bacterial endocarditis |
| Ventricular Arrhythmias | 427.1 | Paroxysmal ventricular tachycardia |
| | 427.41 | Ventricular fibrillation |

Table 6.1: Cardiovascular Diagnoses and Corresponding ICD-9 Codes

**Batch Tokenization of Patient Clinical Timelines**

To prepare structured clinical data for transformer-based modeling, we developed a batch tokenization pipeline that converts patient event histories into sequential token representations. This process standardizes heterogeneous clinical events into discrete, interpretable tokens suitable for autoregressive learning frameworks.

We extracted the following six event classes corresponding to tables in the MIMIC-IV database:

1. Demographics

2. Admissions/Discharges

3. Laboratory results

4. Outpatient measurements

5. Microbiology cultures

6. ICD-coded diagnoses

Each table was filtered for the target subject-ID, relabeled with clinically meaningful descriptors (e.g., itemid $\rightarrow$ test name), and assigned an event tag indicating its provenance. Events were then vertically concatenated and sorted first by time and second by a fixed event precedence (admission $\rightarrow$ OMR $\rightarrow$ lab $\rightarrow$ microbiology $\rightarrow$ diagnosis $\rightarrow$ discharge), and subsequently, alphabetically within each event category to create a deterministic, chronological timeline.

**Input Data.** The pipeline accepts individual patient histories stored as CSV files, where each row represents a timestamped clinical event. Events span multiple categories, including demographic information, laboratory tests, diagnoses, medical observations, and microbiology reports. Each event type is associated with specific fields (e.g., lab test priority, diagnosis codes, observation results).

For this initial project, no therapeutic, pharmacological, or other clinical interventions were included, as the focus was narrowly on the process of disease identification and ultimate diagnosis.

**Tokenization Strategy.** For each event, a corresponding token is generated according to predefined templates:

- **Patient Information:** `[INFO]_{TEST}_{VALUE}`
- **Laboratory Orders:** `[ACTION]_ORDER_{TEST}_{PRIORITY}`
- **Laboratory Results:** `[OBS]_RESULT_{TEST}_{MAGNITUDE}_{FLAG}`

- **Diagnoses:** `[DIAG]_{ICD\_CODE}_{LONG\_TITLE}`
- **Outpatient Medical Observations (OMR):** `[OBS]_OMR_{TEST}_{RESULT\_VALUE}`
- **Microbiology Reports:** `[OBS]_MICRO_{SPEC\_TYPE}_{ORG\_NAME}_{INTERPRETATION}_{`
- **Unknown Events:** `[UNK]_{EVENT\_TYPE}`

All text fields are converted to uppercase, with spaces replaced by underscores for consistency. Missing or null values are imputed using placeholder tokens such as `UNKNOWN`, `NONE`, or `NO\_COMMENTS`, depending on context.

**Batch Processing.** The pipeline recursively processes all CSV files within a specified input directory. For each patient file, a corresponding tokenized document is generated, where tokens are space-delimited and saved with a `.txt` extension in a designated output directory.

**Output.** Each tokenized file captures the full chronological sequence of clinical events for a patient in token form, enabling downstream tasks such as next-action prediction, diagnostic trajectory modeling, and clinician behavior simulation.

Notably, events with simultaneous timestamps, for instance a battery of tests ordered, or multiple concurrent diagnoses being input into and EHR, likely as a result of a batched input, were assigned an arbitrary ordering that results in the appearance of a sequential ordinality not necessarily reflective in the actual diagnostic progression. This limitation will be noted and explored further in the discussion section.

### Vocabulary Construction for Clinical Token Sequences

To facilitate transformer-based modeling of structured clinical data, we implemented a vocabulary construction pipeline that maps discrete clinical tokens to unique integer identifiers. This process ensures standardized input representation for autoregressive language models while controlling vocabulary size to optimize computational efficiency.

**Token Collection.** The vocabulary builder processes all tokenized patient documents within a specified directory. Each document contains space-delimited tokens representing chronological clinical events, including diagnostic tests, observations, diagnoses, and treatments.

**Frequency-Based Pruning.** To address the long-tail distribution inherent in clinical event data, we applied a frequency-based pruning strategy. Token frequencies were computed across the entire corpus, and the vocabulary was limited to the $N$ most frequent tokens, where $N$ denotes the predefined maximum vocabulary size (10,000 tokens in this case). This approach retains common clinical patterns while mapping infrequent or rare tokens to a universal unknown token.

**Special Tokens.** Two reserved tokens were incorporated:

- `[PAD]` : Represents padding for sequence alignment in batch processing.
- `[UNK]` : Captures all tokens excluded due to frequency pruning or unseen during inference.

These tokens were assigned fixed indices (0 for `[PAD]` and 1 for `[UNK]`).

**Vocabulary Output.** The finalized vocabulary was serialized in JSON format, providing a mapping from token strings to integer indices. This dictionary serves as the reference for encoding tokenized clinical sequences into numerical tensors compatible with transformer architectures.

### Model Architecture

We employed a decoder-only transformer architecture inspired by GPT-2 to model sequential clinical events in an autoregressive framework. The model was configured to process tokenized representations of patient timelines, where each token corresponds to a discrete clinical action, observation, or diagnosis.

The transformer configuration included:

- **Vocabulary Size:** 10,000 tokens (including special tokens `[PAD]` and `[UNK]`)

- **Maximum Sequence Length:** 512 tokens
- **Embedding Dimension:** 256
- **Number of Layers:** 6 transformer decoder blocks
- **Number of Attention Heads:** 8
- **Positional Encoding:** Learned embeddings

The model was instantiated using the HuggingFace Transformers library, with random weight initialization to accommodate the custom clinical vocabulary.

### Training Procedure

The model was trained using a causal language modeling objective, where the task is to predict the next token in a sequence given all preceding tokens. Cross-entropy loss was computed across all token positions, ignoring padded tokens.

- **Optimizer:** AdamW with a learning rate of $5 \times 10^{-4}$
- **Batch Size:** 8 sequences
- **Number of Epochs:** 5
- **Loss Function:** CrossEntropyLoss (applied to next-token prediction)
- **Hardware:** Training conducted on a Python 3 Google Compute Engine backend (GPU) with 83.5 GB System RAM, 40 GB GPU RAM and 235.6 GB Disk space.

The dataset of tokenized patient timelines was split into training, validation, and test sets using an 80/10/10 ratio. Padding or truncation was applied to standardize sequence lengths to 512 tokens.

#### Validation and Checkpointing

After each epoch, validation loss was computed across the held-out validation set. The model parameters were saved whenever a new lowest validation loss was achieved. This early stopping strategy ensured retention of the best-performing model without overfitting.

### 6.2.2 Results

#### Training dynamics

Figure 6.1 shows the cross-entropy trajectories for the decoder-only transformer over five epochs. Loss on both the training and validation splits fell sharply during the first two epochs and then converged to a shallow plateau, indicating rapid optimisation followed by stable generalisation. The minimal gap ($< 0.03$) between training and validation loss after epoch 3 suggests that regularisation noise (e.g. dropout and weight decay) rather than over-fitting accounts for the remaining discrepancy.

#### Final model performance

Table 6.2 summarises the quantitative metrics. The best checkpoint, obtained at epoch 5, achieved a validation cross-entropy of **0.7000**, corresponding to a perplexity of **2.01**. In other words, conditioned on the patient's history, the model narrows the 10 000-token clinical vocabulary to roughly two plausible next events on average—removing more than 92 % of the intrinsic sequence entropy.

Table 6.2: Training and validation metrics per epoch. Perplexity (PPL) is computed as $e^{\text{loss}}$.

| Epoch | Training | | Validation | | Best ckpt? |
|---|---|---|---|---|---|
| | Loss | PPL | Loss | PPL | |
| 1 | 3.7257 | 41.5 | 1.5462 | 4.70 | – |
| 2 | 1.0811 | 2.95 | 0.8532 | 2.35 | – |
| 3 | 0.7881 | 2.20 | 0.7517 | 2.12 | – |
| 4 | 0.7096 | 2.03 | 0.7163 | 2.05 | – |
| 5 | 0.6666 | 1.95 | **0.7000** | **2.01** | X |

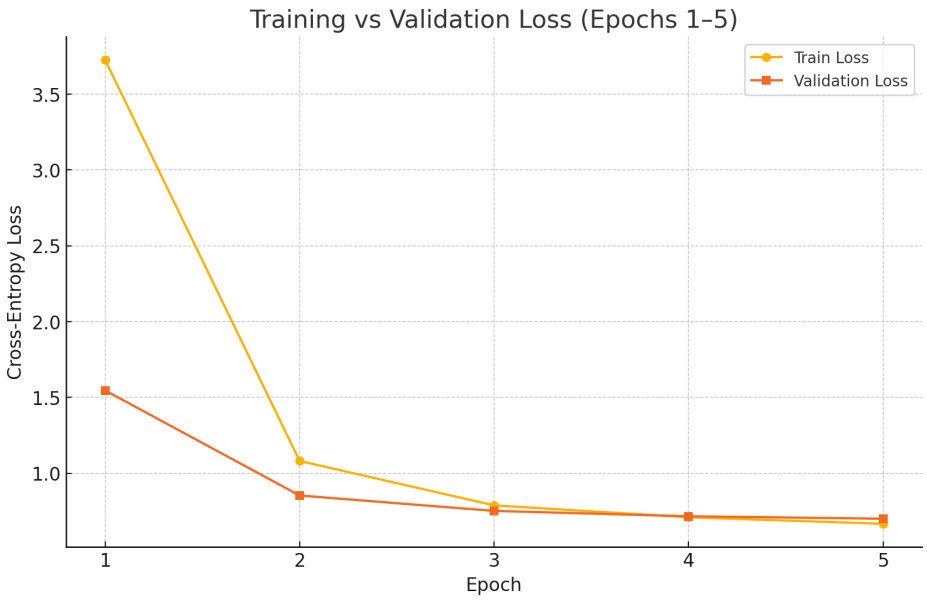



Figure 6.1: Cross-entropy loss on training and validation data over five epochs.

### Interpretation

A validation perplexity of $\sim 2.0$ means the model eliminates $\approx 11.3$ bits of uncertainty per token compared with a uniform 10 000-way guess ($\log_2 10\,000 = 13.3$), retaining only $\sim 1$ bit. Such predictive confidence is sufficient for practical decision-support use-cases (e.g. top-$k$ next-event suggestions) and provides dense clinical embeddings for down-stream risk-stratification tasks.

While no explicit baseline models (e.g., n-gram predictors or recurrent neural networks) were implemented in this study, it is anticipated that such approaches would struggle to capture the long-range dependencies and complex branching logic present in clinical timelines. The transformer's ability to handle extended context windows and model nuanced sequential relationships underscores its suitability for this domain.

These quantitative findings provide a assertive validation of the proposed tokenization strategy and model architecture, demonstrating that large-vocabulary, structured medical data can be effectively modeled using language modeling techniques. Beyond predictive performance, this foundation enables practical applications in clinical decision support, simulation of diagnostic processes, and identification of deviations from standard care pathways.

