UNIVERSITY OF CALIFORNIA,
IRVINE

Robust, Personalized, and Context-aware Affect Monitoring in Daily-life

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Electrical and Computer Engineering

by

Seyed Amir Hossein Aqajari

Dissertation Committee:
Professor Amir M. Rahmani, Chair
Professor Nikil Dutt
Professor Fadi Kurdahi

2024

DEDICATION

I dedicate this work to my parents, who have always supported me both financially and morally, providing their fullest and truest attention, helping me accomplish my work with confidence.

To my brother and two sisters, who have always stood by my side unconditionally, sharing in my joys and sorrows.

I am deeply thankful for having you all in my life and hope to see you soon!

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

ACKNOWLEDGMENTS

First and foremost, I would like to thank to my advisor, Professor Amir M. Rahmani, for the continuous support of my research during the past two years. Without your assistance and dedicated involvement in every step throughout the process, this work would have never been accomplished.

Beside my advisor, I would also like to express my sincere gratitude to my committee member Professor Nikil Dutt not only for assigning their valuable time to review my work but also for his insightful feedback that pushed my work to a higher level.

My appreciation also extends to my committee member Professor Fadi Kurdahi, for assigning their priceless time to review my work.

I am thankful to Professor Salma Elmalaki and Professor Quoc-Viet Dang, and I would like to thank Professor Brian Demsky for his support and help during my first year of Ph.D.

I am also grateful to my friends and fellow lab-mates, who have always been there for me throughout this journey.

I would also like to thank the Academy of Finland and the US National Science Foundation (NSF) to partially funding this work (through the SLIM project under the grants (316810, 316811) and the UNITE project under the grant SCC CNS-1831918 respectively).

VITA

Seyed Amir Hossein Aqajari

EDUCATION

Doctor of Philosophy in Electrical and Computer Engineering **2024**
University of California Irvine *Irvine, CA*

Master of Science in Computer Science and Engineering **2021**
University of California Irvine *Irvine, CA*

Bachelor of Science in Electrical Engineering **2018**
Sharif University of Technology *Tehran, Iran*

WORK EXPERIENCE

Firmware Engineer **2021–NOW**
ASR Microelectronics International *Irvine, California*

Machine Learning Engineer **2022–2023**
Wise IOT Solutions *San Diego, California*

RESEARCH EXPERIENCE

Graduate Research Assistant **2018–2024**
University of California, Irvine *Irvine, California*

TEACHING EXPERIENCE

Teaching Assistant **2019–2023**
University of California, Irvine *Irvine, California*

REFEREED JOURNAL PUBLICATIONS

Enhancing Performance and User Engagement in Everyday Stress Monitoring: An Online Context-Aware Active RL Approach [1] **Under Review**
Seyed Amir Hossein Aqajari, Ziyu Wang, Ali Tazarv, Sina Labbaf, Salar Jafarlou, Nikil Dutt, Marco Levorato, Amir M. Rahmani, ACM Health

Context-Aware Stress Monitoring using Wearable and Mobile Technologies in Everyday Settings [2] **Under Review**
Seyed Amir Hossein Aqajari, Sina Labbaf, Phuc Hoang Tran, Brenda Nguyen, Milad Asgari Mehrabadi, Marco Levorato, Nikil Dutt, Amir M Rahmani, Smart Health

An Accurate Non-accelerometer-based PPG Motion Artifact Removal Technique using CycleGAN [3] **2023**
Amir Hosein Afandizadeh Zargari, Seyed Amir Hossein Aqajari, Hadi Khodabandeh, Amir Rahmani, Fadi Kurdahi, ACM Transactions on Computing for Healthcare

Pain Assessment Tool With Electrodermal Activity for Postoperative Patients: Method Validation Study [4] **2021**
Seyed Amir Hossein Aqajari, Rui Cao, Emad Kasaeyan Naeini, Michael-David Calderon, Kai Zheng, Nikil Dutt, Pasi Liljeberg, Sanna Salanterä, Ariana M Nelson, Amir M Rahmani, JMIR mHealth and uHealth

Sleep Quality Prediction During the Menstrual Cycle based on Daily Sleep Diary Reports [5] **2021**
Negin Sattari, Milad Asgari Mehrabadi, Seyed Amir Hossein Aqajari, Jing Zhang, Katharine Simon, Elisabet Alzueta, Teji Dulai, Massimiliano de Zambotti, Fiona Baker, Amir Rahmani, Sara Mednick, Sleep

Home-visit intervention to reduce stress of underserved family caregivers for persons with dementia [6] **2021**
Jung-Ah Lee, Seyed Amir Hossein Aqajari, Eunae Ju, Priscilla Kehoe, Lisa Gibbs, Amir Rahmani, Innovation in Aging

REFEREED CONFERENCE PUBLICATIONS

Controlling the Latent Space of GANs through Reinforcement Learning: A Case Study on Task-based Image-to-Image Translation [7] **2024**

Mahyar Abbasian, Taha Rajabzadeh, Ahmadreza Moradipari, Seyed Amir Hossein Aqajari, Hongsheng Lu, Amir M Rahmani, Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing

Novel blood pressure waveform reconstruction from photoplethysmography using cycle generative adversarial networks [8] **2022**

MA. Mehrabadi, SAH. Aqajari, AHA. Zargari, N. Dutt, AM. Rahmani, IEEE Engineering in Medicine & Biology Society (EMBC)

pyEDA: An Open-Source Python Toolkit for Preprocessing and Feature Extraction of Electrodermal Activity [9] **2021**

Seyed Amir Hossein Aqajari, Emad Kasaeyan Naeini, Milad Asgari Mehrabadi, Sina Labbaf, Nikil Dutt, Amir M Rahmani, The 12th International Conference on Ambient Systems, Networks and Technologies (ANT)

An End-to-End and Accurate PPG-based Respiratory Rate Estimation Approach Using Cycle Generative Adversarial Networks [10] **2021**

Seyed Amir Hossein Aqajari, Rui Cao, Amir Hosein Afandizadeh Zargari, Amir M Rahmani, IEEE Engineering in Medicine & Biology Society (EMBC)

Detection of covid-19 using heart rate and blood pressure: Lessons learned from patients with ards [11] **2021**

Milad Asgari Mehrabadi, Seyed Amir Hossein Aqajari, Iman Azimi, Charles A Downs, Nikil Dutt, Amir M Rahmani, IEEE Engineering in Medicine & Biology Society (EMBC)

Objective pain assessment using wrist-based ppg signals: A respiratory rate based method [12] **2021**

Rui Cao, Seyed Amir Hossein Aqajari, Emad Kasaeyan Naeini, Amir M Rahmani, IEEE Engineering in Medicine & Biology Society (EMBC)

US PATENTS

Pain assessment method and apparatus for patients unable to self-report pain [13] **2022**

Amir M. Rahmani, Nikil Dutt, Kai Zheng, Ariana Nelson, Pasi Liljeberg, Sanna Salantera, Mingzhe Jiang, Anzanpour Arman, Elise Syrjala, Riitta Mieronkoski, Emad Kasaeyan Naeini, Ajan Subramanian, Seyed Amir Hossein Aqajari, Rui Cao, Geng Yang, US 17669984

SOFTWARES

Python, C/C++, Java, TypeScript, TensorFlow, pyTorch, CUDA, Git

ABSTRACT OF THE DISSERTATION

Robust, Personalized, and Context-aware Affect Monitoring in Daily-life

By

Seyed Amir Hossein Aqajari

Doctor of Philosophy in Electrical and Computer Engineering

University of California, Irvine, 2024

Professor Amir M. Rahmani, Chair

Affect, encompassing both emotions and moods, plays a vital role in shaping human experience and behavior. Emotions are intense, short-lived reactions to specific stimuli, while moods are prolonged, less intense affective states influencing overall perception and behavior. Understanding and monitoring affect is crucial for enhancing mental health and emotional well-being. This thesis explores the intersection of affect, mental health, and the autonomic nervous system (ANS), emphasizing the significance of physiological signals in capturing involuntary physiological processes.

With advancements in wearable IoT devices, continuous monitoring of physiological responses such as photoplethysmography (PPG) has become feasible. This research addresses the limitations of traditional self-reporting methods by proposing a comprehensive system architecture for affect monitoring in daily life. The system integrates three essential components: Context-Awareness, Personalization, and Robustness. Context-awareness enables the system to adapt its monitoring and decision-making processes based on the dynamic environment and user-specific conditions, ensuring that the responses are relevant and timely. Personalization ensures that the monitoring is attuned to individual differences through machine learning algorithms. Robustness guarantees reliable performance against daily-life noise and motion artifacts.

First, we introduce pyEDA, a powerful tool that harnesses deep learning techniques to extract more relevant features from physiological signals like PPG and EDA, with a focus on stress-related indicators. We detail the methodology and implementation of pyEDA, emphasizing its ability to handle large datasets and extract meaningful physiological features critical for effective affective monitoring.

Then we focus on the development of a PPG motion artifact removal module. This module is crucial for ensuring robust performance in everyday scenarios where motion artifacts can significantly distort physiological data. We discuss various signal processing techniques employed to mitigate these artifacts, ensuring the reliability of the collected data.

Next, we present a novel module for extracting respiration rate from PPG signals. Incorporating respiratory rate as an additional modality enriches the affective monitoring system. We elaborate on the algorithms and validation processes used to accurately derive respiration rate, demonstrating its significance in understanding physiological responses.

After that we describe the overall closed-loop system architecture, which integrates physiological data, context data, and stress labels. This integration is crucial for developing a context-aware affect monitoring system tailored for daily life. We provide an in-depth explanation of the system components and their interactions, showcasing how this context-aware closed-loop design enhances the accuracy and relevance of affective monitoring.

Finally, we apply a context-aware active reinforcement learning approach to the proposed closed-loop system. This approach aims to enhance both performance and user engagement by incorporating personalization, dynamically adapting to the user's context, preferences, and feedback. We discuss the reinforcement learning framework, the experimental setup, and the results, demonstrating the effectiveness of this approach in real-world applications.

This work aims to develop a context-aware, robust, and personalized affect monitoring system for real-world applications, advancing the field of mental health and affective computing.

Chapter 1

Introduction

Affect is a broad term used in psychology to describe the experience of feeling or emotion, encompassing both the immediate, intense responses known as emotions and the more prolonged, diffuse states referred to as moods. Emotions are typically short-lived and are often a reaction to a specific stimulus, manifesting as intense feelings such as joy, anger, or fear [14]. They are usually accompanied by distinct physiological responses and are often visible through facial expressions and body language. In contrast, moods are less intense than emotions and are not necessarily tied to a specific event or trigger. Instead, they represent a more sustained affective state that can influence a person's overall perception and behavior over a longer period [15]. Both emotions and moods play a critical role in shaping human experience and behavior, serving as essential components of affective processing [16].

Recent data suggests that approximately 20% of U.S. adults experience mental health issues annually, with a higher incidence among young adults compared to older age groups as of 2019 [17]. Emotional well-being is significantly influenced by affect, which encompasses both the subjective feeling and physiological response to emotions, whether positive or negative [16]. Assessing affect helps identify triggers of emotional states, allowing for personalized therapeutic or medicinal interventions that enhance cognitive and behavioral responses, thereby reducing emotional disturbances [15]. Understanding the intricacies of affective responses is crucial in developing effective strategies for mental health treatment and improving overall psychological resilience.

The autonomic nervous system (ANS) is a crucial component of the human body that regulates involuntary physiological processes, including heart rate, blood pressure, respiration, and digestion [18]. ANS activity plays a direct role in mental health stressors, influencing how the body responds to stress and anxiety [19]. These involuntary physiological processes can be captured using various physiological signals, providing insights into an individual's autonomic function. With the rapid advancement of wearable IoT devices, collecting physiological signals such as photoplethysmography (PPG) and electrodermal activity (EDA) has become increasingly feasible in daily-life settings using smartwatches or wristbands [20–22]. Recent advancements have leveraged differential private federated transfer learning for stress detection in everyday settings, developed energy-efficient adaptive learning algorithms for smart wearables, and utilized blockchain technology for secure data management and anomaly detection in smart healthcare systems [23–25]. Building on the foundational research in circuit design for biomedical applications, existing works in literature extend previous efforts by developing efficient, low-power circuits that integrate seamlessly with machine learning techniques, advancing the capabilities of healthcare technology [26–28]. Recent machine learning frameworks and simulation models have shown potential in reducing execution time and improving resource allocation, which could enhance the efficiency and performance of affect monitoring systems [29, 30]. There is a rich body of literature on the theoretical and experimental analysis of machine learning algorithms in medical, robotics, transportation, and smart grid applications [31–50]. This technological progress enables continuous monitoring of physiological responses, offering valuable data for understanding and managing mental health conditions.

Traditional approaches for monitoring affect heavily rely on patient self-reporting or logging through diaries, which often require assistance from a therapist or provider [51]. Recent research studies have started to utilize contextual data and physiological responses collected from wearable devices to predict daily-life affect, marking a shift towards more automated and less intrusive methods [52, 53]. However, many of these studies are still conducted in laboratory settings, limiting their applicability in real-world environments. The models developed in these studies often cannot be used in daily-life settings as they are not robust and are prone to noise [54].

Despite these challenges, some new research studies have begun to build affect monitoring systems that can be used in daily-life situations [55]. These studies often ask for self-reported assessments, which can result in higher rates of missing data due to the unsatisfactory experience of users. Additionally, while some studies only utilize smartphone contextual data, others integrate physiological data from wearables to predict daily-life affect [56]. A significant limitation of most of these works is the lack of personalization, which is an important factor in affect monitoring as individual differences can greatly influence affective responses and their manifestations [57].

Table 2.1 provides a comprehensive summary of recent advancements in the literature focused on detecting affect in daily-life settings. This table consolidates various methodologies and technologies employed in real-world affect monitoring, highlighting the diversity and evolution of approaches in this emerging field.

Table 1.1: Summary of existing works in literature.

Related Works	**Multi-Modality**	**Context-Awareness**	**Robustness**	**Personalized modeling**
Sano et. al. [58]	✓	✓	✓	✗
HENKER et. al. [59]	✗	✓	✗	✗
Kim et. al. [60]	✗	✗	✓	✗
Fahrenberg et. al. [61]	✗	✗	✓	✗
Lai et. al. [62]	✗	✓	✓	✓
Costaldo et. al. [63]	✗	✗	✓	✗
Shi et. al. [64]	✗	✗	✓	✓
This Work	✓	✓	✓	✓

In this work, we propose a comprehensive system architecture that integrates three essential components while maintaining multi-modality (Fig. 1.1):

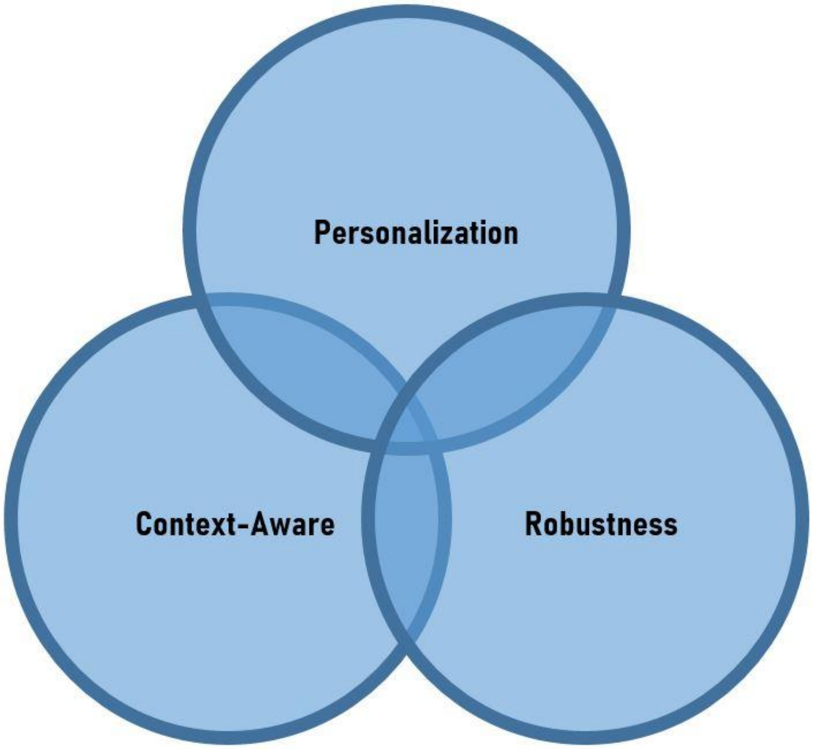


Figure 1.1: Main components of our proposed system.