### Next-Token Generation Experiments

To evaluate the practical applicability of the trained transformer model, we conducted a series of autoregressive next-token generation experiments. The goal was to simulate clinician behavior by predicting subsequent diagnostic actions, test outcomes, or diagnoses based on partial patient timelines.

Three distinct clinical scenarios were designed to assess the model's ability to:

1. Predict likely diagnostic conclusions.
2. Anticipate laboratory test outcomes.
3. Suggest subsequent diagnostic tests in ongoing evaluations.

For each scenario, an initial sequence of tokens reflecting patient demographics and early clinical actions was provided as a prompt. The model then generated a continuation of up to 15 tokens.

### Interpretation of Generated Sequences

The generated sequences demonstrated clinically coherent patterns aligned with standard diagnostic workflows:

- In the **diagnosis prediction** scenario, the model appropriately simulated serial cardiac biomarker testing, reflecting common practice in evaluating suspected acute coronary syndrome.
- For **test outcome prediction**, the model correctly anticipated normal electrolyte and renal function results in a stable hypertensive patient.
- In the **next diagnostic test** scenario, the model followed a logical nephrology workup sequence after detecting elevated creatinine, including eGFR, BUN, and potassium testing.

These results highlight the model's capacity to internalize complex clinical reasoning patterns, suggesting potential utility in decision-support applications. Minor artifacts, such as repetitive test ordering and occasional inconsistencies in lab outcomes, were observed and are discussed in detail in the Discussion section.

Table 6.3: Examples of Next-Token Generation Across Clinical Scenarios

| **Clinical Scenario** | **Initial Clinical Context (Natural Language Prompt)** | **Generated Clinical Sequence (Natural Language conversion of Tokenized output)** |
|---|---|---|
| **Diagnosis Prediction** 75 y/o male with chest pain | A 75-year-old male presents with chest pain. Initial orders include an EKG and Troponin T test. | The model predicted serial cardiac biomarker testing (Troponin T, CK-MB isoenzyme, and total CK), all returning normal results, reflecting standard evaluation for suspected acute coronary syndrome. |
| **Test Outcome Prediction** 55 y/o female hypertension follow-up | A 55-year-old female with essential hypertension undergoes routine electrolyte and renal function testing. | The model anticipated normal potassium, chloride, creatinine, eGFR, and urea nitrogen results, accurately reflecting typical outcomes in stable chronic disease management. |
| **Next Test Prediction** Male with elevated creatinine | A male patient presents with elevated creatinine on STAT testing, suggesting possible renal dysfunction. | The model generated a logical nephrology workup: ordering eGFR, identifying elevated urea nitrogen, confirming persistent creatinine elevation, and proceeding to potassium testing to assess for complications. |

## Discussion

The results of this study demonstrate that a transformer-based language model can effectively learn and predict sequential patterns within structured clinical event data. The PRISM model exhibited strong predictive performance not only for diagnostic conclusions but also for the sequencing of diagnostic tests and expected test outcomes. These findings support the viability of modeling clinical reasoning processes as an autoregressive token prediction task, uncovering latent structure in diagnostic workflows.

A notable strength of this approach lies in its ability to anticipate plausible clinical actions, even when faced with incomplete information. The model's successful simulation of common diagnostic strategies, such as serial cardiac biomarker testing or stepwise nephrology workups, underscores its capacity to internalize complex clinical reasoning patterns. This has immediate implications for clinical decision support, where predictive modeling can guide test ordering, sug-

gest next diagnostic steps, or identify deviations from expected care pathways.

### Limitations

**1. Narrow Patient Cohort Scope** To enhance signal fidelity and reduce the complexity of modeling heterogeneous patient trajectories, we deliberately constrained our training cohort. While over 14,000 patients initially presented with unspecified chest pain, only 3,164 with confirmed cardiac conditions were retained for modeling. This targeted subset dramatically narrowed the range of diagnostic trajectories, yielding cleaner data for learning but limiting the model's generalizability to broader or more ambiguous clinical populations. As a result, PRISM will likely not yet generalize well to patients with non-cardiac pathologies or atypical presentations.

**2. Exclusion of Procedures and Medications** For both conceptual clarity and computational tractability, we excluded therapeutic interventions such as procedures and medication administrations from the tokenized input sequences. This decision was motivated by the need to constrain the vocabulary to a manageable 10,000-token cap, minimizing the number of out-of-vocabulary events and improving model convergence. However, this exclusion restricts the model's diagnostic reasoning capabilities—especially in clinical contexts where treatment response, procedural complications, or drug interactions significantly influence diagnostic trajectories. Consequently, PRISM may overlook important causal pathways or fail to contextualize diagnostic events relative to ongoing therapies.

**3. Linear Modeling of Co-Occurring Events** The current model architecture assumes a strictly sequential ordering of all clinical events, which does not always reflect real-world practice. In many cases, clinicians order multiple tests simultaneously or enter multiple diagnoses in a batch. To accommodate this, events sharing a timestamp were assigned a fixed deterministic order based on event type and alphabetical sorting. While this approach enabled compatibility with autoregressive modeling, it introduces artificial ordering that may distort the temporal relationships between co-occurring events. This is particularly problematic in fast-paced or protocol-driven scenarios

where decision-making occurs in parallel. The inability to represent sets of simultaneous or unordered events limits the fidelity with which PRISM can model genuine clinical workflows.

Future iterations of this work may address these limitations by broadening cohort inclusion criteria to encompass more diverse patient populations, integrating therapeutics into the event vocabulary using hierarchical compression or clinically abstracted embeddings, and adopting hybrid sequence-set or graph-based representations to better capture the multidimensional nature of clinical care. These modifications will further enhance PRISM's ability to emulate clinician reasoning and support real-time diagnostic assistance in complex medical environments.

These findings have several implications. First, they highlight the potential of language models to approximate aspects of clinician behavior, offering avenues for decision-support systems that can anticipate subsequent diagnostic steps or flag deviations from standard workflows. Second, the ability to predict likely test results prior to their execution could inform opportunistic diagnostic strategies, optimizing resource utilization and reducing unnecessary testing.

## 6.3 PRISM-Consult

### 6.3.1 Background

While PRISM had demonstrated considerable promise, it had only been deployed in a limited scope, initially applied only to chestpain presentations in the ED. Under these constrained conditions, PRISM demonstrated strong calibration and efficient inference, validating the premise that carefully scoped tokenization and model size could yield clinically useful predictions without the cost or opacity of very large general-purpose models. However, the same design choices that improve efficiency, namely a streamlined vocabulary and a single-target clinical domain, also *narrow* the applicability of the model: real-world emergency department presentations often traverse multiple organ systems, and differential diagnoses for a single symptom (e.g., chest pain) span cardiac, pulmonary, gastro-oesophageal, musculoskeletal, and psychogenic causes. This motivated an extension from a single-domain backbone to a *routed panel of domain specialists*.

This led us to develop **PRISM-Consult**, a routed *panel-of-experts* architecture that extended the PRISM methodology from a single-domain model to a family of specialist models orchestrated by a lightweight router. The router interprets the earliest events (symptoms and first diagnostic cues) and dispatches the episode to one or more specialist models aligned with clinical organ systems. Each specialist inherits the PRISM tokenization template, shares embeddings for consistency, and is fine-tuned on domain-filtered cohorts labeled by definitive ICD-9 families. This preserves PRISM's efficiency and interpretability while expanding clinical coverage.

This research extended our previous work in three notable ways. First, we formalize a *clinical routing surface* based on initial ICD-9 symptom codes and map it to specialist targets defined by conclusive diagnostic families. Second, we implement a minimalist router and parameter-efficient specialist adapters that retain PRISM's compact footprint while providing cross-domain accuracy. Third, we report partial results across cardiac, pulmonary, gastro-oesophageal, musculoskeletal, and psychogenic domains, showing smooth convergence and low development perplexities (near $\sim$2.0 for all five completed specialists), thereby validating that the PRISM backbone generalizes when routed to domain-consistent corpora.

**Why a Panel of Experts? Tokenization Trade-offs and Clinical Coverage**

Its tokenization strategy sits at the core of PRISM's efficiency. A *more specific* token inventory (e.g., separate tokens for finely binned labs, highly granular diagnoses) captures nuanced patterns and can improve in-domain accuracy and calibration. Yet specificity increases vocabulary size, fragmenting data across rare tokens and slowing inference. Conversely, a *more generic* inventory (broader bins, fewer code variants) compacts the vocabulary and speeds inference but risks losing discriminative power across heterogeneous diseases.

This creates a fundamental tension in single-model designs: optimizing the token schema for one domain (e.g., cardiology) may degrade performance in another (e.g., pulmonary embolism vs. pneumothorax), where different signals and thresholds matter.

**Heterogeneity, Comorbidity, and Cross-Specialty Trajectories in the ED**

Further complicating diagnostic predictions is the fact that emergency presentations are intrinsically heterogeneous and frequently comorbid: a single complaint (e.g., chest pain, dyspnea, syncope) often admits plausible explanations across multiple organ systems, and early measurements (vitals, first labs, first orders) carry overlapping, sometimes contradictory signals. In MIMIC-IV ED cohorts, the same prefix of events can legitimately evolve toward cardiac (AMI, dissection), pulmonary (PE, pneumothorax, pneumonia), gastro–oesophageal (Boerhaave, spasm, reflux), musculoskeletal, or psychogenic end points, or some combination thereof, and patients routinely carry chronic conditions (CKD, COPD, diabetes, atrial fibrillation) that shift priors and confound intermediate tests. These realities create diagnostic pathways that traverse traditional service boundaries and unfold via cross-specialty handoffs.

This cross-disciplinary drift introduces several modeling hazards for single-domain specialists. First, *boundary undercoverage*: a specialist tuned to one domain's vocabulary and thresholds may systematically down-weight rare but life-threatening out-of-domain hypotheses that share early features (e.g., PE vs. AMI in troponin-positive dyspnea). Second, *calibration shift*: domain-specific likelihoods calibrated in-domain can become miscalibrated when the latent etiology changes mid-episode, especially under sparse, prefix-only supervision. Third, *premature closure*: strong local patterns (e.g., ECG orders $\rightarrow$ ischemia) can anchor the trajectory before counter-signals appear (e.g., pleuritic descriptors, D-dimer, CTA), yielding myopic next-event proposals.

Sequential structure further compounds these risks. Clinical events arrive as partially ordered sets (co-occurring orders, batched lab panels) with irregular time gaps, and many ED timelines are better viewed as *sequences of sets* than pure token strings. Methods that enforce set-level invariance or augment code streams with ontology signal (e.g., CCS/knowledge-graph embeddings) help preserve meaning across co-occurrences and synonyms, but they do not by themselves resolve cross-service drift when the *generator* of observations changes during workup[128] [4]. Likewise, zero-shot generative rollouts over patient-health timelines demonstrate breadth but still reflect the chosen pretraining mix and may blur service-specific decision thresholds [7].