Context-Awareness: The system leverages context-aware algorithms to dynamically adjust its monitoring based on the user's environment and activities. By integrating contextual information such as location, time of day, and user activity, the system fine-tunes its monitoring processes, ensuring that affective states are interpreted accurately and in relation to the surrounding context.

Personalization: The system uses personalized models tailored to each user, ensuring that affective monitoring is accurately attuned to individual differences. Machine learning algorithms adapt to unique patterns and preferences, enhancing the relevance and effectiveness of the monitoring.

Robustness: Designed to be resilient against daily-life noise and motion artifacts, the architecture employs advanced signal processing techniques to ensure reliable and accurate monitoring in real-world conditions. This robust design maintains functionality and accuracy during various activities and settings.

The rest of this dissertation is organized as follows. In Chapter 2, we introduce pyEDA, a tool that leverages deep learning to extract more relevant features from physiological signals such as PPG and EDA, with a focus on stress-related indicators. In Chapter 3, we propose a PPG motion artifact removal module to ensure robust performance in everyday scenarios. In Chapter 4, we present a module for extracting respiration rate from PPG, adding respiratory rate as an extra modality to our system. In Chapter 5, we describe our context-aware closed-loop system architecture, which integrates physiological data, contextual information, and stress labels, crucial for understanding affect in daily life. Finally, in Chapter 6, we apply a context-aware active reinforcement learning approach to our proposed closed-loop system which aims to enhance performance and user engagement by incorporating personalization, dynamically adapting to the user's context, preferences, and feedback.

Chapter 2

Leveraging Deep Learning Techniques to Extract more Relevant Stress-Related Features

Physiological responses are automatic reactions that trigger physical changes in response to stimuli such as stress, emotion, or pain. Examples of these changes include variations in heart rate, respiration, perspiration, and pupil dilation. Key physiological signals, such as Photoplethysmogram (PPG), Electrodermal Activity (EDA), and respiratory signals, serve as major indicators for capturing these responses. However, analyzing these signals is complex. Traditionally, statistical features are extracted from these signals to detect different stimuli in human subjects. For instance, in the case of EDA, features like the number of peaks and maximum peak amplitude are identified to monitor stimuli.

With the rapid advancement of AI and machine learning algorithms, various methods have been developed to automatically extract features from these signals using deep learning techniques. In many cases, these methods outperform traditional feature extraction techniques in terms of prediction accuracy. An open-source tool that provides automatic feature extraction from physiological signals can significantly facilitate research studies and enhance the performance of stimulus predictions in closed-loop systems.

In this chapter, we introduce pyEDA, an open-source Python toolkit designed for signal preprocessing and both statistical and automatic feature extraction of EDA signals. Using EDA as a case study, the automatic feature extraction module of pyEDA can also be applied to other physiological signals such as PPG or respiratory signals. To our knowledge, this is the first effort to develop a versatile and generic tool capable of extracting numerous automatic features from physiological signals. We evaluate our toolkit using various machine learning algorithms on the Wearable Stress and Affect Detection (WESAD) dataset. Our results demonstrate higher validation accuracy for stress detection tasks using features automatically extracted by pyEDA compared to traditional methods.

2.1 Introduction

The human body generates a number of physiological signals that can be used to extract valuable information which measures the functional state of various physiological systems [65]. Many studies have validated the effect of different stimuli on the functioning of the physiological systems in the human body, such as emotion, stress, and sleep [66–70].

Table 2.1: Comparing the existing toolkits compared with pyEDA for preprocessing and feature extraction of EDA signals.

Related Works	Statistical Features	Automatically Extracted Features	Application-agnostic	Python
pyEDA	✓	✓	✓	✓
TEAP [71]	✓	✗	✗	✗
PhysioLab [72]	✓	✗	✗	✗
ANSLAB [73]	✓	✗	✗	✗
NeuroKit [74]	✓	✗	✗	✓
Pysiology [75]	✓	✗	✗	✓

The Electrodermal Activity (EDA), also known as galvanic skin response (GSR), is one of these physiological signals widely used in biomedical and digital health research to detect certain stimuli in human subjects. EDA refers to changes in sweat gland activity, which reflects the intensity of the individual's emotional state – or emotional arousal [76].

In prior studies, numerous prediction models are constructed to predict a stimulus based on raw EDA signals [77, 78]. The EDA signals collected from commercially available devices are usually raw and have motion artifacts that are common in natural, uncontrolled settings that involve body gestures and movements. Therefore, the raw EDA itself cannot be used in these prediction models directly. First, several signal processing steps are needed to remove noise and extract the clean signal. Next, a variety of features are extracted from the clean EDA signal. These features can then be fed to machine learning algorithms to build prediction models detecting different types of stimuli.

Traditional methods extract statistical features such as the number of peaks, max peak amplitude, average, standard deviation, etc., from the EDA signal for prediction models. However, with rapid development of AI and machine learning algorithms, various automatic methods are implemented to extract automatic features from the signal using neural networks [79, 80]. In many situations, these methods outperform the traditional methods' performance in prediction accuracy. Needless to say, an open-source tool providing automatic and statistical set of features extracted from EDA signal can significantly facilitate the research studies in EDA signal processing. Its worth mentioning that these neural network techniques are not limited only to automatic feature extraction from the signals. Zargari et al. used combination of convolutional neural network and recurrent neural network to accurately track in-mouth nutrient sensors position [81]. Mehrabadi et al. used convolutional neural network to detect COVID-19 in patients with ARDS [82]. Ashrafiamiri et al. used deep neural networks to secure autonomous driving [83].

The existing open-source tools for EDA signal processing only focus on the statistical features and do not take into account the embedding features extracted using automatic methods

(Table 2.1). Soleymani *et al.* [71] develop a toolbox to extract features from different signals including EDA. The author build classifiers solely using statistical features, including: amplitude and number of peaks, mean, and standard deviation of the signal. Furthermore, there are toolboxes that provide integrated software. PhysioLab [72] and ANSLAB [73] are open-source tools for EDA analysis, which are implemented in Matlab. These tools aggregate the information extracted from different signals including EDA. However, the feature extraction module is limited to non-automated statistical features. Besides, researchers have implemented toolboxes for EDA analysis in Python [74, 75]. These toolboxes also have the same limitation since they also only consider statistical features including: number of peaks, amplitude, rise time and decay time. It should be noted that all of these existing toolboxes tailor the feature extraction phase to their target application (e.g., emotion recognition, pain assessment, etc.) making them application-dependent to a certain degree.

In this chapter, we present pyEDA [84], an open-source tool in Python with the ability to process EDA signals and extract statistical and automatic features from them. To the best of our knowledge, this is the first work presenting a user-friendly open-source Python tool which can be used to extract any number of automatic features of EDA signals without the need to have a background in artificial neural networks and auto-encoders. Depending on the application, different statistical features might be conceded for acceptable prediction accuracy to a stimulus [85]. In other words, certain features to a stimulus may not be captured using statistical methods. For these reasons, we present a tool to automatically extract any number of automatic features highly correlated to any type of stimuli. Providing such a versatile and generic tool to extract automatic features of the EDA signal is valuable for the health science and technology community. We also demonstrate the efficacy of our tool by using the Wearable Stress and Affect Detection (WESAD) dataset [86] and a set of machine learning algorithms to evaluate both statistical and automatic features extracted using pyEDA. Specifically, this work makes the following key contributions:

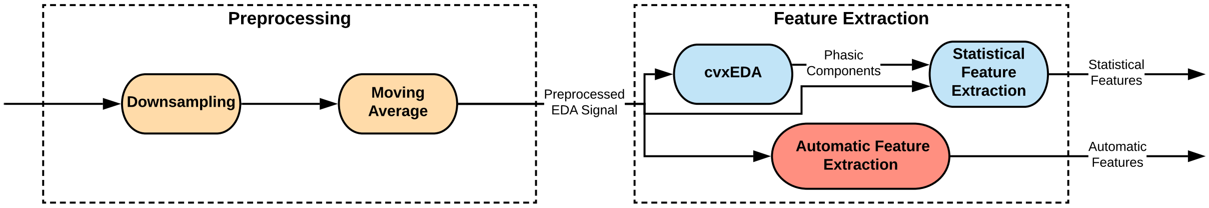


Figure 2.1: Proposed Processing Pipeline Architecture for pyEDA.

- Develop an open-source toolkit to extract any number of features of EDA signals.
- Build a scalable architecture to work with EDA signals with any arbitrary length.
- Enable the toolkit to perform generic automatic feature extraction.
- Evaluate the extracted features using a publicly available dataset on stress assessment.

The rest of this chapter is organized as follows. Section 2.2 briefly outlines the EDA background. Our proposed processing pipeline architecture is presented in Section 2.3. In Section 2.4 we evaluate our result using the WESAD data set. Finally, Section 2.5 concludes the chapter.

2.2 EDA Background

To better understand how EDA is captured, it is helpful to study the physiological characteristics of the skin described in [76]. Sweat glands are small tubular structures of the skin producing sweat. Our body has approximately three million sweat glands having different density across the body. Sweat glands can be found in large numbers on the soles of the feet, the palms and fingers, and on the forehead and cheeks. They produce moisture through pores towards the surface of the skin, whenever they are triggered. When the balance of positive and negative ions in this secreted fluid changes, the electrical current flows more readily. This results in decreased skin resistance, or in other words, increased skin conductance. Electrodermal Activity (EDA) is a term used for this change. EDA is also known as Galvanic Skin Response (GSR), Skin Conductance (SC), Electrodermal Response (EDR), and Psychogalvanic Reflex (PGR).

Although one of the main purposes of sweating is thermoregulation, sweating is also triggered whenever a person is exposed to a stimulus such as emotionally loaded images. This type of sweating is called emotional sweating. Sweat secretion, which reflects the changes in arousal, is driven unconsciously by the automatic nervous system (ANS) in order to meet behavioral requests. A number of commercially available devices (e.g., RespiBAN professional, Empatica E4, Fitbit Sense, and Shimmer3 GSR+) can be used to collect EDA signals.

According to [76], EDA signals consist of two main components: Skin Conductance Level (SCL) and Skin Conductance Response (SCR). The SCL changes slightly on a time scale of tens of seconds to minutes. Depending on hydration, skin dryness or automatic regulation of an individual respondent, the rising and declining SCL is continuously changing. SCL, which is also called the tonic level of EDA signal, can differ significantly across different individuals. Due to this, the actual tonic level on its own is not completely informative. SCR, which is also known as the phasic component of EDA, rides on top of the tonic changes and shows much faster alterations. Variations in the phasic component of a EDA signal are visible as EDA bursts or EDA peaks. The phasic component is sensitive to specific emotionally arousing stimulus events (event-related SCRs, ER-SCRs). These bursts can occur between 1-5 seconds after the onset of emotional stimuli. Quite the opposite, non-specific skin conductance responses (NS-SCRs) are not a consequence of any eliciting stimulus. These responses happen at a rate of 1-3 per minute spontaneously. There is a need for a flexible and usable processing toolchain that can be used to efficiently analyze these EDA signals. In the following, we describe our pyEDA processing pipeline architecture that is designed to meet these needs.

2.3 Proposed Processing Pipeline Architecture

Figure 2.1 shows the proposed processing pipeline architecture for pyEDA to analyze the EDA data. There are two different stages in this pipeline: The pre-processing stage and the feature extraction stage. As shown in this figure, the pre-processing stage consists of two different modules. In the pre-processing stage, the signals are cleaned and prepared for the feature extraction stage. Then, the feature extraction stage uses two different procedures to extract the features from the pre-processed data. We use traditional manual statistical feature extraction as well as automatic feature extraction methods in our proposed architecture. In the following, we explain each stage of the pipeline in detail.

2.3.1 Pre-processing

In this stage, we use down-sampling and moving averaging to pre-process the data. At the end of this stage, a pre-processed EDA signal is ready and accessible for further analysis and feature extraction.

The EDA data is usually sampled at much higher frequency than needed. Therefore, down-sampling is done to reduce memory footprint and processing time of the data with a negligible risk of losing important information in the signal. In the pre-processing stage, the raw EDA data is down-sampled to the lower sampling rate. Based on the studies conducted in [76], the EDA data can safely be down-sampled to 20 Hz or even less if the data originally was collected at 128 Hz.

A raw EDA signal varies before or after a peak. This is due to individual differences in the tonic component of EDA or due to the noise caused by movements or respiration artifacts [76]. After down-sampling the data, a moving average across a 1-second window is first used to smooth the data and reduce artifacts such as body gestures and movements, which are common in everyday settings.

2.3.2 Statistical Feature Extraction

The number of peaks, the mean of EDA, and the max peak amplitude are three statistical features extracted in our pipeline. Calculating the mean of EDA is straightforward. To calculate the other two features, we need to extract the EDA peaks that are induced by eliciting stimulus. A number of signal processing steps are required to derive EDA peaks that are a consequence of eliciting stimulus [76]:

1. The phasic component is extracted from the pre-processed EDA signal.
2. A low-pass Butterworth filter is applied on the phasic data to remove line noise. A cutoff frequency of 5 Hz divided by sampling rate is typically used.
3. Onsets and offsets are identified from the phasic data.
4. The maximum amplitude value within each onset-offset is considered as a peak if and only if its difference with the amplitude value at onset is higher than the threshold, which typically is 0.005 μS.

One of the major tasks in analyzing the EDA signal is to correctly extract the phasic component of the signal from the original signal. In this chapter, we use the cvxEDA algorithm to decompose the original signal into a sparse phasic component and a smooth tonic component. The cvxEDA algorithm is a novel algorithm presented in [87] for the EDA analysis based on maximum a posteriori probability, convex optimization, and sparsity. This algorithm has a desirable capability of properly describing the activity of the autonomic nervous system in response to affective stimulation. This model describes the recorded EDA as the sum of three terms: the phasic component, the tonic component, and an additive white Gaussian

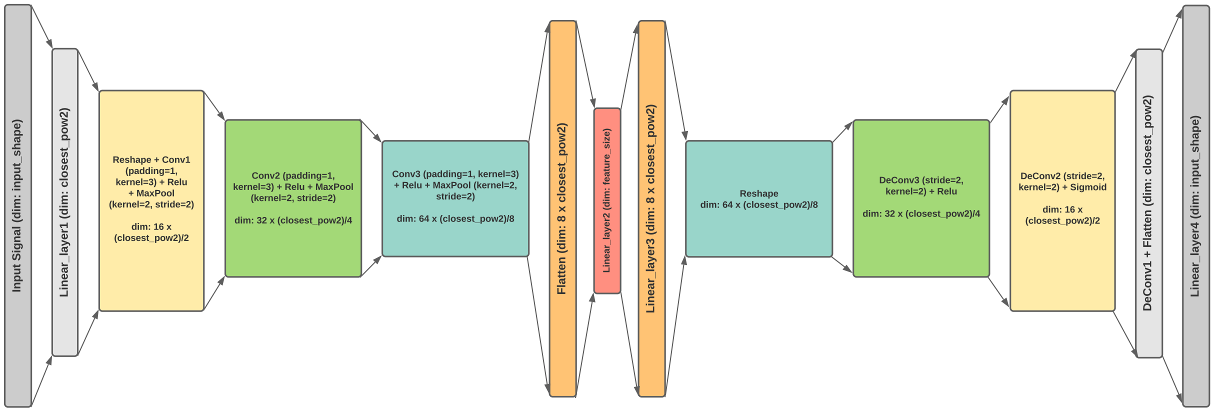

Input Signal (dim: input_shape)
Linear_layer1 (dim: closest_pow2)
Reshape + Conv1 (padding=1, kernel=3) + Relu + MaxPool (kernel=2, stride=2)
dim: 16 x (closest_pow2)/2
Conv2 (padding=1, kernel=3) + Relu + MaxPool (kernel=2, stride=2)
dim: 32 x (closest_pow2)/4
Conv3 (padding=1, kernel=3) + Relu + MaxPool (kernel=2, stride=2)
dim: 64 x (closest_pow2)/8
Flatten (dim: 8 x closest_pow2)
Linear_layer2 (dim: feature_size)
Linear_layer3 (dim: 8 x closest_pow2)
Reshape
dim: 64 x (closest_pow2)/8
DeConv3 (stride=2, kernel=2) + Relu
dim: 32 x (closest_pow2)/4
DeConv2 (stride=2, kernel=2) + Sigmoid
dim: 16 x (closest_pow2)/2
DeConv1 + Flatten (dim: closest_pow2)
Linear_layer4 (dim: input_shape)

Figure 2.2: The architecture of the proposed autoencoder.

noise term incorporating model prediction errors, measurement errors, and artifacts. We use this algorithm to extract the phasic component of the pre-processed EDA signal for further analysis.

To calculate the number of peaks corresponding to events related to SCRs, we need to find the maximum amplitude value within each onset-offset window in the original EDA signal [76]. Onsets are all the points in which the phasic component of the signal crosses above the onset threshold, which is typically 0.01 μS. To find the corresponding offset of the computed onset, we check for the point in which the phasic component of the EDA signal crosses below the offset threshold, which is typically 0 μS. For each window, the time difference between its onset and offset needs to be above the duration threshold of 1s. Any peaks before this duration threshold are considered as nonspecific skin conductance responses and does not need to be extracted.

The clean EDA data, the peaks, and the tonic and phasic components of signal can be used to easily extract extra statistical features if needed.

2.3.3 Automatic Feature Extraction

Feature extraction becomes increasingly important when the data is high dimensional. There have been several studies which attempt to create classification models based on statistical features extracted from physiological signals such as EDA. However, there is no study suggesting which specific set of statistical features is informative for any type of EDA analytics regardless of the final application (e.g., assessing stress, emotion, pain, risk of seizure, etc.). Therefore, using automatic feature extraction and selection to find the most important set of features for any type of application can be significantly important.

An autoencoder is a type of unsupervised artificial neural network which is used to learn efficient data coding [88, 89]. The aim of an autoencoder is to efficiently learn how to compress and encode the data to reduce the dimensional representation. This lower dimensional representation can be regarded as an abstract set of features of the original high dimensional data. An autoencoder consists of two different parts: the encoder and the decoder. These two parts can be defined as two different functions as follows:

$$\phi : \mathcal{X} \rightarrow \mathcal{Y} \tag{2.1}$$

$$\psi : \mathcal{Y} \rightarrow \mathcal{X}' \tag{2.2}$$

Given one hidden layer for encoder and decoder parts in the simplest case: Function (2.1) can be defined as $\boldsymbol{y} = \sigma(\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b})$. σ is an element-wised activation function such as sigmoid. $\boldsymbol{W}$ is a weight matrix and $\boldsymbol{b}$ is a bias vector of the encoder part. $\boldsymbol{x}$ is $\in \mathcal{X}$ and $\boldsymbol{y}$ is $\in \mathcal{Y}$. Function (2.2) can be defined as $\boldsymbol{x'} = \sigma(\boldsymbol{W'}\boldsymbol{y}+\boldsymbol{b'})$. σ is an element-wised activation function such as sigmoid. $\boldsymbol{W'}$ is a weight matrix and $\boldsymbol{b'}$ is a bias vector of the decoder part. $\boldsymbol{x'}$ is $\in \mathcal{X}'$ and $\boldsymbol{y}$ is $\in \mathcal{Y}$.

Autoencoders are trained to minimize the following reconstruction loss:

$$loss(\boldsymbol{x}, \boldsymbol{x'}) = \|\boldsymbol{x} - \boldsymbol{x'}\|^2 \tag{2.3}$$

Weights and biases are initialized randomly, and then iteratively updated based on the reconstruction loss computed in (2.3) during training through Backpropagation. The image $\boldsymbol{y}$ represents a latent code or latent representation of the input $\boldsymbol{x}$. It can be directly used as the set of features of input $\boldsymbol{x}$ for classification.

Figure 2.2 shows the architecture of the autoencoder implemented in our pipeline for automatic feature extraction. First, a linear layer is used to downsample the input EDA signal to the closest power of two length. This is done to make our tool scalable; thus, the input EDA data with any arbitrary length can be entered to our proposed model. Based on this figure, the encoder consists of three 1d convolutional layers. The output of the encoder is flattened and then downsampled to the latent code size. The output of this linear layer is the lower representation of the input signal which is extracted as the set of automatic features. In the decoder part there are three 1d deconvolutional layers to reconstruct the input signal from the latent code. At the end of this part, the data is flattened and a linear layer is used for reconstruction. The Rectified Linear Units (ReLU) are used here as activation functions in the network. It is worth mentioning that the number of extracted features are fed to our tool as an input parameter before training the model (in this work the number of features is 64). We use PyTorch library to implement our autoencoder architecture. PyTorch is an open source machine learning library based on the Torch library, used for applications such as computer vision and natural language processing [90].

2.4 Results

2.4.1 Machine Learning based Classification

To demonstrate the efficacy of the pyEDA toolkit, we use it to build machine learning based stress and affect models using the WESAD dataset and evaluate the performance of our extracted features. We use four different machine learning algorithms: (1) K-nearest-neighbor (kNN) with k between 1 to 10, (2) Naïve Bayes Gaussian classifier, (3) Random Forest with depth between 1 to 10, and (4) support vector machine (SVM). The kNN method uses k number of nearest data-points and predicts the result based on a majority vote [91]. The Naïve Bayes Gaussian classifier predicts the result based on the probabilities of each feature's Gaussian distribution [92]. The SVM tries to find the best hyper-plane to divide the data points into different classes [93]. The Random Forest classifier fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting [94]. We use the scikit-learn software for classification and prediction. The scikit-learn is an open-source machine learning library for the Python programming language [95].

2.4.2 WESAD Dataset

Wearable Stress and Affect Detection (WESAD) is a publicly available multimodal dataset for stress and affect detection [86]. In this data set, physiological and motion data are recorded from Empatica E4 and RaspbiAN professional devices from 15 subjects during a lab study.

The goal of this dataset is to elicit three affective states (neutral, stress, amusement) in the participants. There are two different versions of the study protocol in this dataset. These

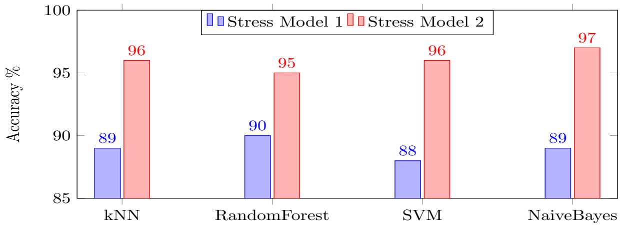


Figure 2.3: Validation accuracy on two different stress models using different machine learning algorithms.

protocols consist of six different tasks labeled as Baseline, Amuesment, Medi I, Stress, Rest, and Medi II. They distinguish two different classification tasks based on these protocols. First, they define a three-class problem: baseline vs. stress vs. amusement. Second, they define a binary classification: baseline vs. stress. In this work, we focus on creating a binary classification to detect stress. We consider EDA data in the Baseline section labeled as "not-stressed" (0), and EDA data in the Stress section labeled as "stressed" (1) to create our model. Our model is created based on the EDA data collected from Empatica E4 wristband [96].

2.4.3 Stress Models Accuracy

We build our stress model based on four different machine learning algorithms (kNN, Naive Bayes, Random Forest, and SVM). We train two different models based on the input features for each machine learning algorithm and report their accuracy: (1) The first model trains the data using only statistical features. (2) The second model trains the data using only automatic features. We consider 30 percent of the data (5 subjects) as test and the rest as training. Figure 2.3 shows the accuracy of the four different classifiers used based on two different stress models. The best accuracy for the first model belongs to the random forest with depth equals to 1, which is 90%. The best accuracy for the second model belongs to the NaiveBayes classifer which is equal to 97%. In [86], they achieve the accuracy of 79.71% on the binary classification for EDA signals collected from Empatica E4 wristband. The results show that our proposed pipeline outperforms their method in extracting related features for creating stress models (for both automatic and statistical features).

Our results show that, in all four machine learning algorithms, we achieve a higher accuracy using automatically extracted features compared with statistical features. To detect other types of stimulus than stress, one might need to add some extra statistical features to achieve an acceptable accuracy. However, automatic feature extraction module in our tool can still be used to extract the most important features regardless of the stimulus. This is the main advantage of our open-source tool in comparison with other tools for EDA feature extraction.

2.5 Conclusion

We presented pyEDA, a user-friendly open-source toolkit in Python, to extract statistical and automatic features from EDA data. To the best of our knowledge, this is the first work presenting an automatic feature extraction module in an open-source tool for EDA data. There is no previous study claiming what statistical features of EDA data are the best features to detect different types of stimuli. Our tool uses autoencoders to automatically extract any arbitrary number of features representing the lower dimensional representation of the input data. These features can directly be used in machine learning algorithms for classification and stimulus detection. Different types of stimuli can have different effects on EDA signal, therefore the type of statistical features to extract from the signal can be different in each scenario. As a result, presenting a toolkit which is able to extract both statistical and automatic features of EDA data can facilitate and accelerate research in EDA

Chapter 3

Motion Artifact Removal for Ensuring System Robustness and Enhanced Performance

Photoplethysmography (PPG) is a simple and cost-effective optical technique widely used in healthcare to extract valuable health-related information such as heart rate variability, blood pressure, and respiration rate. PPG signals can be easily and continuously collected using portable wearable devices. However, these devices are susceptible to motion artifacts caused by daily activities. The most common methods to eliminate these artifacts involve using additional accelerometer sensors, which have two significant limitations: high power consumption and the need to integrate an accelerometer sensor into the wearable device, which is unnecessary for certain wearables.

In this chapter, we propose a low-power, non-accelerometer-based method for removing motion artifacts from PPG signals, which outperforms existing methods in accuracy. Incorporating a motion artifact removal module to accurately filter noise from PPG signals is essential to ensure the robustness of the system, significantly enhancing the performance and accuracy of our closed-loop system architecture for improved stimuli predictions. We utilize a Cycle Generative Adversarial Network to reconstruct clean PPG signals from noisy ones. Our novel machine-learning-based technique achieves a 9.5 times improvement in motion artifact removal compared to the state-of-the-art methods, without using additional sensors such as an accelerometer, leading to a 45% increase in energy efficiency.

3.1 Introduction

A photoplethysmography (PPG) is a simple, low-cost, and convenient optical technique used for detecting volumetric blood changes in the microvascular bed of target tissue [97]. Valuable health-related information can be extracted from PPG signals such as heart rate and heart rate variability.

Nowadays, PPG signals can easily be collected continuously and remotely using inexpensive, convenient, and portable wearable devices (.e.g., smartwatches, rings, etc.) which makes them a suitable source in wellness applications in everyday life. However, PPG signals collected from portable wearable devices in everyday settings are often measured when a user is engaged with different kinds of activities and therefore are distorted by motion artifacts. The signal with a low signal-to-noise ratio leads to inaccurate vital signs extraction which may risk life-threatening consequences for healthcare applications. There exists a variety of methods to detect and remove motion artifacts from PPG signals. The majority of the works related to the detection and filtering of motion artifacts in PPG signals can reside in three categories: (1) non-acceleration based, (2) using synthetic reference data, and (3) using acceleration data.

The non-acceleration based methods do not require any extra accelerometer sensor for motion artifact detection and removal. In existing works, these approaches utilize certain statistical methods due to the fact that some statistical parameters such as skewness and kurtosis will remain unchanged regardless of the presence of the noise. In [98], such statistical parameters are used to detect and remove the impure part of the signal due to motion artifacts. In [99], authors detect motion artifacts using a Variable Frequency Complex Demodulation (VFCDM) method. In this method, the PPG signal is normalized after applying a band-pass filter. Then, to detect motion artifacts, VFCDM distinguishes between the spectral characteristics of noise and clean signals. Then, due to a shift in the frequency, an unclean-marked signal is removed from the entire signal. Another method in this category is proposed in [100] that uses the Discrete Wavelet Transform (DWT) method.

In non-accelerometer based methods, the clean output signal is often shorter than the original signal, since unrecovered noisy data is removed from the signal. To mitigate this problem, a synthetic reference signal can be generated from the corrupted PPG signal. In [101], authors use Complex Empirical Mode Decomposition (CEMD) to generate signals. In [102], two PPG sensors are being used to generate a reference signal. One of the sensors is a few millimeters away from the skin, which only measures PPG during movements. First a band-pass filter is applied on both recorded signals; then, an adaptive filter is used to minimize the difference between two recorded signals.

Sensors are the most critical part of wearable sensing devices, and their sensitivity plays an important role[103, 104]. Often an accelerometer sensor is also embedded in wearable devices. To eliminate the effect of motion artifacts, acceleration data can be used as a reference signal. In [105], with the help of acceleration data, Singular Value Decomposition (SVD) is used for generating a reference signal for an adaptive filter. Then, the reference signal and PPG signal pass through an adaptive filter to remove motion artifacts. With a similar approach, authors in [106] use DC remover using another type of adaptive filter. Another method for motion artifact removal is proposed in [107] which follows three steps:

(1) signals are windowed, (2) the output signal is filtered, and (3) a Hankel data matrix is constructed.