From a probabilistic standpoint, the ED setting resembles large-context inference over many interacting latent causes. Classic graphical approaches (e.g., QMR-DT) formalize comorbidity and conditional dependence but become computationally brittle at realistic scales [59]. Autoregressive transformers approximate these high-order conditionals efficiently, yet as already noted above, a *single* backbone must compromise between compact vocabularies (speed, stability) and domain-specific expressivity (rare tokens, narrow thresholds). In practice, optimizing tokenization and priors for one service can degrade discrimination in another, particularly near safety-critical boundaries.

These observations motivate a *routed, multi-disciplinary engagement* approach. Concretely it: (i) reads the earliest prefix and estimate calibrated probabilities over clinical domains; (ii) consults one or more domain specialists when warranted (with asymmetric, safety-first thresholds for life-threatening etiologies); (iii) arbitrates multi-label suggestions with deterministic, auditable policies; and (iv) fails open when confidence is low or vitals breach guardrails. Such conditional computation aligns with mixture-of-experts and expert-routing advances—activating only the parameters relevant to the emerging hypothesis space while preserving latency and traceability.

**PRISM-Consult** is designed with this approach in mind. By *holding the token template constant*—preserving interpretability and shared embeddings—while *routing* episodes to *domain-specialized fine-tunings*. Each specialist selectively expands or emphasizes a *local* subset of tokens relevant to its domain (e.g., pleural processes, gas exchange markers for pulmonary; ischemic markers and ECG patterns for cardiac), without forcing the global vocabulary to balloon. The router thus acts as a clinical analogue of referral: "right expert, right time," mitigating the specificity–coverage trade-off inherent to a single global model.

### 6.3.2 Related Work

We previously introduced comparable methodologies for PRISM, so at this juncture we'll instead focus specifically on the mixture-of-experts architectures, and where PRISM consult falls in that spectrum.

### Mixture-of-experts and routing among specialists

Mixture-of-Experts (MoE) and conditional computation architectures enable input-dependent parameter activation and have been central to efficient scaling. GShard [68] and Switch Transformer [36] demonstrated that sparse expert routing can train trillion-parameter class models with near-constant per-token cost, provided stability and load-balancing constraints are met. Subsequent work proposed alternative routing rules (e.g., Expert Choice) to improve load balancing and efficiency [132]. In parallel, "routing among models" at inference—choosing which pre-trained system to run for a given query—has become an active area, with routers trained from preference data or heuristics to trade off cost, quality, and latency [93, 130].

Beyond general LLM systems, ensemble/agent-style methods explicitly coordinate multiple specialists. Recent "mixture-of-agents" approaches show that structured collaboration among models can improve reasoning and robustness [119]. In healthcare, most prior work focuses on single-model evaluation (e.g., LLMs for triage or diagnosis) [79, 99] or traditional clinical decision support (CDS) [120]. There remains a gap for lightweight, auditable routers that triage *early* clinical prefixes to compact, domain-specific experts, optimizing safety and compute while preserving traceability—precisely the niche that PRISM-Consult targets.

### Positioning PRISM-Consult

PRISM-Consult operationalizes a routed specialist architecture tailored to ED presentations: (i) compact specialists (same PRISM backbone and schema) trained on domain-consistent corpora; (ii) a calibrated router that reads the first $K$ tokens and dispatches to one or more experts under safety-first thresholds; and (iii) per-domain evaluation and macro-averaging to avoid dominance by common presentations. Compared with monolithic clinical LMs, this design aligns with operational realities (service-specific vocabulary and SLAs), supports explainability (specialist audit trails and calibrated probabilities), and can scale horizontally by adding experts or refining routing policies, drawing on MoE and LLM-routing principles while remaining deployable in resource-limited clinical settings.

### 6.3.3 Methods

#### Overview.

PRISM-Consult extends the original PRISM framework into a routed, multi-specialist system. All data handling, event schema design, tokenization, and baseline modeling choices follow the previously articlulated methodology for PRISM [69]. We instead focus here on the extensions deployed for the PRISM-Consult framework: Specifically an expanded and modified clinical domain, and router protocol.

#### Data and Preprocessing

**Data Source** Data for this study, much like the original PRISM one, was extracted from data from the MIMIC-IV database, an extensive electronic health records repository collected from patients admitted to the Beth Israel Deaconess Medical Center (BIDMC) between 2008 and 2019. The database encompasses detailed clinical information, including demographics, vital signs, laboratory test results, diagnostic procedures, and discharge diagnoses captured during hospital stays [63].

**Data structure and cohort window.** Each record for this model is comprised of a comprehensive longitudinal record of a patient's diagnostic journey, across clinical episodes. Episodes begin at admission and end at patient discharge.

Timelines consist of timestamped *structured* events drawn from:

1. **Patient Information** (E.g., age, gender)
2. **Admission and Discharge Events**
3. **Diagnostics and orders** (e.g., ECG ordered, CTA chest ordered).
4. **Laboratory observations** (e.g., troponin value, D-dimer positive/negative).
5. **Diagnoses** encoded in ICD-9-CM.

Procedures and medication administrations are *intentionally excluded*, mirroring PRISM's emphasis on a concise and clinically interpretable vocabulary focused on diagnostic reasoning. All events are normalized to a patient-relative clock (Initial admission $= 0$) and sorted chronologically and then in a structured manner detailed below to ensure consistency in batched event tokenization.

**Inclusion/exclusion Criteria.** This enhanced study included all adult ED encounters with at least one qualifying initial (pre-diagnosis) symptom code recorded during the index window, along with at least one of the identified terminal diagnoses, in order to have a comprehensive set of patients traversing the diagnostic progression across clinical disciplines.

**Initial (Pre-Diagnosis) ICD-9-CM Code Set — Router Inputs** Figure 6.2 details the set of preliminary diagnostic codes a patient presents with that are notionally indicative of a diagnosis pathway. Inclusion of at least one such code is a requisite for inclusion in the study.

**Final (Conclusive) ICD-9-CM Code Set — Specialist Gold Labels** Figure 6.3 details the set of final, or target, Gold-Label diagnostic codes representing an ultimate diagnostic determination by a clinical expert. Patients must ultimately be diagnosed with one of these conditions for inclusion in the study cohort.

**Sequence construction.** Once eligible patients were determined, their medical histories were extracted and converted to token sequences using a fixed event-precedence policy, based foremost on chronological timing and then a predetermined hierarchy of token types: `DIAG` $>$ `LAB` $>$ `ORDER`, with explicit time-gap markers inserted between events when inter-event intervals exceed prespecified thresholds (e.g., 1 hour, 6 hours) to preserve chronology. A final alphabetical ordering within each token-type is then done to ensure that batched sequences are ordered consistently across episodes (this ensures consistency in prediction accuracy given the singular 'next token' generation schema to the PRISM model). Each sequence is truncated or windowed to a maximum length of 512 tokens for an entire patient chronology, potentially spanning multiple admission

| Category | Symptom / Provisional Descriptor | ICD-9-CM |
|---|---|---|
| Core chest-pain & cardiorespiratory | Chest pain, unspecified | 786.50 |
| | Precordial pain | 786.51 |
| | Other chest pain (incl. pleuritic) | 786.52 |
| | Shortness of breath / dyspnea | 786.05 |
| | Tachypnea | 786.06 |
| | Palpitations | 785.1 |
| | Syncope and collapse | 780.2 |
| Pulmonary-leaning | Hemoptysis | 786.3 |
| | Cough | 786.2 |
| GI-leaning | Heartburn / pyrosis | 787.1 |
| | Nausea and vomiting | 787.0 |
| | Epigastric abdominal pain | 789.06 |
| | Abdominal pain, site unspecified | 789.00 |
| Musculoskeletal / trauma | Costochondritis / Tietze's syndrome (provisional) | 733.6 |
| | Contusion of chest wall | 924.01 |
| | Closed fracture of rib | 807.02 |
| Psychogenic / functional | Panic disorder (episodic paroxysmal anxiety) | 300.01 |
| | Anxiety state, unspecified | 300.00 |
| | Hyperventilation syndrome | 306.4 |

Figure 6.2: Initial set of diagnoses, notionally tied to associated diagnostic grouping

| Specialist | Condition Family | ICD-9-CM (incl. 4th/5th digits) |
|---|---|---|
| Cardiac–Vascular | Acute myocardial infarction (AMI) | 410.xx |
| | Unstable angina / intermediate coronary syndrome | 411.1 |
| | Pericardial–myocardial inflammation (incl. pericarditis/myocarditis)[6] | 423.9 |
| | Aortic dissection (thoracic/abdominal) | 441.0x |
| | Hypertrophic cardiomyopathy (HCM) flare/decompensation | 425.1 |
| Pulmonary | Pulmonary embolism / infarction | 415.1x |
| | Pneumothorax (spontaneous/tension/other) | 512.8x |
| | Pleurisy / pleural processes | 511.x |
| | Severe pneumonia (unspecified organism) | 486 |
| Gastro–Oesophageal | Gastro–oesophageal reflux disease (GERD) | 530.81 |
| | Diffuse oesophageal spasm | 530.5 |
| | Spontaneous oesophageal rupture (Boerhaave) | 530.4 |
| | Peptic-ulcer perforation (gastric / duodenal) | 533.1x; 533.4x |
| Musculoskeletal | Costochondritis / Tietze's syndrome | 733.6 |
| | Rib fracture (closed) | 807.0x |
| | Chest-wall contusion | 924.01 |
| Psychogenic / Functional | Panic disorder | 300.01 |
| | Anxiety state, unspecified | 300.00 |
| | Hyperventilation syndrome | 306.4 |

Figure 6.3: Set of final 'Gold Label' diagnostic codes, representing an ultimate diagnostic determination

episodes. The final (gold-label) diagnosis token is *not* included on the input side for training tasks that predict it, preventing label leakage.

**Tokenization template and vocabulary.** We used the same compact, hand-designed token schema with type prefixes to preserve semantics originally developed for PRISM:[1]

- `[DIAG]_ICD9_`⟨code⟩ for diagnoses (e.g., `[DIAG]_ICD9_786.50`).
- `[OBS]_LAB_`⟨test⟩`:`⟨bin/value⟩ for labs (e.g., `[OBS]_LAB_TROP:HIGH`).[2]
- `[ACTION]_ORD_`⟨order⟩ for diagnostic orders (e.g., `[ACTION]_ORD_ECG`).
- `[GAP]_H`⟨k⟩ for time-gap markers (e.g., `[GAP]_H1` for $>$1 hour).
- Special sentinels: `[BOS]`, `[EOS]`, `[PAD]`, `[UNK]`.

Vocabulary growth is controlled by (i) whitelisting code families relevant to target presentations and (ii) merging infrequent variants into `[UNK]` or higher-level bins (e.g., rare lab subtypes). This yields a compact vocabulary that supports low-latency inference while maintaining clinical interpretability.