Even though using an accelerometer-based method increases the model's accuracy, it suffers from two limitations: i) high power consumption and ii) the need to integrate an accelerometer sensor in a wearable device (which is not required in certain wearables). To overcome these issues, machine learning techniques can be employed as an alternative method to remove noise and reconstruct clean signals [83, 108, 109]. Furthermore, machine learning techniques, proven to be useful in numerous research areas [32, 34, 37, 110, 111], are utilized in healthcare domain in processing of a variety of physiological signals such as PPG for data analysis purposes [4, 9, 10, 12, 112, 113]. The aim of this chapter is to propose a machine learning non-accelemoter-based PPG motion artifacts removal method which is low-power and can outperform the accuracy of the existing methods (even the accelerometer-based techniques). In recent studies, applying machine learning for image noise reduction has been investigated extensively. The most recent studies use deep generative models to reconstruct or generate clean images [114, 115]. In this chapter, we propose a novel approach which converts noisy PPG signals to a proper visual representation and uses deep generative models to remove the motion artifacts. We use a Cycle Generative Adversarial Network (CycleGAN) [116] to reconstruct clean PPG signals from noisy PPG. CycleGAN is a novel and powerful technique in unsupervised learning, which targets learning the distribution of two given datasets to translate an individual input data from the first domain to a desired output data from the second domain. The advantages of CycleGAN over other existing image translation methods are i) it does not require the pairwise dataset, and ii) the augmentation in CycleGAN makes it practically more suitable for datasets with fewer images. Hence, we use CycleGAN to remove motion artifacts from noisy PPG signals and reconstruct the clean signals. Our experimental results demonstrate the superiority of our approach compared to the state-of-the-art with a 9.5 times improvement with approximately 45% improvement in the energy efficiency due to eliminating accelerometer sensors.

The rest of this chapter is organized as follows. Section Methods introduces the employed dataset and our proposed pipeline architecture. In section Results we summarize the result obtained by our proposed method and compare our result with the state-of-the-art in motion artifact removal from PPG signals. Finally, in the Conclusion section we discuss the strengths and limitations of our method and we cover the future work.

3.2 Methods

In this chapter, we present an accurate non-accelerometer-based motion artifacts removal model from PPG signals. This model mainly consists of a module for artifact detection and another one for motion artifact removal. We present in Figure 3.1 the flow chart of our proposed model.

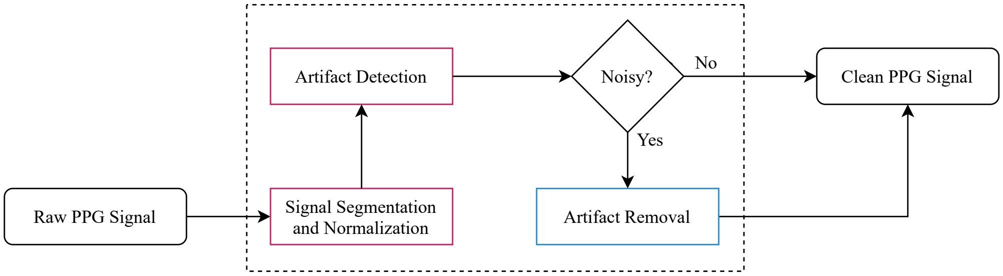


Figure 3.1: Flowchart of the proposed method

The artifact removal module consists of sub-modules (Figure 3.2) that deliver the task of cleaning the input signal by transforming it into a two dimensional image and using Cycle-GAN to remove the two dimensional noise induced by the artifacts. Consequently, the clean image is transformed to a signal that is returned in the output. Each of these modules are discussed in detail in their corresponding sections.

In order to train this model, two datasets of PPG signals are required: one consisting of clean PPG signals and the other one containing noisy PPG signals (In the rest of this chapter, by noisy PPG signals we are refering to PPG signals affected by motion artifact). The

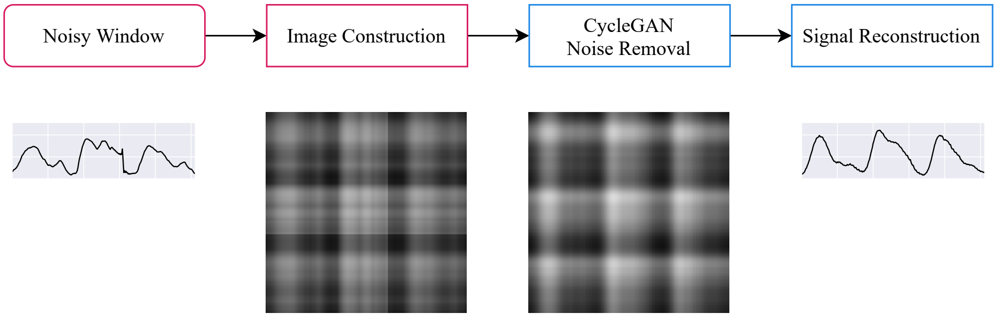


Figure 3.2: Flowchart of the artifact removal module

model's evaluation requires both clean and noisy signals to be taken from the same patient in the same period of time. However, recording such data is not feasible as patients are either performing an activity, which leads to recording a noisy signal or are in a steady-state, which produces a clean signal. For this reason, we simulate the noisy signal by adding data from an accelerometer to the clean signal. This is a common practice and has been used earlier in related work (e.g., [117]) to address this issue. This way, the effectiveness of the model can be evaluated efficiently by comparing the clean signal with the reconstructed output of the model on the derived noisy signal. In the following subsections, we explain the process of data collection for both clean and noisy datasets.

3.2.1 BIDMC Dataset

For the clean dataset, we use BIDMC dataset [118]. This dataset contains signals and numerics extracted from the much larger MIMIC II matched waveform database, along with manual breath annotations made from two annotators, using the impedance respiratory signal.

The original data was acquired from critically ill patients during hospital care at the Beth Israel Deaconess Medical Centre (Boston, MA, USA). Two annotators manually annotated individual breaths in each recording using the impedance respiratory signal. There are 53 recordings in the dataset, each being 8 minutes long and containing:

- Physiological signals, such as the PPG, impedance respiratory signal, and electrocardiogram (ECG) sampled at 125 Hz.
- Physiological parameters, such as the heart rate (HR), respiratory rate (RR), and blood oxygen saturation level (SpO2) sampled at 1 Hz.
- Fixed parameters, such as age and gender. The ages range from 19 to higher than 90. Also, out of 53 subjects in this dataset, 20 of them are males and 32 are females (one subject's sex is not determined).
- Manual annotations of breaths.

3.2.2 Data Collection

We conducted laboratory-based experiments to collect accelerometer data for generating noisy PPG signals. Each of these laboratory-based experiments consisted of 27 minutes of data. A total of 33 subjects participated in the laboratory-based experiments. The ages of the subjects ranged from 20 to 62, and 17 of them were males while 16 were females. In each experiment, subjects were asked to perform specific activities while the accelerometer data were collected from them using an Empatica E4 [119] wristband worn on their dominant hand. The Empatica E4 wristband is a research-grade wearable device that offers real-time physiological data acquisition, enabling researchers to conduct in-depth analysis and visualization. A recent research study detects and discriminates acute psychological stress (APS) in the presence of concurrent physical activity (PA) using the PPG and the Accelerometer data collected from Empatica E4 wristband [120]. Figure 3.3 shows our experimental procedure. Note that the accelerometer signals are only required for generating/emulating noisy PPG signals, and our proposed motion artifact removal method does not depend on having access to acceleration signals.

<table>
<tr><td rowspan="2">Rest</td><td>Finger Tappping</td><td>Finger Tappping</td><td rowspan="2">Rest</td><td>Waving</td><td>Waving</td><td rowspan="2">Rest</td><td>Shaking Hands</td><td>Shaking Hands</td></tr>
<tr><td>Low Intensity</td><td>High Intensity</td><td>Low Intensity</td><td>High Intensity</td><td>Low Intensity</td><td>High Intensity</td></tr>
<tr><td colspan="3">Activity 1</td><td colspan="3">Activity 2</td><td colspan="3">Activity 3</td></tr>
</table>

<table>
<tr><td rowspan="2">Rest</td><td>Runnig Arm Swings</td><td>Runnig Arm Swings</td><td rowspan="2">Rest</td><td>Fist Open/Close</td><td>Fist Open/Close</td><td rowspan="2">Rest</td><td>3D Arm Movement</td><td>3D Arm Movement</td></tr>
<tr><td>Low Intensity</td><td>High Intensity</td><td>Low Intensity</td><td>High Intensity</td><td>Low Intensity</td><td>High Intensity</td></tr>
<tr><td colspan="3">Activity 4</td><td colspan="3">Activity 5</td><td colspan="3">Activity 6</td></tr>
</table>

Figure 3.3: Experimental procedure to collect accelerometer data.

According to Figure 3.3, each experiment consists of 6 different activities: (1) Finger Tapping, (2) Waving, (3) Shaking Hands, (4) Running Arm Swing, (5) Fist Opening and Closing, and (6) 3D Arm Movement. Each activity lasts 4 minutes in total, including two parts with two different movement intensities (low and high), each of which lasts 2 minutes. Activity tasks are followed by a 30 seconds rest (R) period between them. During the rest periods, participants were asked to stop the previous activity and put both their arms on a table, and stay in a steady state. Accelerometer data collected during each of the activities were later used to model the motion artifact. We describe this in the next subsection.

3.2.3 Noisy PPG signal generation

To generate noisy PPG signals from clean PPG signals, we use accelerometer data collected in our study. Clean PPG signals are directly collected from the BIDMC dataset. Accelerometer data is taken at 32 Hz, thus we down-sample the clean signals to 32 Hz to ensure they are synchronized with the collected accelerometer data.

Empatica has an onboard MEMS type 3-axis accelerometer that measures the continuous gravitational force (g) applied to each of the three spatial dimensions (x, y, and z). The scale is limited to ± 2g. Figure 3.4 shows an example of accelerometer data collected from E4.

Along with the raw 3-dimensional acceleration data, Empatica also provides a moving average

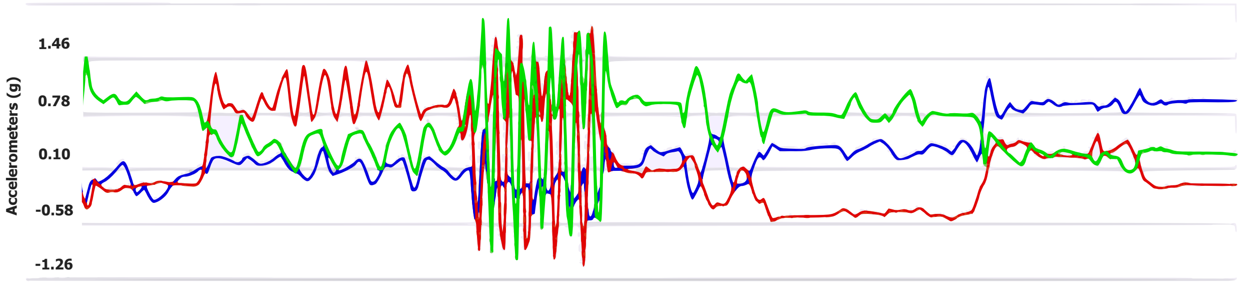


Figure 3.4: An example of Accelerometer data from Connect, the subject moves into position, walks, runs, and then simulates the turning of a car's steering wheel. The dimensional axes are depicted in red, green and blue.

of the data. Every second, the following summation is calculated over the (32 samples) input received from the accelerometer sensor,

$$S = \sum_{t=1^{32}\max(|\mathrm{Acc}_x[t]-\mathrm{Acc}_x[1]|,|\mathrm{Acc}_y[t]-\mathrm{Acc}_y[1]|,|\mathrm{Acc}_z[t]-\mathrm{Acc}_z[1]|)} \tag{3.1}$$

where $\mathrm{Acc}_i[t]$ is the value of the accelerometer sensor (g) along the i-th dimension at time frame (sample) t, and $\mathrm{Acc}_i[1]$ is the first value of the accelerometer sensor (g) along the i-th dimension in the current window. The $\max(a, b, c)$ function simply returns the maximum value among a, b, and c. It is worth to mention that the values stored in the arrays Acc_x, Acc_y, and Acc_z change after each window is processed.

Afterwards, the value of the moving average for the new window will be calculated based on this summation and the value of the moving average on the previous window,

$$\mathrm{Avg}[w] = 0.9 \times \mathrm{Avg}[w-1] + 0.1 \times \frac{S}{32} \tag{3.2}$$

Figure 3.5 visualizes this moving average over the data.

This filtered output (Avg) is directly used as a model for motion artifacts in our study.

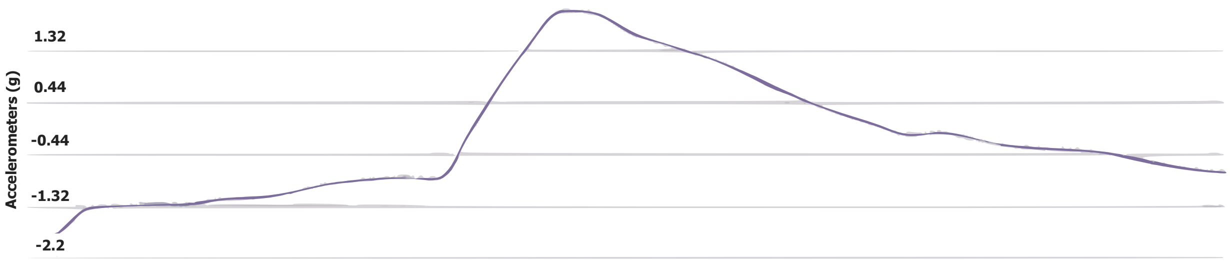


Figure 3.5: The same data as Figure 3.4 is visualized using the moving average. From Connect, the subject moves into position, walks, runs, and then simulates the turning of a car's steering wheel. The dimensional axes are depicted in red, green and blue.

To simulate the noisy PPG signals, we add this artifact model to a 2 minutes window of the clean PPG signals collected from the BIDMC dataset. We use 40 out of 53 signals in BIDMC directly as the clean dataset for training. Among these 40 signals, 20 are selected and augmented with the accelerometer data to construct the noisy dataset for training. The 13 remaining BIDMC signals and accelerometer data were added together to form the clean and noisy datasets for testing. In the rest of this section we describe each part of the model introduced in Figure 3.1.

3.2.4 Noise Detection

To perform noise detection, first, the raw signal , which is downsampled to 32 Hz, is normalized by a linear transformation to map its values to the range $(0, 1)$. This can be performed using a simple function as below:

$$\mathrm{Sig}_{\mathrm{norm}} = \frac{\mathrm{Sig}_{\mathrm{raw}} - \min(\mathrm{Sig}_{\mathrm{raw}})}{\max(\mathrm{Sig}_{\mathrm{raw}}) - \min(\mathrm{Sig}_{\mathrm{raw}})} \tag{3.3}$$

where $\mathrm{Sig}_{\mathrm{raw}}$ is the raw signal and $\mathrm{Sig}_{\mathrm{norm}}$ is the normalized output. Then, the normalized signal is divided into equal windows of size 256, which is the same window size we use for noise removal. These windows are then used as the input of the noise detection module to identify the noisy ones.

The similar type of machine learning network used in [81] can be employed as a noise detection system. To explain the network structure for the noise detection method (Table 3.1 and Figure 3.6), first, we use a 1D-convolutional layer with 70 initial random filters with a size of 10 to select the basic features of the input data and convert the matrix size from 256 × 1 to 247 × 70. To extract more complex features from the data, another 1D-convolutional layer with the same filter size 10 is required. As the third layer, a pooling layer with a filter size of 3 is utilized. In this layer, a sliding window slides over the input of the layer and in each step, the maximum value of the window is applied to the other values. This layer converts a matrix size of 238 × 70 to 79 × 70. To select additional complex features, another set of convolutional layers are used with a different filter size. This set is followed by two fully connected layers of sizes 32 and 16. Lastly, a dense layer of size 2 with a softmax activation would produce the probability of each class: clean and noisy. The maximum of these two probabilities would be identified as the result of the classification. The accuracy of our proposed binary classification method is 99%, which means that the system can almost always detect a noisy signal from a clean signal.