**Model backbone (PRISM) Configuration and Training**

The baseline PRISM model is a decoder-only transformer with:

- $L$=6 transformer blocks; model dimension $d_{model}$=256; MLP expansion $4d$; $n_{heads}$=4.
- Learned absolute positional embeddings; tied input/output token embeddings for parameter efficiency.
- Dropout $0.1$ in attention and MLP layers; layer normalization pre-attention.

[1]These templates are inherited from PRISM and reused verbatim. New domains add tokens but do not alter templates.

[2]Continuous labs are discretized into clinically meaningful bins (e.g., LOW/NORMAL/HIGH/CRITICAL) defined a priori clinically-validated thresholds; raw numeric values are not emitted as free-text.

The objective is next-token prediction over the event sequence (autoregressive cross-entropy). For temporal awareness, an auxiliary time-to-next-event head (Huber loss) can be added during pre-training; this head is discarded at inference.

**Optimization** We replicated our training schema, training with AdamW (weight decay $0.01$), linear warmup over the first 5% of steps followed by cosine decay to 10% of the peak learning rate. Typical settings: batch size 64–128 sequences, peak LR in 1e-3 to 2e-4 depending on corpus size, gradient clipping at 1.0, early stopping on dev NDCG@3. Mixed-precision training is used when supported.

**Evaluation (PRISM baseline)** Primary metrics for model performance were Top-1/Top-3 next-event recall, NDCG@k, and calibration (Brier score, reliability curves). For diagnosis-prediction ablations, AUROC/PR for specific gold-label families were measured. All metrics are computed per domain and macro-averaged to avoid dominance by common events.

**PRISM-Consult (panel of experts) Confguration and Training**

PRISM-Consult's framework extends PRISM by the expansion of the following elements:

1. A **router** trained on the earliest events of each episode to emit one or more domain flags (Cardiac–Vascular, Pulmonary, Gastro–Oesophageal, Musculoskeletal, Psychogenic). Inputs are the first 2–5 coded events represented as bag-of-concepts TF–IDF features projected to a 256-dim space, optionally concatenated with simple temporal features (e.g., time since triage). The router is a 2-layer transformer classifier with sigmoid outputs and focal loss to up-weight life-threatening domains.

2. A set of **specialist models**—one per domain—initialized from the PRISM backbone and fine-tuned on domain-filtered corpora defined by the final gold-label codes specified above. To preserve parameter sharing and inference speed, each specialist uses low-rank adaptation (LoRA) on attention and feed-forward layers while keeping shared embeddings frozen;

adapter ranks and learning rates are selected on a held-out dev set per domain.

3. A **dispatch policy** at inference: if any life-threatening domain (e.g., AMI, PE, dissection) exceeds a calibrated threshold, a single high-priority specialist is invoked; otherwise the top-2 domains are consulted in parallel and their suggestions are merged by a deterministic arbitration layer (Cardiac $>$ Pulmonary $>$ Gastro $>$ Musculoskeletal $>$ Psychogenic).

**Labeling surface (initial & final codes)** For each episode, only codes occurring *before* the first definitive diagnosis are considered "initial" to prevent leakage; when multiple definitive diagnoses are present (e.g., AMI + PE), multi-label targets are assigned.

**Calibration, safety, and auditability** All classifier outputs (router and specialists) were temperature-calibrated on a development fold using cross-entropy minimization. We logged router logits, specialist logits, and the final arbitration decision per episode to enable post hoc review. If the router emits uniformly low confidence (all domain probabilities below a safety threshold) or vitals cross pre-defined danger thresholds, the system fails open to parallel consultation of all specialists.

**Compute and reproducibility.** Experiments for the router were run on modern GPUs (e.g., A100-class) using deterministic seeds. Data preprocessing and training pipelines are version-controlled; model checkpoints, tokenizer vocabularies, and ICD-9 codebooks (both initial and final sets) are archived with SHA checksums.

**Overview of Light-Weight Router Design** We model early Emergency Department (ED) episodes as a prefix time-series classification task. After the first $K$ coded events (default $K{=}5$), the router produces calibrated probabilities over five specialist domains: *Cardiac–Vascular*, *Pulmonary*, *Gastro–Oesophageal*, *Musculoskeletal*, and *Psychogenic*. Episodes may be multi-label in principle, though the present corpus yielded disjoint (single-label) episodes. The router dispatches to one or more PRISM-Consult specialists under a safety-first policy.

**Data ingestion and harmonization** Tokenized episodes reside in five domain directories (Cardiac, Pulmonary, Gastro–Oesophageal, Musculoskeletal, Psychogenic).

Each record is mapped to a canonical schema:

$$\text{episode} \;=\; \big(\texttt{episode\_id},\; \mathbf{e} = [e_1, \ldots, e_L],\; \texttt{time\_feats} \in R^{\leq 2}\big),$$

where optional `time_feats` contain simple scalars (minutes-to-first-order; max inter-event gap). Episodes with the same `episode_id` across directories are merged (longest token list retained; non-empty `time_feats` preferred), and a 5-dim multi-hot label $\mathbf{y} \in \{0,1\}^5$ indicates domain membership. In this study, directories were disjoint by construction, so $\|\mathbf{y}\|_0{=}1$.

**Proportional sampling** Let $\mathcal{E} = \{(\mathbf{e}_i, \mathbf{y}_i)\}_{i=1}^{N}$ denote the merged episode table. To obtain a target size $T$ while preserving domain mixture, we compute per-domain prevalence $p_d$ on $\mathcal{E}$ and allocate quotas $q_d \approx p_d T$. We first fill quotas with domain-exclusive episodes, then top up from remaining sets (including potential multi-label cases), avoiding duplicates; if fewer than $T$ remain after deduplication, we top up uniformly at random. Setting $T{=}0$ disables sampling.

**Prefix expansion (anytime supervision)** To enable predictions after each early event, we expand episodes into prefixes of lengths $\ell \in \{1, \ldots, \min(K, L_i)\}$:

$$\mathcal{P} \;=\; \big\{(i, \ell,\; \mathbf{e}_{i,1:\ell},\; \mathbf{y}_i,\; w_\ell = \ell/K)\big\}.$$

Each prefix inherits the final label $\mathbf{y}_i$ but only uses the first $\ell$ tokens as input, preventing label leakage. We optionally use $w_\ell$ as a sample weight to mildly emphasize later (more informative) prefixes.

**Featurization: TF–IDF $\rightarrow$ SVD** We join prefix tokens with spaces to form a short "document" and compute 1–2-gram TF–IDF with `min_df=2`. A 256-dim truncated SVD yields a compact vector $\mathbf{z}_{i,\ell} \in R^{256}$; optional `time_feats` ($\leq$2 scalars) are concatenated to form $\tilde{\mathbf{z}}_{i,\ell}$.

**Router model and calibration** We train one-vs-rest logistic regression heads (saga, L2, $C$=2.0, `max_iter`=3000) for the five domains on the prefix table $\mathcal{P}$, using stratified patient-level splits (70/10/20 train/dev/test on the indicator of any positive label). Probabilities are calibrated per head on the development split with Platt scaling (3-fold). Let $\hat{p}_d(\tilde{\mathbf{z}})$ be the calibrated probability for domain $d$.

**Routing policy and threshold tuning** At inference, with $\hat{\mathbf{p}} = \{\hat{p}_d\}_{d=1}^{5}$:

1. If a life-threatening domain (Cardiac, Pulmonary) satisfies $\hat{p}_d \geq \tau_{\text{hi}}$, route *top-1* (the argmax).
2. Else if $\max_d \hat{p}_d \geq \tau_{\text{lo}}$, route *top-2*.
3. Else, *fail-open* to all five experts.

We grid-search $(\tau_{\text{hi}}, \tau_{\text{lo}})$ on the development split to minimize expected experts per episode subject to a safety constraint on life-threatening recall (Cardiac$\vee$Pulmonary). Unless otherwise noted, we target $\geq 0.98$ on development; if no grid point satisfies the constraint, we select the point with maximal life-threat recall (tie-break: lower compute).

**Safety behavior** If calibrated probabilities are uniformly low (i.e., $\max_d \hat{p}_d$<0.25) or if triage vitals cross pre-defined danger thresholds, the router fails open to all experts and emits an audit record (raw logits, thresholds, selected route). Temperature scaling is re-checked on each model refresh to maintain probability fidelity.

**Evaluation** *Discrimination.* ROC-AUC and PR-AUC are reported per domain on the held-out test set using calibrated probabilities.

**Routing quality.** With routed set $R_i$ and truth $Y_i$ for episode $i$:

$$\text{Recall}_{\text{any}} = \tfrac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \mathbf{1}[R_i \cap Y_i \neq],$$

$$\text{Recall}_{\text{all}} = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \mathbf{1}[Y_i \subseteq R_i].$$

Life-threat recall is computed on:

$\{i : Y_i \cap \{\text{Cardiac}, \text{Pulmonary}\} \neq\}$

and requires $R_i$ to include at least one of these domains.

**Compute proxy and latency.** We report $E[|R|] = \frac{1}{N} \sum_i |R_i|$ and an estimated latency $L_i = L_{\text{router}} + \sum_{d \in R_i} L_d$ using fixed per-expert times.

**Anytime curves.** Metrics are stratified by prefix length $\ell = 1, \ldots, K$ to quantify earliness.

**Anytime performance.** To quantify earliness, we repeat the above at each prefix length $\ell \in \{1, \ldots, K\}$, plotting discrimination and routing recalls as functions of $\ell$.

**Ablations and robustness checks** We assess: (i) $K \in \{2, 3, 5\}$; (ii) text-only vs. text+time features; (iii) uncalibrated vs. calibrated probabilities; and (iv) alternative linear heads (linear SVM with Platt). As sanity baselines we compare against *consult-all*, *fixed Cardiac+Pulmonary*, and a single generalist PRISM variant (no routing).

### 6.3.4 Results

#### Specialist Model Training & Performance

**Cohorts and training setup** A total of $N$=20,436 Emergency Department (ED) episodes met all inclusion criteria for PRISM-Consult. Domain-eligible cohorts are not disjoint (multi-label episodes may appear in multiple domain pools). Where noted, some domains were *scoped* to a capped training subset to ensure compute parity and class balance. All specialists inherit the PRISM backbone and tokenization; optimization and early-stopping follow the protocol in the Methods section. Unless otherwise stated, training used the default causal language modeling loss of autoregressive cross-entropy) with mixed precision.

| Domain | Eligible $N$ | Train $N$ | Best Epoch | Val Loss | PPL | $\Delta$ Val Loss (%) |
|---|---|---|---|---|---|---|
| Cardiac–Vascular | 1,128 | 1,128 | 5 | 0.7917 | 2.21 | −56.97 |
| Pulmonary | 3,150 | 3,150 | 5 | 0.7041 | 2.02 | −52.27 |
| Gastro–Oesophageal | 9,350 (scoped 4,500) | 4,500 | 5 | 0.7004 | 2.01 | −38.03 |
| Musculoskeletal | 523 | 523 | 5 | 1.2692 | 3.56 | −52.87 |
| Psychogenic | 14,924 (scoped 4,500) | 4,500 | 5 | 0.6289 | 1.88 | −29.42 |

Figure 6.4: Development-set performance by specialist model. Perplexity is $\exp(\text{Val Loss})$. $\Delta$ Val Loss is the percentage reduction from epoch 1 to the best epoch. Scoped domains used capped training sets for balance.

**Domain-specific outcomes (Specialist Model Set Performance)** Each specialist exhibited stable convergence with steadily improving validation loss across epochs, consistent with the baseline PRISM behavior on cardiac timelines. Table 6.3.4 reports the best development loss, derived perplexity ($\text{PPL} = \exp(\text{loss})$), and percentage reduction from epoch 1 to the best epoch.

**Cardiac–Vascular (AMI/UA/pericardial/aortic/HCM).** Validation loss decreased from $1.8397$ at epoch 1 to $0.7917$ at epoch 5 ($-56.97\%$; $\text{PPL}=2.21$), closely mirroring the original PRISM cardiac behavior and supporting the transferability of the backbone.

**Pulmonary (PE/pneumothorax/pleura/pneumonia).** Validation loss improved from $1.4753$ to $0.7041$ by epoch 5 ($-52.27\%$; $\text{PPL}=2.02$), indicating a well-captured structure in early pulmonary presentations.

**Gastro–Oesophageal (GERD/spasm/Boerhaave/ulcer perforation).** On a scoped training pool of 4,500 episodes, validation loss fell from $1.1303$ to $0.7004$ ($-38.03\%$; $\text{PPL}=2.01$) by epoch 5, with steady epoch-on-epoch gains and no signs of overfitting.

**Musculoskeletal (costochondritis/rib injury/chest-wall contusion).** Despite the smallest cohort ($N=523$), validation loss dropped from $2.6927$ to $1.2692$ ($-52.87\%$; $\text{PPL}=3.56$). The higher perplexity is consistent with data scarcity and etiologic heterogeneity; nevertheless, the relative improvement and stable trajectory indicate learnability under limited data.

**Psychogenic (panic/anxiety/hyperventilation).** Training on a scoped subset of 4,500 episodes converged from $0.8910$ to $0.6289$ ($-29.42\%$; PPL$=1.88$), the lowest perplexity among the specialists, reflecting a relatively compact symptom-to-diagnosis mapping for this domain.

**Cross-domain interpretation** Figure 6.5 visualizes the training of all specialist models across epochs. All five specialists converge to low development perplexities (three near $\sim 2.0$ and one below $2.0$), with smooth validation curves and substantial loss reductions from epoch 1 (Table 6.3.4). This *cross-disciplinary consistency*—achieved without altering tokenization, model size, or optimization hyperparameters—validates our central design choice: the compact PRISM backbone, when routed to domain-consistent corpora, yields strong and stable learning dynamics beyond its original cardiac scope. The musculoskeletal model's higher perplexity likely reflects limited sample size and broader label variance; we anticipate improvements with targeted data augmentation and modest adapter-rank increases.

**Training curves (dev) and calibration** For each specialist, validation loss decreased monotonically through epoch 5 with no divergence from training loss, suggesting adequate regularization (dropout 0.1; early stopping at epoch 5). Final checkpoints correspond to the minima reported above. Temperature scaling for probabilistic outputs is fit on the development fold.

### Router Training Results

**Cohort and training rows** Across domain directories we ingested $N{=}13{,}801$ unique episodes: Cardiac $1{,}128$ (8.2%), Pulmonary $3{,}150$ (22.8%), Gastro–Oesophageal $4{,}500$ (32.6%), Musculoskeletal $523$ (3.8%), Psychogenic $4{,}500$ (32.6%). Episodes were single-label by design (no cross-directory duplicates). Prefix expansion with $K{=}5$ yielded $69{,}005$ training/evaluation rows.

### Cohort and Routing Policy Summary

**Selected thresholds and routing mix** Fig 5. details specific thresholds and overall results mix. The development grid selected $(\tau_{\text{hi}}, \tau_{\text{lo}}) = (0.70,\ 0.30)$ under our objective. With these thresholds,

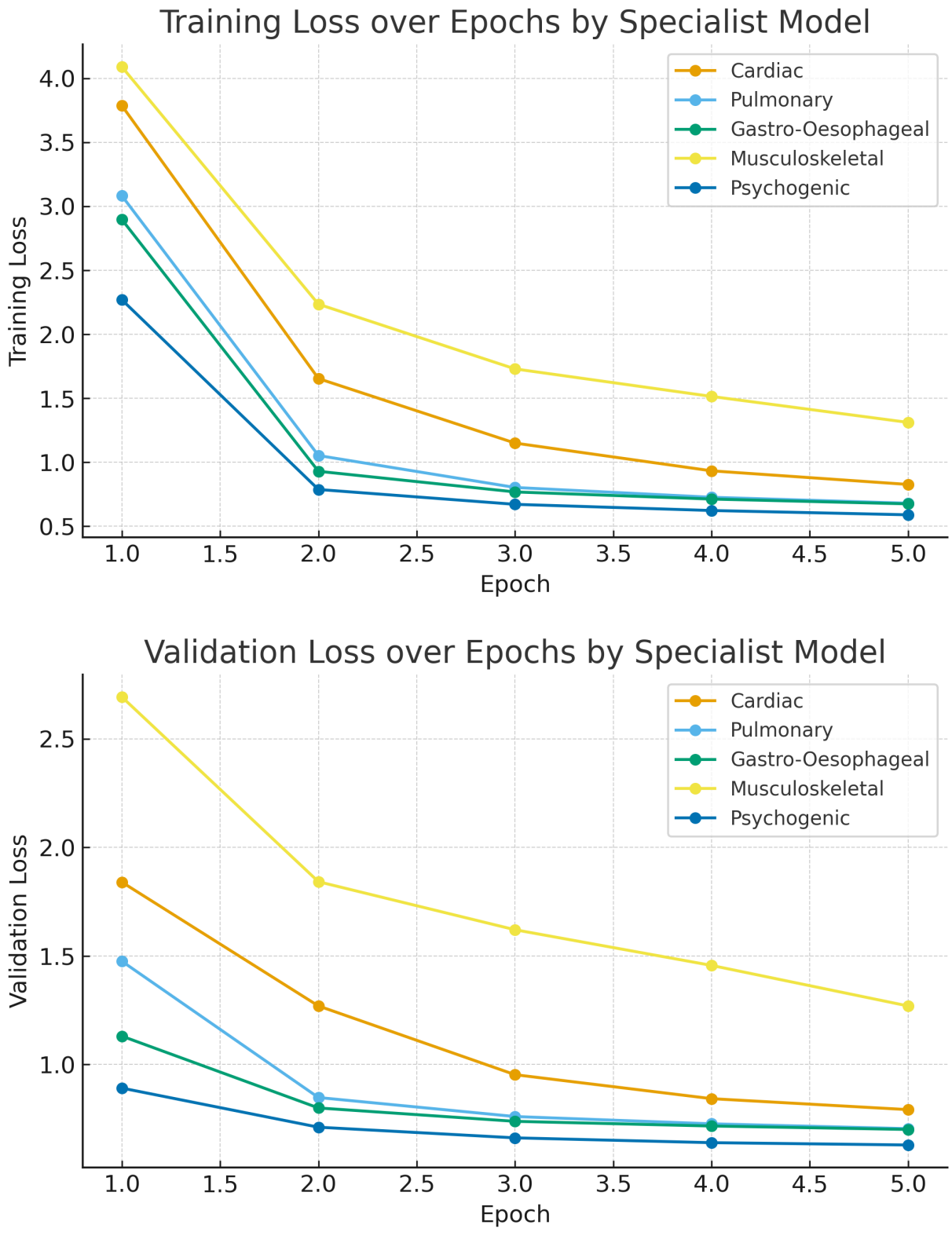



Figure 6.5: Cross-specialist Training Loss over epochs for both training and validation cohorts

| | Card | Pulm | GI | MSK | Psych |
|---|---|---|---|---|---|
| Episodes (unique) | 1,128 | 3,150 | 4,500 | 523 | 4,500 |
| Share (%) | 8.2 | 22.8 | 32.6 | 3.8 | 32.6 |
| Prefixes ($K$=5) | 69,005 total rows | | | | |
| Selected thresholds | $\tau_{\text{hi}} = 0.70,\ \tau_{\text{lo}} = 0.30$ (dev grid) | | | | |
| Dev life-threat recall | 0.973 (Card $\vee$ Pulm) | | | | |
| Dev $\text{recall}_{\text{any}}$ / $\text{recall}_{\text{all}}$ | 1.000 / 0.945 | | | | |
| Dev expected experts | 1.549 (vs. 5.0 consult-all; $\sim$69% reduction) | | | | |
| Dev latency (ms) | 88.4 (router + routed specialists) | | | | |

Figure 6.6: Light-Weight Router Training Cohort and Overall Results Summary

the router predominantly returned *top-1* or *top-2* routes; fail-open was rare. The expected consulted experts per episode on test was $1.565$, a $\sim 69\%$ reduction versus consult-all (5 experts).

**Discrimination (per-domain; macro-averaged)**

Fig. 6 details domain specific discriminative results, along with macro-averaged results across domains for both the development and testing cohorts.

**Safety and routing quality** On the development split, life-threatening recall (Cardiac$\vee$Pulmonary) was $0.973$; on test it was $0.965$. Both $Recall_{any}$ metrics were $1.000$ (the routed set always included a correct domain), and $Recall_{all}$ was $0.945$ (dev) and $0.942$ (test). Estimated latencies, using the fixed per-expert constants, were $88.4$ ms (dev) and $89.5$ ms (test).

### 6.3.5 Discussion

**Cross-disciplinary performance of PRISM**

PRISM-Consult extends the original PRISM design from a cardiology-focused study to a routed, multi-specialist system while preserving the compact backbone and token template. Empirically,

| Domain | Dev | | Test | |
|---|---|---|---|---|
| | AUROC | AP | AUROC | AP |
| Cardiac | 0.984 | 0.985 | 0.964 | 0.958 |
| Pulmonary | 0.965 | 0.960 | 0.968 | 0.964 |
| Gastro–Oesophageal | 0.970 | 0.972 | 0.974 | 0.978 |
| Musculoskeletal | 0.939 | 0.910 | 0.951 | 0.936 |
| Psychogenic | 0.972 | 0.969 | 0.961 | 0.958 |
| **Macro-avg** | **0.966** | **0.959** | **0.964** | **0.959** |

Figure 6.7: Domain-specific Discriminative results

all five specialists trained with the same optimization recipe and achieved smooth, monotonic declines in development loss with low final perplexities (near $\sim$2.0 for Cardiac, Pulmonary, Gastro–Oesophageal, and Psychogenic; higher but improving for Musculoskeletal). This cross-disciplinary consistency, obtained without increasing model width/depth or altering tokenization, indicates that the PRISM backbone transfers effectively when exposed to domain-consistent corpora and calibrated with light routing. That is, a single representational space and schema can support heterogeneous organ-system reasoning as long as data partitioning and supervision align with clinical provenance.