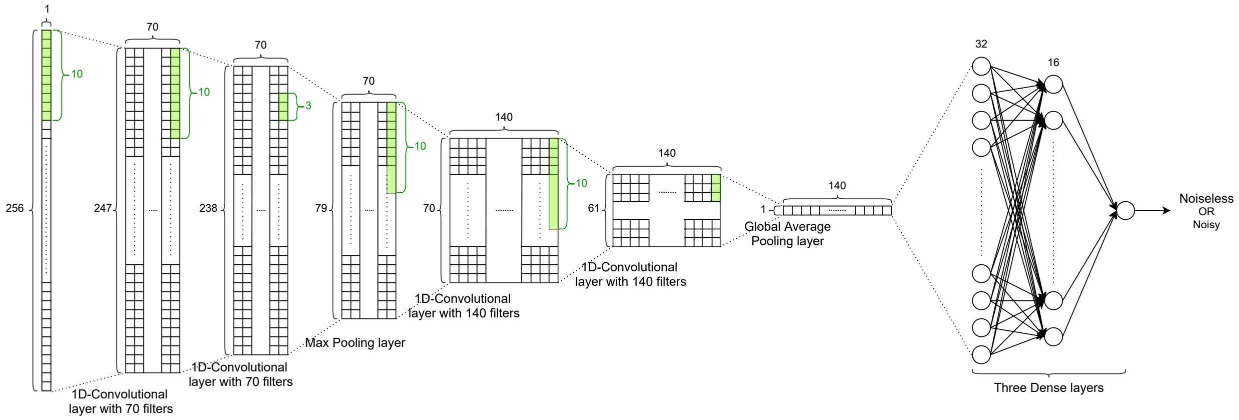

1
10
256
1D-Convolutional layer with 70 filters
70
10
247
1D-Convolutional layer with 70 filters
70
3
238
Max Pooling layer
70
10
79
1D-Convolutional layer with 140 filters
140
10
70
1D-Convolutional layer with 140 filters
140
61
140
1
Global Average Pooling layer
32
16
Three Dense layers
Noiseless
OR
Noisy

Figure 3.6: The structure of the noise detection model.

Table 3.1: The layer configuration of the noise detection model.

Layer	Structure	Output
Conv1D+Relu	70 × 10	247 × 70
Conv1D+Relu	70 × 10	238 × 70
Max pooling 1D	3	79 × 70
Conv1D+Relu	140 × 10	70 × 140
Conv1D+Relu	140 × 10	61 × 140
Global average pooling	N/A	140
Dense+Relu	128	32
Dense+Relu	16	16
Dense+Softmax	2	2

3.2.5 Noise Removal

In this section, we explore the reconstruction of noisy PPG signals using deep generative models. Once a noisy window is detected, it is sent to the noise removal module for further processing. First, the windows are transformed into 2-dimensional images, to exploit the power of existing image noise removal models, and then a trained CycleGAN model is used to remove the noise induced by the motion artifact from these images. In the final step of the noise removal, the image transformation is reversed to obtain the clean output.

The transformation needs to provide visual features for unexpected changes in the signal so that the CycleGAN model would be able to distinguish and hence reconstruct the noisy parts. To extend the 1-dimensional noise on the signal into a 2-dimensional visual noise on the image, we apply the following transformation:

$$\mathrm{Img}[i, j] = \mathrm{floor}((\mathrm{Sig}[i] + \mathrm{Sig}[j]) \times 128) \tag{3.4}$$

where Sig is a normalized window of the signal, Img is the 2d array storing the grayscale image, and i and j are time frames in the window. Each pixel, i.e. each entry of Img, will then have a value between 0 and 255, representing a grayscale image. An example of such transformation is provided in Figure 3.7 for both clean and noisy signals. According to this figure, the noise effect is visually observable in these images.

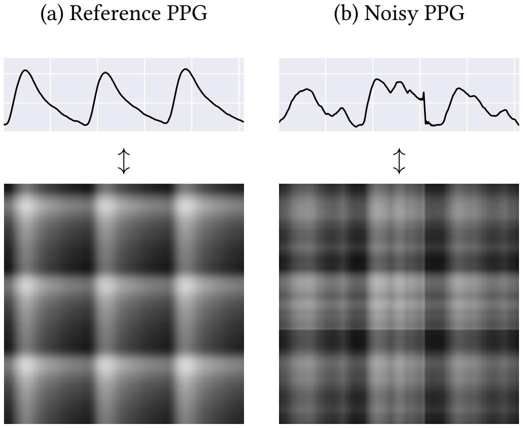


Figure 3.7: An example of signal to image transformation.

Autoencoders and CycleGAN are two of the most powerful approaches for image translation. These methods have proven to be effective in the particular case of noise reduction. Autoencoders require the pairwise translation of every image in the dataset. In our case, clean and noisy signals are not captured simultaneously, and their quantity differs. CycleGAN, on the other hand, does not require the dataset to be pairwise. Also, the augmentation in CycleGAN makes it practically more suitable for datasets with fewer images. Hence, we use CycleGAN to remove motion artifacts from noisy PPG signals and reconstruct the clean signals.

CycleGAN is a Generative Adversarial Network designed for the general purpose of image-to-image translation. CycleGAN architecture was first proposed by Zhu et al. in [116].

The GAN architecture consists of two networks: a generator network and a discriminator network. The generator network starts from a latent space as input and attempts to generate new data from the domain. The discriminator network aims to take the generated data as an input and predict whether it is from a dataset (real) or generated (fake). The generator is updated to generate more realistic data to better fool the discriminator, and the discriminator is updated to better detect generated data by the generator network.

The CycleGAN is an extension of the GAN architecture. In the CycleGAN, two generator networks and two discriminator networks are simultaneously trained. The generator network takes data from the first domain as an input and generates data for the second domain as an output. The other generator takes data from the second domain and generates the first domain data. The two discriminator networks are trained to determine how plausible the generated data are. Then the generator models are updated accordingly. This extension itself cannot guarantee that the learned function can translate an individual input into a desirable output. Therefore, the CycleGAN uses a cycle consistency as an additional extension to the model. The idea is that output data by the first generator can be used as input data to the second generator. Cycle consistency is encouraged in the CycleGAN by adding an additional loss to measure the difference between the generated output of the second generator and the original data (and vice versa). This guides the data generation process toward data translation.

In our CycleGAN architecture, we apply adversarial losses [121] to both mapping functions ($G : X \to Y$ and $F : Y \to X$). The objective of the mapping function G as a generator and its discriminator D_Y is expressed as below:

$$L_{GAN}(G, D_Y, X, Y) = E_{y \sim p_{data}(y)}[\log \log D_Y(y)] + E_{x \sim p_{data}(x)}[\log \log(1 - D_Y(G(x)))] \quad (3.5)$$

where the function G takes an input from domain X (e.g., noisy PPG signals), attempting to generate new data that look similar to data from domain Y (e.g., clean PPG signals). In the meantime, D_Y aims to determine whether its input is from the translated samples $G(x)$ (e.g., reconstructed PPG signals) or the real samples from domain Y. A similar adversarial loss is defined for the mapping function $F : Y \to X$ as $L_{GAN}(F, D_X, Y, X)$.

As discussed before, adversarial losses alone cannot guarantee that the learned function can map an individual input from domain X to the desired output from domain Y. In [116], the authors argue that to reduce the space of possible mapping functions even further, learned mapping functions (Y and F) need to be cycle-consistent. This means that the translation cycle needs to be able to translate back the input from domain X to the original image as $X \rightarrow G(X) \rightarrow F(G(X)) \sim X$. This is called forward cycle consistency. Similarly, backward cycle consistency is defined as: $y \rightarrow F(y) \rightarrow G(F(y)) \sim y$. This behavior is presented in our objective function as:

$$L_{\text{cyc}}(G, F) = E_{x \sim p_{data}(x)}[||F(G(x)) - x\|_1] + E_{y \sim p_{data}(y)}[||G(F(y)) - y\|_1] \tag{3.6}$$

Therefore, the final objective of CycleGAN architecture is defined as:

$$L(G, F, D_X, D_Y) = L_{\text{GAN}}(G, D_Y, X, Y) + L_{\text{GAN}}(F, D_X, Y, X) + \lambda L_{\text{cyc}}(G, F) \tag{3.7}$$

where λ controls the relative importance of the two objectives.

In Equation 3.7, G aims to minimize the objective while an adversary D attempts to maximize it. Therefore, our model aims to solve:

$$G^*, F^* = \operatorname{argmin} L(G, F, D_X, D_Y) \tag{3.8}$$

The architecture of the generative networks is adopted from Johnson et al. [122]. This network contains four convolutions, several residual blocks [123], and two fractionally-strided convolutions with stride 0.5. For the discriminator networks, they use 70 × 70 PathGANs [124–126].

After the CycleGAN is applied to the transformed image, the diagonal entries are used to retrieve the reconstructed signal.

$$\mathrm{Sig}_{\mathrm{rec}}[i] = \mathrm{Img}[i, i]/256 \tag{3.9}$$

3.3 Results

In this section, we assess the efficiency of our model based on the following measures: root mean square error (RMSE) and peak-to-peak error (PPE). A signal window size of 256 and an image size of 256 by 256 were used for all experimental purposes, and 25% of the data was assigned for validation. The noise detection module had an accuracy of 99%. The summary of the results for noise removal, including the improvement for each noise type and noise intensity, can be found in Table 3.2.

For each noise type, there are two entries in this table, one corresponding to the slow movement and the other one corresponding to the fast movement. The average S/N value for slow movements is 21.7dB, as provided in the table, while the average S/N value for fast movements is 14.0dB. For each of the measures, RMSE and PPE, we calculated the error between the generated signal and the reference signal as well as the error between the noisy signal and the reference signal in order to observe the improvement of the model on the noisy signal. The degree of improvement on each noise type is added in a separate column in the table. According to the table, the average of improvement on RMSE is 41× and the average of improvement on PPE is 58×.

An example of a reconstructed signal is presented in Figure 3.8, together with the noisy PPG and the reference PPG signal. As we can see in this figure, the noise is significantly reduced, and the peak values are adjusted accordingly, confirming that the image transformation successfully represents the noise in a visual format.

Table 3.2: Results of the proposed method.

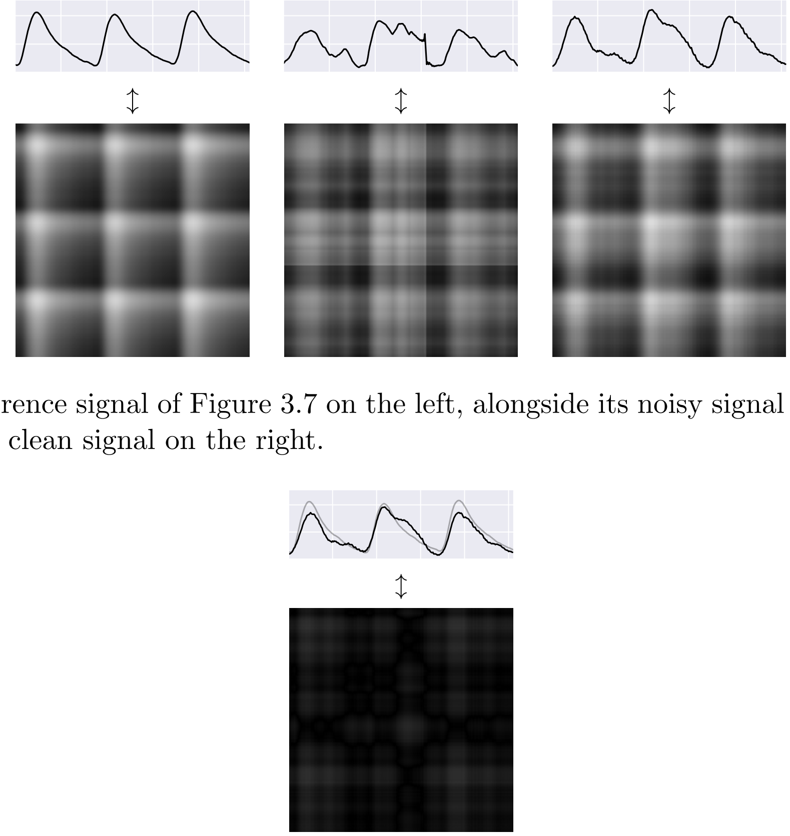
Noise Type	S/N (dB)	RMSE Gen. (BPM)	RMSE Nsy. (BPM)	RMSE Imprv.	PPE Gen. (BPM)	PPE Nsy. (BPM)	PPE Imprv.
Waving	20.04	0.213	41.76	196.07×	0.136	32.89	241.60×
Waving	11.30	2.43	55.30	22.75×	1.088	37.90	34.84×
3D Arm Movement	20.17	1.644	92.12	56.03×	0.772	44.03	57.06×
3D Arm Movement	13.12	1.688	65.99	39.10×	0.700	48.49	69.29×
Shaking Hands	21.66	1.556	61.89	39.78×	0.576	28.62	49.71×
Shaking Hands	14.96	4.203	84.31	20.06×	2.677	64.58	24.12×
Finger Tapping	22.99	1.758	63.43	36.07×	0.653	45.14	69.14×
Finger Tapping	13.99	3.008	21.76	7.235×	1.191	10.70	8.99×
Fist Open Close	25.11	1.648	35.74	21.69×	0.528	24.51	46.44×
Fist Open Close	16.69	2.151	51.28	23.84×	1.113	42.65	38.33×
Running Arm	20.14	2.056	22.93	11.16×	0.715	19.32	27.02×
Running Arm	13.98	3.807	77.73	20.42×	1.348	50.75	37.64×
Average	17.85	2.18	56.19	41.18×	0.958	37.465	58.68×

3.3.1 Comparison

In this section we compare our model's efficiency with the state-of-the-art (Table 3.3). To minimize the difference between our experimental setup and the setups used in the related works we use the same measures. Such comparison for the state-of-the-art artifact detection and artifact removal algorithms has been made comprehensively in [127] , where the algorithms are compared according to their input/output PPE and RMSE. We use [127] as the base of our comparison and we provide PPE and RMSE improvements of our method to display its efficiency with respect to both measures. It should be noted that it is not feasible to perform a close comparison between our model and the existing works, due to the differences in the datasets and the lack of a public dataset providing noisy and clean signals simultaneously.

Table 3.3: The summary comparison of our result with the existing methods. MAE stands for Mean absolute error.

Paper	Method	Accelerometer	Before	Outcome
Proposed method	**CycleGAN**	**No**	**PPE 37.46 BPM** **RMSE 56.18 BPM**	**PPE 0.95 BPM** **RMSE 2.18 BPM**
Hanyu and Xiaohui [98]	Statistical Evaluation	No	PPE 8.1 BPM	PPE 7.85 BPM
Bashar et al. [99]	VFCDM	No	N/A	6.45% false positive
Lin and Ma [100]	DWT	No	PPE 13.97 BPM	PPE 6.87 BPM
Raghuram et al. [101]	CEMD LMS	Syn.	PPE 0.466 BPM	PPE 0.392 BPM
Hara et al. [102]	NLMS and RLS	Syn.	RMSE 28.26 BPM	RMSE 6.5 BPM
Tanweer et al. [105]	SVD and X-LMS	Yes	N/A	PPE 1.37 BPM
Wu et al. [106]	DC remover and RLS	Yes	N/A	STD 3.81
Bac´a et al. [107]	MAR and AT	Yes	N/A	MAE 2.26 BPM
Askari et al. [128]	SSA + MA Removal	No	N/A	RMSE 6.73 BPM



(a) The clean reference signal of Figure 3.7 on the left, alongside its noisy signal in the middle, and the reconstructed clean signal on the right.