Additionally the router's near-perfect $Recall_{any}$ and high $Recall_{all}$ further suggest that early tokens carry enough discriminative signal to select the appropriate specialist(s) reliably, even under strict latency constraints.

**Next steps: validation and extension**

In our research we outlined three directions to harden and extend this framework further:

1. **External and temporal validation.** Replicate training/evaluation on (i) a second site with

distinct coding habits and lab panels; (ii) a temporally held-out slice to assess drift.

2. **Safety-first routing refinements.** Enforce a hard development constraint of life-threatening recall $\geq 0.98$ via (a) asymmetric thresholds (lower $\tau_{\text{hi}}$ for Cardiac/Pulmonary than for other domains), (b) a guardrail that includes both Cardiac and Pulmonary whenever $\max(\hat{p}_{\text{card}}, \hat{p}_{\text{pulm}}) \geq \tau_{\text{lo}}^{\text{life}}$, and (c) per-head isotonic calibration when under-confidence is detected near decision boundaries. Re-report the compute–safety Pareto frontier (expected experts vs. life-threat recall).

3. **Broader coverage and richer inputs.** Add specialists (e.g., aortic catastrophes vs. general vascular, neurologic mimics) as new gold-label families mature; incorporate small, domain-agnostic time features (minutes-to-first-order; max inter-event gap) that are already supported by the router. Where multi-label episodes exist (e.g., AMI+PE), explicitly train/evaluate multi-label routing quality (*Recall*$_{all}$ at top-$k$) and measure arbitration behavior.

Operationally, we recommend prefix-stratified audits (metrics by $\ell = 1..K$), route-mix telemetry (top-1/top-2/fail-open), and episode-level audit logs (router/specialist logits, thresholds, decision) to support prospective QA and IRB-facing safety reviews.

### Limits

Three limitations temper interpretation. First, the present corpus is effectively single-label across directories, so the multi-label routing regime was not strongly exercised; future cohorts should include confirmed multi-diagnosis cases to probe arbitration. Second, class imbalance (e.g., the relatively small Musculoskeletal pool) likely contributes to higher perplexity and calibration variance for that head; targeted sampling, modest adapter rank increases, and data augmentation (token normalization for near-synonymous events) are straightforward mitigations. Third, learned absolute positional embeddings fix the maximum context length and may not extrapolate; if longer horizons are needed, rotary or ALiBi encodings can be substituted with minimal disruption to the rest of the stack. Finally, threshold selection in the current run favored a near-constraint point (life-

threat recall $< 0.98$ on test); future reports should hard-enforce prespecified safety constraints and present confidence intervals via bootstrap over episodes.

### 6.3.6 Conclusion

PRISM-Consult operationalizes a clinician-aligned *panel-of-experts* by pairing a calibrated, lightweight router with parameter-efficient PRISM specialists trained on domain-consistent corpora. Using only the earliest tokens, the router attains high coverage and substantial compute savings relative to consult-all, while specialists exhibit smooth, low-perplexity convergence across disparate organ systems—evidence that the PRISM backbone generalizes beyond its original cardiac scope. With minor policy and calibration refinements to meet strict life-threat recall targets, the framework provides an auditable, low-latency pathway to deployable clinical decision support that routes each episode to the right expert at the right time.

# Chapter 7

# Conclusion

The work presented here spans several critical domains including—mental-health systems, data collection and modeling, and AI assurance, model optimization, and charting the diagnostic pathways of patients in clinical environments—reflecting a cohesive effort to develop technology systems and AI models that are both technically robust and deeply rooted in human needs and motivations. Across these initiatives, human motivations have consistently guided and influenced the evolution of the work: from how data are collected, to when users are asked to respond, to what models are allowed to do, and how their decisions are explained.

**Human-mediated systems for mental health.** The mental-health systems design initiative exemplifies the fusion of IT systems design, hardware integration, and AI capabilities with human-centered concerns. Motivations to enhance efficiency in diagnosis and treatment planning—and to integrate therapeutic relationships—often collided with hardware limitations, UX constraints, privacy expectations, and the known fragility of self-report. This required an ongoing, pragmatic balance: using passive sensing to lower burden, asking for input only when it is likely to be useful, and making the system an empathetic partner that amplifies clinicians rather than replacing them.

Concretely, we evolved a multimodal framework that combines wearable and mobile signals with just-in-time user prompts. An *opportunistic sampling* policy (§2.4) issued EMAs only when recent data are predictive of acute distress, capped by an anti-fatigue budget.

**Labeling amorphous states, and what "an episode" means.** We documented the practical difficulty of mapping amorphous affective states onto discrete labels in continuous time. Users resisted in-situ labeling of stress; daily retrospectives were noisy and prone to telescoping; and accuracy was unobservable even when completion was verifiable. The resulting methodology emphasized (i) explicit time-windowing and baselining, (ii) hysteresis to avoid alert oscillation, and (iii) an action-aware label semantics (labels that actually trigger something). Empirically, our multi-modal episode classifier, adopting a Late-fusion + Supervised Training + Contrastive Regularization achieved an *AUC/PR-AUC = 73%*.

**Assurance: from cross-model correlation to information-centric evaluation.** Motivated by the growing need for independent verification of AI systems, we studied whether cross-model neuronal correlations could predict performance and generalizability across architectures and datasets. We observed encouraging signals in controlled, same-data settings, but results degraded under domain shift and non-comparable feature maps, underscoring limits of correlation-based surrogates. Data-sufficiency tests also proved unsatisfactory, with simple effect-size heuristics failing to align with model performance or sample size sufficiency. These findings motivated the pivot to information-centric criteria that we later operationalized in our compression work—treating information preservation (rather than raw correlation) as the object to optimize.

**M2M-DC: efficient models via information-guided pruning and brief distillation.** To make deployment feasible in footprint- and latency-constrained settings, we introduced MI-to-Mid Distilled Compression (M2M-DC), a residual-safe recipe that couples block-level mutual-information pruning with planes-aware inner slicing and a short, staged knowledge-distillation "repair." Across residual and inverted-residual families, M2M-DC produced strong accuracy–efficiency trade-offs with simple, hardware-friendly edits. For example:

- **ResNet-34 (CIFAR-100):** Top-1 improved by $\approx$ **+2.0** points while reducing parameters and multiply–adds by $\approx$ **74%** each.

- **MobileNetV2:** Top-1 improved by $\approx$ **+2.5** points with $\approx -$**27%** parameters and $\approx -$**24%** compute.

These results show that information-guided pruning, when paired with minimal distillation, can deliver compact models that maintain—or even exceed—baseline accuracy, offering a practical path to clinic-adjacent deployments and on-device inference.

**PRISM and PRISM-Consult: care as next-action prediction.** Finally, we reframed clinical decision support as *next-action prediction* over structured EHR event streams. PRISM, a compact language model for clinical timelines, trained cleanly with a shared backbone and domain-specific vocabularies; specialist models reached low development perplexities (four of five domains near $\approx$ **2.0**), supporting transfer without enlarging capacity. Building on this, PRISM-Consult uses a lightweight, calibrated router to consult only the specialists needed for a given episode. On test sets, the router achieved **Recall_any = 1.000** and **Recall_all** $\approx$ **0.94**, preserved safety for life-threatening domains (**recall** $\approx$ **0.965**), and reduced consulted experts to an average of **1.565** per episode ($\sim$**69%** fewer than consult-all), meeting strict latency and burden constraints.

**Limitations and lessons.** Three themes recur. First, truthful labeling is context-dependent: rapport and timing matter at least as much as questionnaire design. Second, evaluation proxies can mislead when architectures, datasets, or label schemas differ; information-centric objectives travel better across domains. Third, deployment constraints (privacy, compute, latency, clinician cognitive load) are not afterthoughts—they shape what "good" modeling looks like from the outset.

**Outlook: from signals to steps.** Together, these results outline a principled path from human-in-the-loop sensing and labeling, through assurance-driven efficiency, to fast, specialist-routed clinical next-action prediction. The next phase is to close the loop: *(i)* embed the opportunistic EMA policy into routine care with clinician-configured tiers, *(ii)* extend M2M-DC to sequence models and streaming inference on commodity hardware, and *(iii)* prospectively evaluate PRISM-Consult

in workflow, measuring not only predictive metrics but *decision quality, time-to-action*, and *user trust*.

Throughout this work, the fundamental recognition that these systems and analytics are intrinsically intertwined with users and stakeholders has been the overriding consideration in their development. It is a recognition that ultimately no system can stand apart from the limitations and restrictions its human users will ultimately impose upon it. Incorporating this mandate into our work has forced it in unexpected and exciting ways, and ultimately resulted in systems sensitive of and accountable to, the users they seek to assist.

# Bibliography


[1] High-relevance (hrel) mutual-information-based filter pruning for convolutional networks. *arXiv preprint arXiv:2202.10716*, 2022. Layer-wise MI ranking with iterative prune–retrain; baseline we reimplement.

[2] A. Ng, M. Rddy, and S.M. Schueller. Veterans' perspectives on fitbit use in treatment for post-traumatic stress disorder: An interview study. *MIR Mental Health*, 2018.

[3] Gavin Andrews and Tim Slade. Interpreting scores on the kessler psychological distress scale (k10). *Australian and New Zealand journal of public health*, 25(6):494–497, 2001.

[4] Anonymous. Sequential diagnosis prediction with transformer and ontological representation. *IEEE Journal of Biomedical and Health Informatics*, 2023.

[5] Anonymous. Automatic differential diagnosis using transformer-based multi-label sequence classification. 2024.

[6] Anonymous. Transformer-based deep learning model for the diagnosis of lung cancer in primary care. *npj Digital Medicine*, 2024.

[7] Anonymous. A transformer-based model for zero-shot health trajectory prediction. *medRxiv*, 2024.

[8] American College Health Association et al. American college health association–national health assessment ii: Reference group executive summary spring 2016, 2016.

[9] American Psychological Association. What are anxiety disorders? https://www.psychiatry.org/patients-families/anxiety-disorders/what-are-anxiety-disordersg, 2019.

[10] Australian Government Australian Bureau of Statistics. Chapter - k10 scoring.

[11] Hajra Banoo, Vibha Gangwar, and Nusrat Nabi. Effect of cold stress and the cold pressor test on blood pressure and heart rate. 2016.

[12] Anna M Bardone, Terrie E Moffitt, Avshalom Caspi, Nigel Dickson, Warren R Stanton, and Phil A Silva. Adult physical health outcomes of adolescent girls with conduct disorder, depression, and anxiety. *Journal of the American Academy of Child & Adolescent Psychiatry*, 37(6):594–601, 1998.

[13] Feras A Batarseh, Laura Freeman, and Chih-Hao Huang. A survey on artificial intelligence assurance. *Journal of Big Data*, 8(1):60, 2021.

[14] Neeltje M Batelaan, Jan Spijker, Ron de Graaf, and Pim Cuijpers. Mixed anxiety depression should not be included in dsm-5. *The Journal of nervous and mental disease*, 200(6):495–498, 2012.

[15] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilović, et al. Ai fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development*, 63(4/5):4–1, 2019.

[16] Rebecca H Bitsko, Joseph R Holbrook, Reem M Ghandour, Stephen J Blumberg, Susanna N Visser, Ruth Perou, and John T Walkup. Epidemiology and impact of health care provider–diagnosed anxiety and depression among us children. *Journal of developmental and behavioral pediatrics: JDBP*, 39(5):395, 2018.

[17] Mehdi Boukhechba, Philip Chow, Karl Fua, Bethany A Teachman, Laura E Barnes, et al. Predicting social anxiety from global positioning system traces of college students: feasibility study. *JMIR mental health*, 5(3):e10101, 2018.

[18] Ronny Bruffaerts, Philippe Mortier, Glenn Kiekens, Randy P Auerbach, Pim Cuijpers, Koen Demyttenaere, Jennifer G Green, Matthew K Nock, and Ronald C Kessler. Mental health problems in college freshmen: Prevalence and academic functioning. *Journal of affective disorders*, 225:97–103, 2018.

[19] Jana Campbell and Ulrike Ehlert. Acute psychosocial stress: Does the emotional stress response correspond with physiological responses? *Psychoneuroendocrinology*, 37:1111–34, 01 2012.

[20] Jana Campbell and Ulrike Ehlert. Acute psychosocial stress: does the emotional stress response correspond with physiological responses? *Psychoneuroendocrinology*, 37(8):1111–1134, 2012.

[21] Sam Cartwright-Hatton. Anxiety of childhood and adolescence: Challenges and opportunities, 2006.

[22] Y. Chen and Z. Wang. An effective information theoretic framework for channel pruning. *IEEE Transactions on Neural Networks and Learning Systems*, 2024. Also available as arXiv:2408.16772.

[23] Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *arXiv preprint*, 2024. Comprehensive survey of pruning across CNNs/ViTs/LLMs; taxonomy and practical guidance.

[24] 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):e6820, 2017.

[25] Jenny Chum, Min Suk Kim, Laura Zielinski, Meha Bhatt, Douglas Chung, Sharon Yeung, Kathryn Litke, Kathleen McCabe, Jeff Whattam, Laura Garrick, et al. Acceptability of the fitbit in behavioural activation therapy for depression: a qualitative study. *BMJ Ment Health*, 20(4):128–133, 2017.

[26] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature, 2018.

[27] Jack Clark and Gillian K Hadfield. Regulatory markets for ai safety. *arXiv preprint arXiv:2001.00078*, 2019.

[28] Mayo Clinic, 2018.

[29] Rand Corporation. Improving the quality of mental health care for veterans: Lessons from rand research, 2019.

[30] Patrick W. Corrigan and Amy C. Watson. The paradox of self-stigma and mental illness. *Clinical Psychology: Science and Practice*, 9(1):35–53, 2002.

[31] Joana C. Costa, Tiago Roxo, Hugo Proença, and Pedro Ricardo Morais Inácio. How deep learning sees the world: A survey on adversarial attacks and defenses. *IEEE Access*, 12:61113–61136, 2024.

[32] Sajad Darabi, Babak Moatamed, Wenhao Huang, Migyeong Gwak, Casey Metoyer, Mike Linn, and Majid Sarrafzadeh. Heart rate compression & time reduction method for hrv monitoring in athletes. In *2017 IEEE Healthcare Innovations and Point of Care Technologies (HI-POCT)*, pages 152–155. IEEE, 2017.

[33] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[34] Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *International Conference on Machine Learning (ICML)*, pages 2943–2952, 2020.

[35] Shayan Fazeli, Lionel Levine, Mehrab Beikzadeh, Baharan Mirzasoleiman, Bita Zadeh, Tara Peris, and Majid Sarrafzadeh. Passive monitoring of physiological precursors of stress leveraging smartwatch data. In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 2893–2899. IEEE, 2022.

[36] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. 2021.

[37] Sajjad Ghiasvand, Haniyeh Ehsani Oskouie, Mahnoosh Alizadeh, and Ramtin Pedarsani. Few-shot adversarial low-rank fine-tuning of vision-language models. *arXiv preprint arXiv:2505.15130*, 2025.

[38] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15190, 2023.

[39] Jiaqi Gong, Yu Huang, Philip I Chow, Karl Fua, Matthew S Gerber, Bethany A Teachman, and Laura E Barnes. Understanding behavioral dynamics of social anxiety among college students through smartphone sensors. *Information Fusion*, 49:57–68, 2019.

[40] Alex Guilherme. Ai and education: the importance of teacher and student relations. *AI & society*, 34:47–54, 2019.

[41] Migyeong Gwak. *Internet of Things (IoT)-Enabled Health Monitoring Systems: Implementation and Validation*. PhD thesis, University of California, Los Angeles, 2022.

[42] Migyeong Gwak, Shayan Fazeli, Ghazaal Ershadi, Majid Sarrafzadeh, Melina Ghodsi, Afshin Aminian, and John A Schlechter. Extra: exercise tracking and analysis platform for remote-monitoring of knee rehabilitation. In *2019 IEEE 16th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pages 1–4. IEEE, 2019.

[43] Arnar Ingi Halldórsson, Jóhann Ívar Björnsson, and Sveinn Björnsson. *AGR System Monitor*. PhD thesis, 2019.

[44] Paul Hammerness, Theresa Harpold, Carter Petty, Chantal Menard, Claire Zar-Kessler, and Joseph Biederman. Characterizing non-ocd anxiety disorders in psychiatrically referred children and adolescents. *Journal of affective disorders*, 105(1-3):213–219, 2008.

[45] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *International Conference on Learning Representations (ICLR)*, 2016. Workshop track.

[46] Song Han, Jeff Pool, John Tran, and William J Dally. Learning both weights and connections for efficient neural networks. In *Advances in neural information processing systems (NIPS)*, pages 1135–1143, 2015.

[47] Shayan Hassantabar, Joe Zhang, Hongxu Yin, and Niraj K Jha. Mhdeep: Mental health disorder detection system based on wearable sensors and artificial neural networks. *ACM Transactions on Embedded Computing Systems*, 21(6):1–22, 2022.

[48] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[49] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[50] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, pages 1389–1397, 2017.

[51] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[52] United States White House. Reducing the economic burden of unmet mental health needs, Jun 2022.

[53] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[54] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

[55] Qingyuan Hu. A survey of adversarial example toolboxes. In *2021 2nd International Conference on Computing and Data Science (CDS)*, pages 603–608. IEEE, 2021.

[56] Yu Huang, Haoyi Xiong, Kevin Leach, Yuyan Zhang, Philip Chow, Karl Fua, Bethany A Teachman, and Laura E Barnes. Assessing social anxiety using gps trajectories and point-of-interest data. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 898–903, 2016.

[57] *Artificial Intelligence Research Development and Regulation Adopted by the IEEE-USA Board of Directors 10 Feb. 2017*, 2017.

[58] J. Chum, M.S. Zielinski, M. Bhatt, D. Chung, S.Yeung and Z. Samaan. Acceptabiity of the fitbit in behavioural activation therapy for depression: a qualitative study. *Evidence-Based Mental health*, 2017.

[59] Tommi S. Jaakkola and Michael I. Jordan. Variational probabilistic inference and the qmr-dt network. *Journal of Artificial Intelligence Research*, 10:291–322, 1999.

[60] Benoit Jacob, Skirmantas Kursun, Kameron Swersky, Z. M., J. H., D. G., F. K., and R. R. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 2704–2713, 2018.

[61] Ng Chirk Jenn. Designing a questionnaire. *Malaysian family physician : the official journal of the Academy of Family Physicians of Malaysia*, 1(1):32–35, 2006.

[62] Gaojie Jin, Xinping Yi, and Xiaowei Huang. Neuronal correlation: a central concept in neural network. *CoRR*, abs/2201.09069, 2022.

[63] Alistair E. W. Johnson, Leo Bulgarelli, Lucas Shen, Amaia Gayles, Amir Shammout, Steven Horng, Tom J. Pollard, Benjamin Moody, Benjamin Gow, Li-wei H. Lehman, Leo A. Celi, and Roger G. Mark. Mimic-iv, a freely accessible electronic health record dataset. *Scientific Data*, 10(1):1–18, 2023.

[64] Zachary King, Judith Moskowitz, Laurie Wakschlag, and Nabil Alshurafa. Predicting perceived stress through mirco-emas and a flexible wearable ecg device. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, pages 106–109, 2018.

[65] Lydia Kogler, Veronika I Müller, Amy Chang, Simon B Eickhoff, Peter T Fox, Ruben C Gur, and Birgit Derntl. Psychosocial versus physiological stress—meta-analyses on deactivations and activations of the neural correlates of stress reactions. *Neuroimage*, 119:235–251, 2015.

[66] Yann LeCun. The mnist database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998.

[67] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Snip: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations (ICLR)*, 2019.

[68] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, et al. Gshard: Scaling giant models with conditional computation and automatic sharding. 2020.

[69] L. Levine, J. Santerre, et al. PRISM: A transformer-based language model of structured clinical event data. 2025.

[70] Lionel M Levine, Migyeong Gwak, Kimmo Kärkkäinen, Shayan Fazeli, Bita Zadeh, Tara Peris, Alexander S Young, and Majid Sarrafzadeh. Anxiety detection leveraging mobile passive sensing. In *EAI International Conference on Body Area Networks*, pages 212–225. Springer, 2020.

[71] Fuzhong Li, Peter Harmer, Terry E. Duncan, Sue C. Duncan, Alan Acock, and Takashi Yamamoto. Confirmatory factor analysis of the task and ego orientation in sport questionnaire with cross-validation. *Research Quarterly for Exercise and Sport*, 69(3):276–283, 1998. PMID: 9777664.

[72] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In *International Conference on Learning Representations (ICLR)*, 2017.

[73] Daniel Liu, Ronald Yu, and Hao Su. Extending adversarial attacks and defenses to deep 3d point cloud classifiers. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2279–2283, 2019.

[74] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, pages 2736–2744, 2017.

[75] Lotus Health AI. Transforming disease prediction with LotusAI-Predict: A fine-tuned LLaMA model. Lotus Health AI Blog, 2025. Accessed: 2025-04-11.

[76] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[77] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[78] et al Margarita Triguero-Mas. Natural outdoor environments and mental and physical health: Relationships and mechanisms. *Environment International*, 77:35–41, 04 2015.

[79] Lukas Masanneck, Marcel Günther, Sebastian Körber, et al. Triage performance across large language models and healthcare professionals. *Journal of Medical Internet Research*, 26:e53297, 2024.

[80] Inc. Mental Health America. Access to care data 2022.