(b) A comparison of the reference signal and the reconstructed signal together with the residual image of the reconstruction is included in this figure. The brighter the pixels are the more difference the reconstructed image has in that pixel compared to the original image, meaning that the motion artifact is slightly apparent in the reconstructed signal.

Figure 3.8: Signal reconstruction

In comparison to non-accelerometer-based methods, our model significantly outperforms these models. The best performance observed in previous work is reported in [102] that improves the average RMSE from 28.26BPM to 6.5BPM (4.3× improvement). However, our model's improvement on average RMSE is from 56.18 to 2.18 (25.8× improvement). In most of the existing accelerometer-based methods, no value is provided for the degree of the input noise. Although the best reported PPE belongs to [101] with an outcome of 0.392BPM, the best improvement is achieved by [100] from 13.97BPM to 6.87BPM (2.03× improvement). However, our model's improvement on average PPE is from 37.46BPM to 0.95BPM (39.4× improvement).

3.3.2 Resource Usage

In this section, we provide detailed information about the resources being used. Previously, we claimed that our implemented model consumes a lower amount of power in comparison to the accelerometer-based models. We designed an experiment to measure the consumed power for our implemented model and accelerometer-based models to compare power consumption. First, we did 32Hz-sampling in a Raspberry Pi 4 device for five minutes by using a low-power accelerometer, ADXL343 [129]. Then, we measured the average consumed power of this task by using SmartPower2 5VDC Power Supply [130]. Secondly, we measured the average power consumption of our CycleGAN model in the test phase for five minutes with the same raspberry pi and power analyzer. In other words, we did these sub-tasks (1) trained the model, (2) tested our pre-trained model on the raspberry pi, and (3) monitored the power consumption. For embedded devices, the critical key is being low power in the training phase is not crucial; since training can be done on the cloud instead of the device itself. In table 3.4, you can find the results of this experiment. According to Table 3.4, our proposed method in average uses 45% less power compared to an accelerometer-based artifact removal method.

Table 3.4: Average power consumption of raspberry pi 4

	Idle	Accelerometer-based	**Proposed Method**
Power Consumption	2.23 W	3.76 W	**3.07 W**

To complete the analysis for resource usage, we also considered average time and memory consumption. We completed our proposed model's training and testing phase on a server, with a GPU of RTX 3080 Ti and a CPU of Xeon E5-2680 v2, and monitored the resources. In table 3.5, the information about resource usage is provided. Since our purpose is cleaning the PPG signal in real-time, low time consumption in the test phase is important. Based on the results, our model needs just 0.3 Sec to clean the PPG signal, which makes it a feasible solution for real-time PPG noise removal.

Table 3.5: Average time and memory consumption of the proposed method in training and testing phase

	Time	CPU Memory	GPU Memory
Train	909.786 Sec	488 MB	10828MB
Test	**0.398 Sec**	26 MB	1MB

3.4 Discussion

Noise reduction has been extensively studied in image processing, and the introduction of powerful models such as CycleGAN has shown promising results in terms of noise reduction in images. Inspired by this fact, we proposed a signal to image transformation that visualizes signal noises in the form of image noise. To the best of our knowledge, this is the first use of CycleGAN for bio-signal noise reduction which eliminates the need for an accelerometer to be embedded into wearable devices, which in turn helps to reduce the power consumption and cost of these devices.

It should be noted that despite the significant benefits of our proposed method in removing noise in different situations, it may not be effective in all possible scenarios. Clearly, the intensity of noise applied to the signals, and the variations of the noise, also called noise categories, are controlled for the purpose of this study. In other words, if the source of the generated motion artifact is changed in a way that the range in heart rate variations is observable in compared with existing activities in this work, this method may not be applicable. Although it will improve the error, it does not guarantee a reasonable upper bound. However, the same limitations also exist in the related works.

3.5 Conclusions

In this chapter, we presented an image processing approach to the problem of noise removal from PPG signals where the noise is selected from a set of noise categories that simulate the daily routine of a person. This method does not require an accelerometer on the sensor, therefore, it can be applied to other variations of physiological signals, such as ECG, to reduce the power usage of the measuring device and improve its efficiency. In this work, the novel use of CycleGAN as an image transformer is leveraged to transform such physiological signals. On average, the reconstructed PPG performed using our proposed method offers 41× improvement on RMSE and 58× improvement on PPE, outperforming the state-of-the-art by a factor of 9.5.

Chapter 4

Integrating Respiratory Rate Extraction for Advanced Multimodal Analysis

Respiratory rate (RR) is a crucial clinical sign indicating ventilation. Abnormal changes in RR are often the earliest indicators of health deterioration as the body strives to maintain oxygen delivery to its tissues. The growing interest in remotely monitoring RR in everyday settings has made photoplethysmography (PPG) wearable devices an attractive option.

PPG signals are valuable for RR extraction due to the presence of respiration-induced modulations. Traditional PPG-based RR estimation methods primarily depend on hand-crafted rules and manual parameter tuning. Recently, an end-to-end deep learning approach was proposed. However, despite its automated nature, this method's performance on real-world data remains suboptimal.

In this chapter, we present a comprehensive, end-to-end pipeline for respiratory rate (RR) extraction, utilizing Cycle Generative Adversarial Networks (CycleGAN) to accurately reconstruct respiratory signals from raw PPG data. This integration adds a critical modality to our system, enhancing its capability for advanced multimodal analysis. Our results show a significant improvement in RR estimation accuracy, achieving up to twice the accuracy (mean absolute error of 1.9±0.3 using five-fold cross-validation) compared to the state-of-the-art methods on an identical publicly available dataset. Our findings suggest that CycleGAN is a valuable technique for RR estimation from raw PPG signals.

In our proposed closed-loop system architecture, we collect PPG signals in daily life settings. Consequently, having an accurate module to extract respiratory rate from PPG signals is of significant importance.

4.1 INTRODUCTION

Respiratory rate (RR), often referred to as breathing rate, is the number of breaths a person takes per minute. A normal resting RR for adults ranges from 12 to 20 [131]. Abnormal changes in respiratory rate are an accurate indicator of physiological conditions such as anxiety, hypoxia, hypercapnia, metabolic and respiratory acidosis [132]. A diverse body of research studies has indicated the significance of respiration rate for forecasting events such as cardiac arrest, patient deterioration, and care escalation [133–135].

The importance of respiratory rate as one of the first indicators of health deterioration has attracted significant attention in RR's daily monitoring [136]. However, RR's reliable measurement devices are bulky and cumbersome, and are mainly used for inpatients. With the rapid development of wearable technologies, a change in an individual's physiological systems' functional state can be tracked and monitored in an everyday setting, for instance, by using photoplethysmography (PPG) [97]. PPG signals can easily be collected continuously and remotely using a wide range of inexpensive, convenient, and portable wearable devices (.e.g., smart watches, rings, etc.). The blood perfusion dynamics are known to carry information on breathing, as respiration-induced modulations in PPG signals [137]. Hence, PPG signals are considered as a suitable source for respiratory rate extraction to forecast unexpected care admissions in a daily life setting .

The RR estimation from PPG signals has received remarkable attention in the literature [138]. Traditional RR estimation methods require several steps, including digital filtering, time/frequency domain analysis, extraction of signal components from composite signals, deriving respiratory surrogate waveforms and features using the fiducial points, signal quality estimations, and sensor fusion [139]. These techniques rely heavily on manual parameter tuning, optimization, and hand-crafted rules designed for specific target patient population. In the past few years, machine learning techniques and neural networks have been widely used in health monitoring domains [4, 9, 81, 82, 140]. Bian et al. [141] recently proposed an end-to-end deep learning approach in order to automatically and accurately estimate RR from raw PPG signals. Despite the automatic nature of their proposed model, the performance of this method is not ideal (mean absolute error (MAE) of 3.8 ± 0.5 bpm, which is about %25 inaccuracy considering 16 bpm as an average RR per minute).

This chapter proposes an automatic end-to-end generative deep learning approach using cycle generative adversarial networks (CGAN) [142] to reconstruct respiratory signals from raw PPG signals and estimate RR with a high accuracy. CGAN is a novel and powerful approach in the field of unsupervised learning, which targets learning the structure of two given data domains to translate an individual input from one domain to a desired output from the second domain. We also propose a novel loss function to be integrated in our CGAN model that takes into account the key attribute (i.e., RR) of the generated respiratory signals. Our results demonstrate that the proposed GAN-based approach estimates RR from raw PPG signals with 2× higher accuracy compared with the state-of-the-art approach [141] using real-world data. Furthermore, our method outperforms the classical RR estimations methods, despite utilizing the complete automatic end-to-end design.

In summary, this work makes the following key contributions:

- Proposing an end-to-end automatic approach based on CGAN which outperforms the performance of the classical RR estimation methods (using an identical setting and dataset).
- Proposing a novel loss function for our CGAN model that takes into account the RR of the generated respiratory signals.
- Demonstrating the performance of our approach using the real-world data and comparing it against the state-of-the-art (using an identical setting and dataset).

The rest of this chapter is organized as follows. Section 4.2 introduces the employed dataset and our proposed pipeline architecture. In Section 4.3 we summarize the result obtained by our proposed method. Section 4.4 compares our result with the state-of-the-art in RR estimation from raw PPG signal. Finally, Section 4.5 concludes the chapter.

4.2 MATERIAL AND METHODS

4.2.1 Dataset

We employed BIDMC PPG and Respiration Dataset [143] to evaluate our RR estimator method. This is a publicly available dataset which contains signals and numerics extracted from the much larger MIMIC II matched waveform Database, along with manual breath annotations made from two annotators, using the impedance respiratory signal.

In this dataset, PPG and impedence respiratory signals are collected from 53 adult patients for about 8-minute duration at sampling rate of 125 Hz. This dataset is widely used to evaluate the performance of different algorithms for estimating respiratory rate from PPG signals [138, 141, 144, 145].

4.2.2 RR Estimation Pipeline

Figure 4.1 shows our proposed pipeline for estimating respiratory rate from PPG signals. There are three different main stages in this pipeline: (1) Data Preparation, (2) PPG to Respiration Translator (PRT), and (3) RR Estimator. In the following subsections, we discuss each part in detail.

Data Preparation

The primary purpose of this stage is to prepare the data and pre-process it for the PPG-to-Respiration Translator (PRT) module. PPG signals are sampled at much higher frequency than required in the BIDMC dataset. Therefore, downsampling is done to save memory, and reduce processing time and computational complexity of our model while preserving the signals integrity. In this stage, first, raw PPG data are normalized to 0-1. Then, the signals are down-sampled to 30 Hz. Finally, 30-second windows of data are extracted from the signals to be used in the PRT module for translation.

PPG to Respiration Translator (PRT)

In this module, the Cycle Generative Adversarial Networks (The Cycle GAN) are employed to reconstruct respiratory signals from raw PPG signals. The Generative Adversarial Networks belong to the field of unsupervised learning targeting to learn the structure of a given data in order to generate new unseen data. The GANs are composed of two models: a gen-

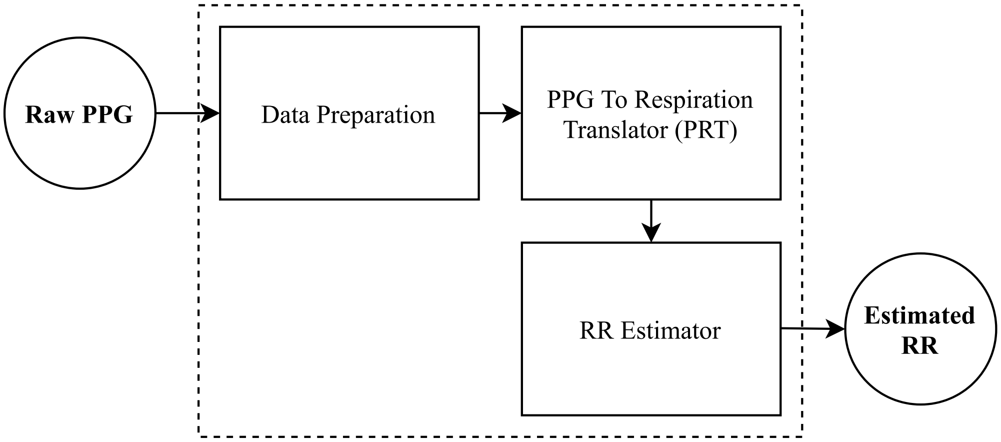


Figure 4.1: RR Estimation Pipeline

erator network and a discriminator network. The generator network starts at a point from a latent space as an input and aims to generate new data similar to the expected domain. The discriminator network on the other hand attempts to recognize if an input data is real (belongs to the original dataset) or fake (generated by the generator network).

The Cycle GAN is an extension of the Generative Adversarial Networks which was first proposed by Jun-Yan Zhu et al. [142]. The idea behind the Cycle GAN is to take an input from the first domain and generate an output of the second domain. In our case, the goal of Cycle GAN is to learn the mapping between PPG signals (domain X) and respiratory signals (domain Y). Each domain contains set of training samples $\{x_i\}_{i=1}^{N} \in X$ and $\{y_i\}_{i=1}^{N} \in Y$ used directly from BIDMC dataset. The model includes two generators with mapping functions as $G : X \rightarrow Y$ and $F : Y \rightarrow X$ and two discriminators D_X and D_Y. In the discriminator networks, D_X aims to distinguish between real PPG signals (x_i) and synthetic PPG signals ($F(y)$) while D_Y aims to discriminate between real respiratory signals (y_i) and synthetic respiratory signals ($G(x)$).

We indicate the distributions of our data as $x \sim p_{data}(x)$ and $y \sim p_{data}(y)$. Our objective loss function contains three terms: (1) adversarial losses [146], (2) cycle consistency losses, and (3) RR Loss.

Adversarial losses are employed for matching the distribution of generated synthetic signals to the data distribution of original signals. We apply adversarial loss function on both of our mapping functions. The objective function applied to the mapping function G is expressed as below:

$$\begin{aligned} L_{GAN}(G, D_Y, X, Y) = E_{y \sim p_{data}(y)}[log D_Y(y)] \\ + E_{x \sim p_{data}(x)}[log(1 - D_Y(G(x)))] \end{aligned} \tag{4.1}$$

where G tries to generate respiratory signals ($G(x)$) that look similar to original respiratory signals collected from BIDMC dataset (domain Y), while D_Y aims to discriminate between synthetic respiratory signals ($G(x)$) and real samples (y). In a same way, adversarial loss for the mapping function F is expressed as $L_{GAN}(F, D_X, Y, X)$.

A mapping function trained only by adversarial loss as an objective function can map the same set of signals from the first domain to any random permutation of signals in the second domain. Therefore, cycle consistency losses are added to guarantee the mapping from an individual input (x_i) to a desired output (y_i) by considering learned mapping functions to be cycle consistent. This means that for each PPG signal x from domain X we must have $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$ while for each respiratory signal y we have $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$. This behaviour is indicated as:

$$\begin{aligned} L_{cyc}(G, F) = E_{x \sim p_{data}(x)}[||F(G(x)) - x||_1] \\ + E_{y \sim p_{data}(y)}[||G(F(y)) - y||_1] \end{aligned} \tag{4.2}$$

In order to force the synthesized respiratory signals to only keep their main features, we define

RR loss function which attempts to take into account the RR of the generated respiratory signals. The BioSPPy [147] public python library is used to calculate the respiration rate of the synthetic and original respiratory signals. This additional loss function can be expressed as:

$$L_{RR}(G) = E_{y \sim p_{data}(y)}[||G(F(y))_{RR} - y_{RR}||_1] \tag{4.3}$$

Therefore, the final objective is the weighted sum of the above loss functions:

$$\begin{aligned} L(G, F, D_X, D_Y) = L_{GAN}(G, D_Y, X, Y) \\ + L_{GAN}(F, D_X, Y, X) \\ + \lambda_1 L_{cyc}(G, F) \\ + \lambda_2 L_{RR}(G) \end{aligned} \tag{4.4}$$

where λ_1 and λ_2 are the weights of cycle consistency loss and RR loss respectively (both are empirically selected as 10 in our work).