[81] Kathleen Ries Merikangas, Jian-ping He, Marcy Burstein, Sonja A Swanson, Shelli Avenevoli, Lihong Cui, Corina Benjet, Katholiki Georgiades, and Joel Swendsen. Lifetime prevalence of mental disorders in us adolescents: results from the national comorbidity survey replication–adolescent supplement (ncs-a). *Journal of the American Academy of Child & Adolescent Psychiatry*, 49(10):980–989, 2010.

[82] Maura Mitrushina, Kyle B Boone, Jill Razani, and Louis F D'Elia. *Handbook of normative data for neuropsychological assessment*. Oxford University Press, 2005.

[83] Decebal Constantin Mocanu et al. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. In *Nature Communications*, volume 9, page 2383, 2018.

[84] Mohammad Mofatteh. Risk factors associated with stress, anxiety, and depression among university undergraduate students. *AIMS public health*, 8(1):36, 2021.

[85] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. In *International Conference on Learning Representations (ICLR)*, 2017.

[86] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *2016 IEEE Conference on CVPR*, pages 2574–2582, 2016.

[87] Lalli Myllyaho, Mikko Raatikainen, Tomi Männistö, Tommi Mikkonen, and Jukka K Nurminen. Systematic literature review of validation methods for ai systems. *Journal of Systems and Software*, 181:111050, 2021.

[88] Muhammad Muzammal Naseer, Salman H Khan, Muhammad Haris Khan, Fahad Shahbaz Khan, and Fatih Porikli. Cross-domain transferability of adversarial perturbations. *Advances in Neural Information Processing Systems*, 32, 2019.

[89] Jill M Newby, Louise Mewton, Alishia D Williams, and Gavin Andrews. Effectiveness of transdiagnostic internet cognitive behavioural treatment for mixed anxiety and depression in primary care. *Journal of Affective Disorders*, 165:45–52, 2014.

[90] NIST. Artificial intelligence risk management framework (ai rmf 1.0). Technical report, US Department of Commerce, 2023.

[91] National Institute of Mental Health.

[92] Albertine J Oldehinkel, Johan Ormel, Nienke M Bosch, Esther MC Bouma, Arie M Van Roon, Judith GM Rosmalen, and Harriëtte Riese. Stressed out? associations between perceived and physiological stress responses in adolescents: The trails study. *Psychophysiology*, 48(4):441–452, 2011.

[93] Ian Ong, Aohan Huang, Peng Lu, et al. Routellm: Learning to route llms with preference data. OpenReview, 2024.

[94] Haniyeh Ehsani Oskouie and Farzan Farnia. Interpretation of neural networks is susceptible to universal adversarial perturbations. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2022.

[95] Shane O'Sullivan, Nathalie Nevejans, Colin Allen, Andrew Blyth, Simon Leonard, Ugo Pagallo, Katharina Holzinger, Andreas Holzinger, Mohammed Imran Sajid, and Hutan Ashrafian. Legal, regulatory, and ethical frameworks for development of standards in artificial intelligence (ai) and autonomous robotic surgery. *The international journal of medical robotics and computer assisted surgery*, 15(1):e1968, 2019.

[96] Dimitrios P Panagoulias, Maria Virvou, and George A Tsihrintzis. Regulation and validation challenges in artificial intelligence-empowered healthcare applications—the case of blood-retrieved biomarkers. In *Joint Conference on Knowledge-Based Software Engineering*, pages 97–110. Springer, 2022.

[97] Martin Pinquart and Silvia Sörensen. Gender Differences in Self-Concept and Psychological Well-Being in Old Age: A Meta-Analysis. *The Journals of Gerontology: Series B*, 56(4):P195–P213, 07 2001.

[98] NL Rane, SK Mallick, O Kaya, and J Rane. Tools and frameworks for machine learning and deep learning: A review. *Applied Machine Learning and Deep Learning: Architectures and Techniques*, pages 80–95, 2024.

[99] S. R. Ranji and A. F. Hernandez-Boussard. Large language models—misdiagnosing diagnostic reasoning? *JAMA Network Open*, 7(10):e2433381, 2024.

[100] M. et al. Rostami. Information theoretic pruning of coupled channels in deep neural networks. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2025.

[101] Manthan Sanghavi. Software for explainable ai. In *Explainable, Interpretable, and Transparent AI Systems*, pages 266–278. CRC Press.

[102] Victor Sanh, Thomas Wolf, and Alexander M. Rush. Movement pruning: Adaptive sparsity by fine-tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[103] L. Sauer et al. Pruning by explaining revisited: Optimizing attribution methods to prune cnns and transformers. *arXiv preprint*, 2024. Also circulated 2025; ResearchGate/arXiv.

[104] Dhruv R Seshadri, Ryan T Li, James E Voos, James R Rowbottom, Celeste M Alfes, Christian A Zorman, and Colin K Drummond. Wearable sensors for monitoring the physiological and biochemical profile of the athlete. *NPJ digital medicine*, 2(1):1–16, 2019.

[105] Alexandra L Shilton, Robin Laycock, and Sheila G Crewther. The maastricht acute stress test (mast): Physiological and subjective responses in anticipation, and post-stress. *Frontiers in psychology*, 8:567, 2017.

[106] American Psychiatric Society, 2020.

[107] Matthias Spielkamp. Algorithmwatch: What role can a watchdog organization play in ensuring algorithmic accountability? *Transparent Data Mining for Big and Small Data*, pages 207–215, 2017.

[108] Thomas Stevens. Physical activity and mental health in the United States and Canada: Evidence from four population surveys. *Preventive Medicine*, 17:35–47, 01 1988.

[109] Mark Sujan, Cassius Smith-Frazer, Christina Malamateniou, Joseph Connor, Allison Gardner, Harriet Unsworth, and Haider Husain. Validation framework for the use of ai in healthcare: overview of the new british standard bs30440. *BMJ Health & Care Informatics*, 30(1), 2023.

[110] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *Proc. ICLR*, pages 1–15, 2014.

[111] Chuanqi Tao, Jerry Gao, and Tiexin Wang. Testing and quality validation for ai software–perspectives, issues, and practices. *IEEE Access*, 7:120164–120175, 2019.

[112] Texas A&M Engineering. Veterans take wearable ptsd monitor on test ride. 2017.

[113] Maurice Topper, Paul MG Emmelkamp, Ed Watkins, and Thomas Ehring. Prevention of anxiety disorders and depression by targeting excessive worry and rumination in adolescents and young adults: a randomized controlled trial. *Behaviour research and therapy*, 90:123–136, 2017.

[114] Konstantinos Tsamakis, Andreas S Triantafyllis, Dimitrios Tsiptsios, Eleftherios Spartalis, Christoph Mueller, Charalampos Tsamakis, Sofia Chaidou, Demetrios A Spandidos, Lampros Fotis, Marina Economou, et al. Covid-19 related stress exacerbates common physical and mental pathologies and affects treatment. *Experimental and therapeutic medicine*, 20(1):159–162, 2020.

[115] Felipe Reis Valente, Marcelo de Paiva Guimarães, Elder José Reioli Cirilo, and Diego Roberto Colombo Dias. A multi-agent body tracking application framework applied to physical and neurofunctional rehabilitation. In *International Conference on Computational Science and Its Applications*, pages 459–472. Springer, 2022.

[116] Thomas Vilarinho, Babak Farshchian, Daniel Gloppestad Bajer, Ole Halvor Dahl, Iver Egge, Sondre Steinsland Hegdal, Andreas Lønes, Johan N Slettevold, and Sam Mathias Weggersen. A combined smartphone and smartwatch fall detection system. In *2015 IEEE*

*international conference on computer and information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive intelligence and computing*, pages 1443–1448. IEEE, 2015.

[117] SL Wagner, C Koehn, MI White, HG Harder, IZ Schultz, Kelly Williams-Whitt, O Warje, CE Dionne, M Koehoorn, R Pasca, et al. Mental health interventions in the workplace and work outcomes: a best-evidence synthesis of systematic reviews. *Int J Occup Environ Med (The IJOEM)*, 7(1 January):607–1, 2016.

[118] Chaoqi Wang, Guodong Zhang, Rui Luo, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations (ICLR)*, 2020.

[119] Jiawei Wang, Jian Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities. 2024.

[120] A. T. M. Wasylewicz and M. W. M. Jaspers. Clinical decision support systems. In *Health Informatics: eHealth*. NIH / NCBI Bookshelf, 2018.

[121] Charles Westphal, Stephen Hailes, and Mirco Musolesi. Mutual information preserving neural network pruning. 2025.

[122] M. Wicker, X. Huang, and M. Kwiatkowska. Feature-guided black-box safety testing of deep neural networks. In *TACAS*, 2018.

[123] Daryl J Wile, Ranjit Ranawaya, and Zelma HT Kiss. Smart watch accelerometry for analysis and diagnosis of tremor. *Journal of neuroscience methods*, 230:1–4, 2014.

[124] Lianne J Woodward and David M Fergusson. Life course outcomes of young people with anxiety disorders in adolescence. *Journal of the American Academy of Child & Adolescent Psychiatry*, 40(9):1086–1093, 2001.

[125] Daniel J Wu, Andrew C Yang, and Vinay U Prabhu. Afro-mnist: Synthetic generation of mnist-style datasets for low-resource languages, 2020.

[126] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[127] Zachary Young and Richard Steele. Empirical evaluation of performance degradation of machine learning-based predictive models–a case study in healthcare information systems. *International Journal of Information Management Data Insights*, 2(1):100070, 2022.

[128] Y. Zhang, L. Chen, and D. Ghosh. Diagnostic prediction with sequence-of-sets representation learning for clinical events. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, 2020.

[129] Zhongjin Zhao, Dashun Que, Dinglin Jiang, and Guozheng Shi. Design of intelligent monitoring system for knee joint. In *Journal of Physics: Conference Series*, volume 1449, page 012106. IOP Publishing, 2020.

[130] Zijian Zhao, Fangchen Huang, Yao Xu, et al. Eagle: Efficient training-free router for multi-llm inference. NeurIPS ML for Systems Workshop, 2024.

[131] Aojun Zhou, Yukun Ma, Jianhao Zhu, Cong Liu, Yanzhi Sun, Bo Yuan, Shuai Wang, Xue Lin Chen, Yanzhi Guo, and Weiyao Chen. Learning n:m fine-grained structured sparse neural networks from scratch. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[132] Yanqi Zhou, Tao Liang, et al. Mixture-of-experts with expert choice routing. Google Research Blog, 2022. NeurIPS 2022 Spotlight.

[133] Wolfgang Ziegler. A landscape analysis of standardisation in the field of artificial intelligence. *Journal of ICT Standardization*, 8(2):151–184, 2020.

[134] Mark Zimmerman, Jacob Martin, Heather Clark, Patrick McGonigal, Lauren Harris, and Carolina Guzman Holst. Measuring anxiety in depressed patients: A comparison of the hamilton anxiety rating scale and the dsm-5 anxious distress specifier interview. *Journal of psychiatric research*, 93:59–63, 2017.