G and F attempt to minimize this objective against adversaries D_X and D_Y that try to maximize it. Hence, we aim to solve:

$$G^*, F^* = \arg \min_{G,F} \max_{D_X, D_Y} L(G, F, D_X, D_Y) \tag{4.5}$$

We use the CGAN architecture proposed by [142]. The architecture of generative networks is adopted from Johnson et al. [148]. This network contains two stride-2 convolutions, several residual blocks [149], and two fractionally-strided convolutions with stride 0.5. For the discriminator networks we use 70×70 PathGANs [150–152] which aims to classify whether the signals are fake or real.

RR Estimator

This module uses the BreathMetrics [153] for pre-processing, filtering, and calculating the respiratory rate of the signals. The BreathMetrics is a Matlab toolbox for analyzing respiratory recordings. Breathmetrics was designed to make analyzing respiratory recordings easier by automatically de-noising the data and extracting the many features embedded within respiratory recordings including respiratory rate. The methods used in this tool (de-noising and features extraction) have been validated, peer-reviewed, and published in Chemical Senses, a scientific journal.

4.2.3 Performance Metric

We calculate the mean absolute error (MAE) as the performance metric in our study in order to evaluate our RR estimation method. MAE is calculated by averaging the absolute differences between the values estimated by a model and the values observed.

MAE is defined as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|RR_e^i - RR_r^i| \tag{4.6}$$

Where N is total number of respiratory segments. RR_e^i and RR_r^i are estimated and reference

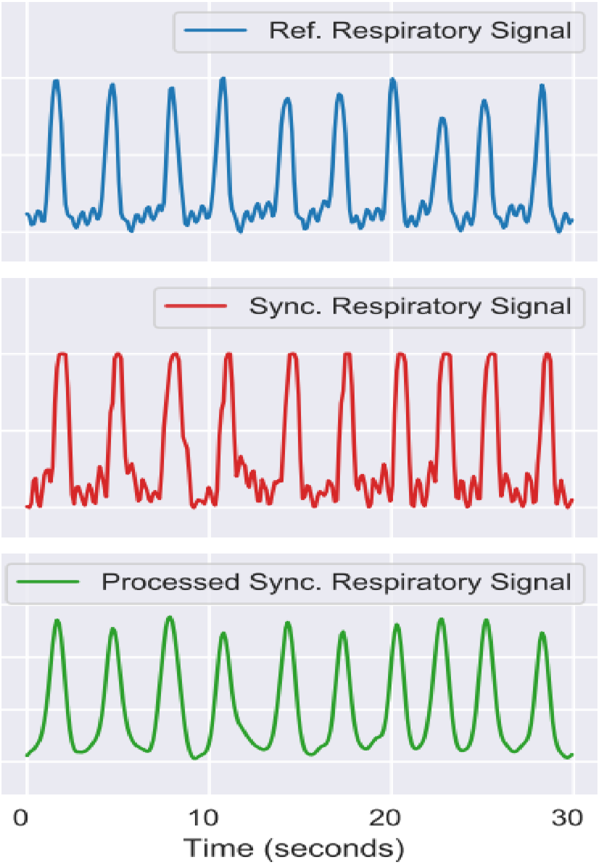


Figure 4.2: Example of the synthetic respiratory signal generated by our PRT module, from top to bottom: reference respiratory signal, synthetic respiratory signal, and processed synthetic respiratory signal using the BreathMetrics module.

respiratory rate for each 1 minute data respectively.

To evaluate the performance of our algorithm, we use 5-fold cross validation. We split the BIDMC dataset into five folds while making sure each subject's data is appeared in one fold only. We perform the training 5 times and each time 4 folds are for training and 1 fold is for testing. The average MAE of 5 runs is presented as the final performance results for our proposed model. We trained our models on Nvidia Quadro RTX 5000 GPU with 125 GB of RAM. For each training experiment we performed 100 epochs in which early stopping technique were employed to reduce over-fitting.

4.3 RESULTS

Figure 4.2 shows an example of a reconstructed respiration signal using our method alongside with its reference respiration signal for the same timing window of the same test subject. The blue signal represents the reference respiration signal from the BIDMC dataset. The red signal is the synthetic respiratory signal which is an output of our PRT module. The green signal shows the final respiratory signal after de-noising and pre-processing of the signal using the BreathMetrics tool. By comparing the reference respiratory signal and the processed synthetic respiratory signal it can be seen that our pipeline is capable of reconstructing the respiratory signal itself from the PPG signal with high precision.

Table 4.1 shows the summary of the MAE performance (average±std) of our method compared with the state of the arts. The same publicly available dataset (BIDMC) and the same performance metric (MAE using five fold cross validation) are used to fairly evaluate and compare the performance of our automatic approach.

Table 4.1: The summary of MAE Performances in compare with state of the arts

RR Estimation Models	**MAE**
Bian et al. [141] DL method	3.8±0.5
Bian et al. [141] SmartQualityFusion method	2.6±0.4
Our proposed CycleGAN-based method	1.9±0.3

As can be seen from the table, our CGAN-based method significantly (up to 2×) outperforms the state-of-the-art RR estimation methods from a raw PPG signals. In the next section, we discuss the results in more details.

4.4 DISCUSSION

Existing PPG-based RR estimation methods heavily rely on features derived from various hand-crafted algorithms and tuning parameters for specific settings. In this chapter, we presented a novel fully end-to-end automatic approach for estimating RR from raw PPG signals.

A classical RR estimation method has been implemented in [141] which is referred as SmartQualityFusion, as a combination of Smart fusion [154] and Quality fusion [155]. The SmartQualityFusion algorithm achieved the MAE of 2.6±0.4 breaths/min (Table 4.1). According to Table 4.1, our proposed RR estimation pipeline demonstrate a significant improvement in accurately estimating RR compared with their algorithm (the MAE of 1.9 breaths/min), despite the complete end-to-end automatic environment of our method.

Bian et al. [141], also proposed an end-to-end learning approach based on deep learning to estimate RR from PPG. Their results demonstrate a clinically reasonable performance; however, the performance of their proposed automatic approach heavily depends to the availability of real world data. They used the mean absolute error (MAE) as a performance metric in their work. According to their results, their deep learning approach trained with real data could achieve the MAE of 3.8±0.5 breath/minutes which is substantially higher than the MAE error achieved by their classical algorithm (SmartQualityFusion). According to Table 4.1, our proposed pipeline architecture provides about 2 times better performance of estimating respiratory rate compared with their proposed deep learning approach.

One existing limitation of our work, which is also present in the other state-of-the-art methods, is the lack of noisy PPG data in the real datasets used for training our models. Therefore, the performance of our proposed method might not be as expected on a noisy PPG data collected during daily life and physical activities. This limitation is also present in the other state-of-the-arts methods since the employed datasets are mostly consist of stationary data.

However, as also mentioned in [141], this problem can easily be diminished by retraining the model using the new existing noisy PPG and respiratory signals.

In [141], the authors increased the performance of their proposed method, by augmenting the real datasets using the generated synthetic PPG data. This enhanced the MAE of their model from 3.8 brpm to 2.5 brpm. Thus, the data augmentation approach significantly enhanced the performance of their model. However, our proposed model still achieves a better performance (1.9 brpm), despite being trained only on a real data. As a future work, we intend to further increase the performance of our proposed method by augmenting our dataset using synthetic data.

4.5 Conclusion

In summary, in this chapter we presented a novel pipeline architecture in order to estimate respiratory rate using PPG signals. We are the first one to use cycle adversarial networks in our model to reconstruct the respiratory signals from PPG signals. According to our results, our proposed pipeline architecture is able to estimate RR with the MAE of 1.9 which has the performance of 2.0x better than the state-of-the-art automatic RR estimation method.

Chapter 5

Developing a Context-Aware Closed-Loop System

In this chapter, we present our proposed context-aware closed-loop affect monitoring system, which collects both physiological data (PPG) and contextual information (such as weather conditions, phone battery level, phone usage, and location) in daily life settings. As a case study, we focus on detecting stress—an important type of affect—to evaluate the performance of our system.

Long-term unmanaged stress can lead to chronic health conditions such as diabetes, heart disease, high blood pressure, and obesity. Consequently, recent research has focused on developing systems to accurately monitor and predict daily-life stress in real time. However, most existing research evaluates stress levels in controlled settings. Current daily-life stress monitoring systems build their stress models using physiological data collected from wearable devices, such as smartwatches. A major limitation of these works is the lack of contextual data, leading to less reliable stress predictions. The literature highlights the significant importance of context data in predicting stressful moments.

To address this gap, we captured the stress levels of eleven volunteers through ecological momentary assessments (EMAs) over a period of two weeks. Our system employs a binary stress detector that differentiates between stressful and non-stressful samples, achieving an F1-score of up to 70%. This result is promising, given that the data is collected in the challenging environments of everyday settings.

5.1 Introduction

Based on recent reports, a remarkable 70% of individuals in the United States have encountered at least one symptom of stress within a given month [156]. Long-term stress can lead to a compromised immune system, cancer, cardiovascular disease, depression, diabetes, and substance addiction, among other serious effects [156]. In light of these consequences, the routine monitoring of stress levels has become increasingly essential. Thus, developing dependable techniques for promptly detecting human stress is paramount.

The utilization of physiological signals as a modality for identifying stress has been extensively explored in the literature [157, 158]. Among the physiological signals, the photoplethysmograph (PPG) signal is considered a valuable information source for stress detection [159]. This signal is influenced by the cardiac, vascular, and autonomic nervous systems, which are all known to be impacted by stress [160]. With the rapid development of wearable technologies, PPG signals can now be conveniently monitored in daily life settings using cost-effective wearable devices [157]. Moreover, the advancement of context-logging mobile applications has furnished a mechanism for continuously monitoring and tracking a user's contextual information, which encompasses location, activities, weather, and other pertinent factors, in real-time. Existing research has already illustrated the importance of this contextual information in comprehending and detecting stressful events experienced by individuals [161].

Real-time monitoring of physiological signals and contextual data presents a formidable challenge. The acquisition of physiological signals via smartwatches and wearable devices is particularly prone to motion artifact noise [162], necessitating extensive filtering and processing to enable their use in stress detection algorithms within daily life settings. Moreover, developing a daily life stress monitoring system mandates access to real-time stress level labels from participants, a task that poses several challenges [163]. The timing of label querying is critical, requiring careful selection of moments when participants are not engaged in activities such as sleeping, studying, or working, to ensure optimal participation and reliable labels. Additionally, capturing the moments most conducive to experiencing stressful situations is crucial [163]. Notwithstanding these obstacles, designing a robust and accurate system that captures both physiological and contextual information in real-time while querying labels from participants is an even greater challenge.

In this study, we present a context-aware daily life stress monitoring system that leverages physiological and contextual data and incorporates a smart label querying method. The system utilizes a publicly accessible life-logging mobile application to gather real-time contextual data from participants. The task of simultaneously collecting both contextual information and physiological signals while querying for stress labels in daily life presents significant challenges. To address these challenges, we propose a three-tier Internet of Things (IoT) based system architecture for our real-time monitoring system. In summary, the key contributions of this paper are as follows:

- Propose a three-tier IoT-based system architecture to efficiently collect and record both physiological and contextual data alongside labels throughout the day.
- Implement a smart EMA triggering-based system to capture sufficient and high-quality labels multiple times daily.
- Investigate the impact of personalization on stress detection by examining how the

performance of our algorithm improves with more subject-specific data available in the training phase.

5.2 Related Works

This section presents an overview of the related works as summarized in Table 6.1. The majority of the existing research works in stress detection are conducted in laboratory settings or controlled environments [164–166]. In these studies, participants are typically required to wear wearable devices while engaging in a sequence of experimental tasks, such as viewing a series of images or videos or being exposed to stressful activities. During the study, various kinds of bio-signals, such as PPG, Electrocardiogram (ECG), Electrodermal Activity (EDA), and Electroencephalogram (EEG) are recorded and employed for building models for stress detection. Despite the remarkable performance obtained by these controlled experimental methods, such algorithms are not feasible for usage in real-world stress detection systems. The data gathered in daily life is susceptible to contextual confounders and motion artifact noise resulting from movements and routine activities. Moreover, the type of stress encountered in daily life scenarios can substantially differ from that induced in the controlled laboratory setting [163].

Recent advancements in stress detection methods have involved using physiological signals collected in real-world settings [167–171]. This is achieved through the use of wearable devices, such as smartwatches or smart wristbands, which continuously collect physiological data from participants. Multiple questionnaires are sent randomly throughout the day to gather information on stress levels. Finally, machine learning techniques and statistical algorithms are applied to the collected data to build a stress model. A disadvantage of these studies is the absence of contextual information in their stress models, which can result in less reliable stress detection algorithms. The importance of contextual data in stress detection tasks has already been extensively demonstrated in the literature [172].

Can *et al.* [171] propose an objective stress detection system that uses smart bands and contextual information, such as weather information and activity type (e.g., lecture, presentation, or relaxation). However, one of the major limitations of this study is its semi-controlled setting. In their study, the data was obtained during an eight-day training event, where all the participants followed a predetermined schedule, including designated training days, free days, midterm presentations, and other similar activities. Consequentially, the contextual data captured in this study was captured manually and is limited to the time and date of the predetermined schedule.

Only a limited number of studies have investigated the integration of physiological signals and contextual information in a non-controlled, real-world setting [173, 174]. However, the issue with these studies is the infrequency of their survey administration (i.e., once a day), as stress levels can vary greatly throughout the course of a day in response to various daily life events. These models cannot be properly coupled with mHealth-based just-in-time interventions due to their lack of assessing stress instantaneously. Increased frequency of survey administration would improve the likelihood of capturing fine-grained stress-inducing moments. Additionally, these studies lack a smart query system to capture the labels, which can result in more missing labels, such as instances where wearable devices are not being worn or carried. Our work aims to address these shortcomings using an objective and automatic physiological and contextual data collection approach focusing on fine-grained stress detection.

Table 5.1: Related Works

Study	**EMA Frequency**	**Smart Query**	**Physiology**	**Context**	**Daily Life**
Jeong Han et al. [164]	N/A	✗	✓ (PPG, ECG, SC)	✗	✗
Cho et al. [166]	N/A	✗	✓ (ECG)	✗	✗
Wang et al. [172]	every 3 months	✗	✗	✓	✓
Yu et al. [167]	10/day	✗	✓ (ECG, SC, ST, Motion)	✗	✓
Sah et al. [168]	4/day	✗	✓ (PPG, SC, Motion, ST)	✗	✓
Tazarv et al. [169]	distribution based	✓	✓ (PPG)	✗	✓
Battalio et al. [170]	1/day	✗	✓ (ECG, Motion, Resp)	✗	✓
Can et al. [171]	1/day	✗	✓ (SC, PPG, ST, Motion)	✗	✗
Yu et al. [173]	1/day	✗	✓ (SC, ST, Motion)	✓	✓
Mundnich et al. [174]	1/day	✗	✓ (ECG, Motion)	✓	✓
This Work	7/day	✓	✓ (PPG, Motion)	✓	✓

5.3 Methods

5.3.1 Study

Starting in November of 2021, we recruited a sample of college students (n = 11) from the University of California, Irvine, via flyers and faculty outreach. The participants, comprising both male (n = 4) and female (n = 7) populations, ranged in age from 18 to 37 years (Mean = 22.91, SD = 5.05). The students were enrolled on a rolling basis at different intervals, depending on their enrollment date, and participated for a total of 2 weeks. During the enrollment process, participants review our study information document and are asked afterward if they agree to continue their participation. Consent is obtained verbally and is then documented in an Excel file by the research team member running the participant session. As a component of the enrollment process, students were instructed to download 2 mobile applications (one foreground app to provide EMAs and one background app to perform passive mobile logging) and were equipped with a smartwatch. Throughout the

2-week period, while wearing the smartwatch that continuously measured physiology and activity levels, students were prompted to complete multiple daily EMAs that was triggered by a smart EMA query system.

The experimental procedures involving human subjects described in this paper were approved by the Institutional Review Board (IRB) at the University of California, Irvine.

5.3.2 System Architecture

The architecture of our proposed system is shown in Figure 5.1. The system comprises three primary layers that facilitate the collection of physiological and movement data, capturing contextual data, and querying labels.

Sensor Layer

This study uses Samsung Galaxy Gear Sport Watches as the wearable device. This smartwatch is equipped with sensors capable of recording PPG (20Hz), accelerometer, and gyroscope (movement) signals. We designed a custom smartwatch application for Samsung Galaxy Gear Sport Watches running on the Tizen operating system to gather these unprocessed PPG and movement signals. The data collected by the watch is transferred to the cloud layer when it is connected to a local Wi-Fi network, and in the absence of such a network, the data is transmitted via Bluetooth to a smartphone. Two services and a user interface (UI) are included in the raw signal acquisition program. The initial service delivers sensor data to the cloud at intervals of two minutes which take place every fifteen minutes.

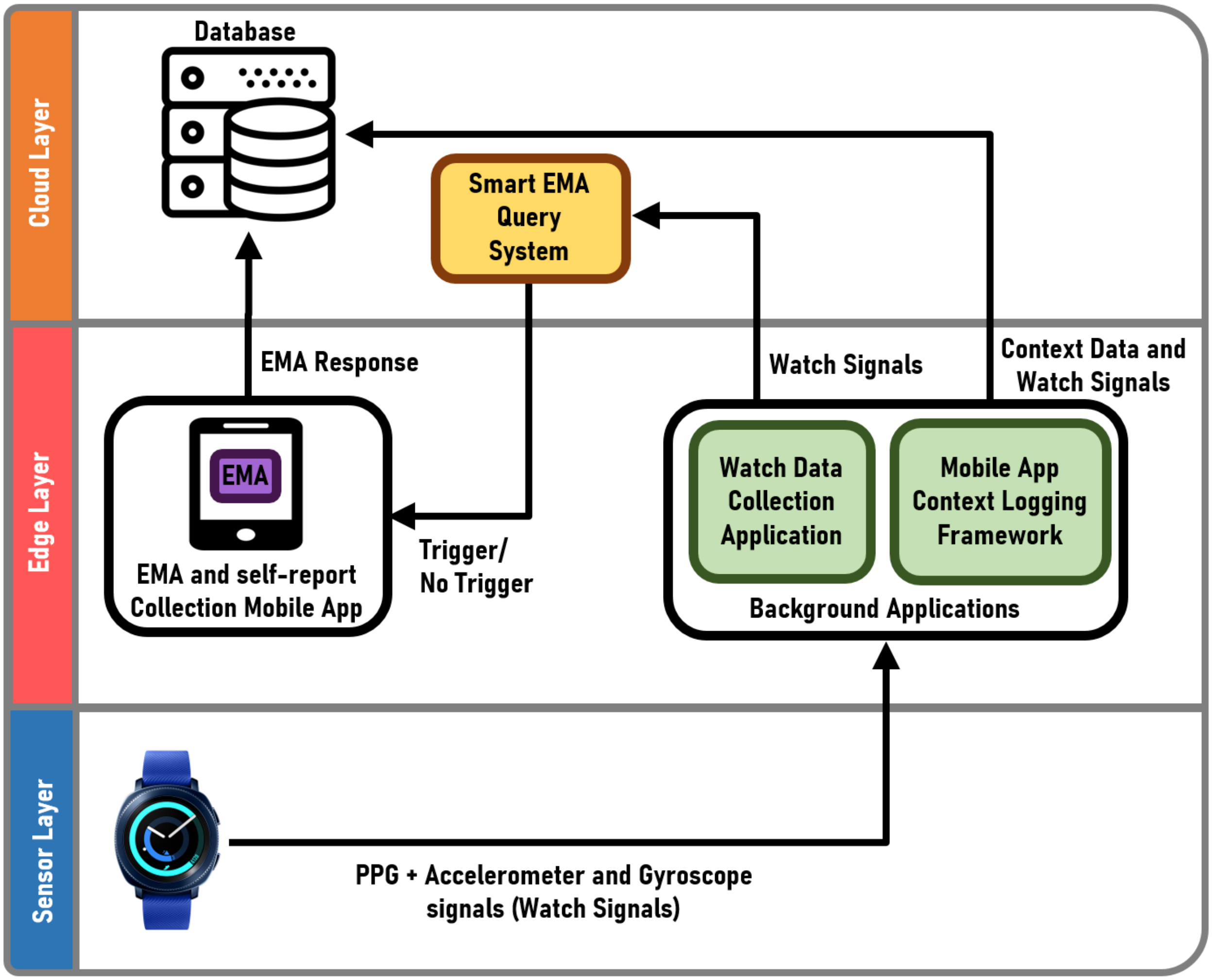

Cloud Layer
Edge Layer
Sensor Layer
Database
Smart EMA Query System
EMA Response
EMA
EMA and self-report Collection Mobile App
Trigger/ No Trigger
Watch Signals
Context Data and Watch Signals
Watch Data Collection Application
Mobile App Context Logging Framework
Background Applications
PPG + Accelerometer and Gyroscope signals (Watch Signals)

Figure 5.1: Proposed System Architecture

Edge Layer

We use the AWARE framework [175] to capture contextual data in everyday settings. AWARE is an open-source mobile instrumentation framework for logging, sharing, and reusing mobile context. AWARE uses smartphone built-in sensors to capture daily life logging information such as phone battery level, weather, location, screen status, etc. In the event that the Wi-Fi network is inaccessible, an alternative smartphone application installed on the edge is employed to gather the raw PPG signals and accelerometer data from the sensor layer via Bluetooth and then transmit the data to the cloud for storage. To elicit stress level labels from the study participants, a supplementary smartphone application has been developed which employs an EMA to solicit stress level ratings from the participants.

Cloud Layer

A smart EMA query system is implemented (S-EMA) on the cloud to query labels throughout the day. The followings are the summary of the main rules used by the S-EMA module to trigger EMAs:

- Sending EMAs only between 7 AM and midnight.
- Sending EMAs only when the user is wearing the watch using the accelerometer data.
- Sending EMAs only when the collected data is recent (the watch may record the data without the Internet and sync it later).
- It is intended to query labels seven times per day. The frequency of querying labels is adjusted dynamically to ensure that approximately seven labels are captured daily. The waiting period is calculated based on the initial wear time of the watch to achieve this target.

The stress levels in the EMAs are listed as “not at all” (1), “a little bit” (2), “some” (3), “a lot” (4), and “extremely” (5).

Contextual data collected from the AWARE Framework (at the Edge layer), labels queried through EMAs (the Edge layer), and raw physiological (PPG) and movement (accelerometer, gyroscope, and gravity) signals captured by wearable platforms (the Sensory layer) are sent and stored in the cloud for cleaning, filtering, preprocessing, and being utilized in the predictive models.

5.3.3 Preprocessing

The PPG signals stored in the cloud layer are collected from wearable devices and hence are prone to noise. To mitigate this issue, a number of preprocessing and filtering techniques are applied to the raw PPG signals in order to prepare them for further analysis. To detect stress as a stimulus in human subjects, a variety of features are extracted from these signals, such as heart rate, heart rate variability, and breathing rate, to name a few. Raw contextual data collected from the AWARE framework are also too broad and non-informative. A feature extraction module is designed with the purpose of transforming the raw contextual logging data into informative contextual life-logging features. These extracted features serve as inputs for our predictive machine-learning models. Data Imputation and Feature Selection are two postprocessing techniques employed to improve our models' performance.

Data Cleaning and Windowing

In order to clean the collected PPG data, first, we apply a Butterworth band-pass filter of order 3, with cut-off frequencies at 0.7Hz and 3.5 Hz. Then, a moving average across a 1-second window is used to smooth the data and reduce the artifacts such as body gestures and movements, which are common in everyday settings. These clean PPG signals, alongside contextual data collected from the AWARE framework, are resampled to 15-minute timing windows. Each of these 15-minute time frames, which consists of 2-minute continuous photoplethysmography (PPG) signals and context data collected through the AWARE framework, is then processed by the feature extraction module.

Feature Extraction

In the feature extraction module, we use the HeartPy library [176] to process the clean PPG signals to extract PPG peaks and PPG-relevant features including heart rate variability. The HeartPy is a Python Heart Rate Analysis Toolkit. The toolkit is designed to handle (noisy) PPG data. Using this library, the following 12 features are extracted from the PPG signals: BPM, IBI, SDNN, SDSD, RMSSD, PNN20, PNN50, HR_mad, SD1, SD2, S, and BR. Table 6.3 outlines the definitions of these features for reference.

Table 5.2: PPG Features

Feature	**Definition**
BPM	Beats per minute, Heart Rate
IBI	Inter-Beat Interval, the average time interval between two successive heartbeats (NN intervals)
SDNN	Standard deviation of NN intervals
SDSD	Standard deviation of successive differences between adjacent NNs
RMSSD	Root mean square of successive differences between the adjacent NNs
PNN20	The proportion of successive NNs greater than 20ms
PNN50	The proportion of successive NNs greater than 50ms
HR_mad	Median absolute deviation of NN intervals
SD1 and SD2	Standard deviations of the corresponding Poincare plot
S	Area of ellipse described by SD1 and SD2
BR	The number of breaths per minute (breathing rate)

The raw contextual data captured from AWARE framework are presented in Table 6.2. These raw contextual data are not immediately usable in our predictive models and thus require further processing. To address this issue, we have implemented a feature extraction module to translate the raw data into numerical features that can be utilized by the models.

In Table 6.2, the “Values” column details the range, type, and units for each raw feature. The “Cut-offs” column lists the threshold values utilized to convert the raw features into a more comprehensible and abstract numeric format suitable for usage in the predictive models. For example, the cut-off values for battery_level are 10%, 25%, and 50%. Therefore, for battery_level (BL) values $BL \leq 10$, $10 < BL \leq 25$, $25 < BL \leq 50$, and $BL > 50$, we

Table 5.3: Aware Features

Feature	**Definition**	**Values**	**Cut-offs**
battery_adaptor	Indicator of power source	0=No source, 1=AC, 2=Dock, 3=USB	-
battery_level	Battery percentage	(0:100]%	[10, 25, 50]
speed	Movement speed of user	Double value in m/s unit	[0, 1, 5]
device_off	Device not used duration	Double value in minutes unit	[2, 10, 20, 60, 180, 540]
device_on	Device is being used duration	Double value in minutes unit	[2, 10, 20]
air_pressure	The ambient air pressure	Double value in mbar/hPa unit	[900, 1000, 1100]
weather_temperature	Measured temperature	Double value in Celsius unit	[5, 10, 20, 30]
weather	Weather forecast (API)	Weather forecast in text, ex. 'Clear'	-
wind_degrees	Degree of wind	Double value in degree	[45, 90, 135]
wind_speed	Speed of wind	Double value in m/s unit	[0, 2, 5, 10]
screen_status	Status of phone screen	0=off, 1=on, 2=locked, 3=unlocked	-
location	Longitude, Latitude, Altitude	Three double values	-

have respectively assigned the following numerical values: 0, 1, 2, 3. For battery_adaptor and screen_status features, there are no cut-off values, and their raw values are used in the model. For weather we used the following mapping function from text to the numerical values: {'clear': 0, 'mist': 1, 'clouds': 2, 'rain': 3, 'snow': 4}. In terms of location, since all participants are students at the University of California, Irvine (UCI), we abstracted and categorized their position into four distinct areas at the edge layer to preserve their privacy. These areas are defined as follows: 0: within the UCI recreation center (UCI ARC), 1: within the university premises (for work/study), 2: within UCI housing, and 3: outside of the aforementioned locations. To assign a location to each participant, we established circular boundaries encompassing each of these areas. Once the Longitude, Latitude, and Altitude of the participant were determined, we checked whether their location lay within one of the circular boundaries. If it did, we would assign the corresponding numeric value; otherwise, the outside numeric value (3) would be assigned.

Data Labeling

The EMA protocol is designed to trigger a maximum of seven times per day and prompt participants to indicate their stress level on a five-point Likert scale: (1) not at all, (2) a little bit, (3) some, (4) a lot, and (5) extremely. The stress level reports, along with the corresponding timestamps, are recorded in the cloud for subsequent analysis. Each 15-minute timing window of collected physiological and contextual data is then labeled based on the closest subsequent EMA query. The label distribution is shown in Figure 5.2.

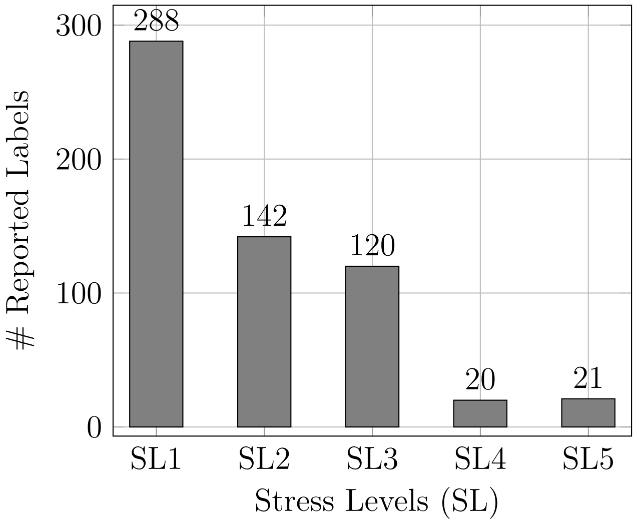


Figure 5.2: The distribution of reported stress levels

Data Imputation

The contextual features obtained from the AWARE framework are sourced from various sensors integrated within smartphones. As a result, certain 15-minute timing windows may exhibit missing data for some features, which can occur due to differences in the frequency of the sensors or technical limitations of a specific sensor. In order to optimize the construction of efficient machine learning models, it is recommended to utilize a data imputation technique rather than solely discarding incomplete data. This ensures the utilization of all available features. To this end, we use the k-nearest neighbor's imputation algorithm [177] in order to compensate for missing contextual values in our data. The method predicts the values of any additional data points using "feature similarity." In other words, the value given to the new point depends on how much it resembles the points in the training set. Employing a technique that identifies the k-nearest neighbors to the observation with missing data, and subsequently imputes them based on the non-missing values in the vicinity, can be an advantageous approach for predicting absent values. This technique creates a rudimentary mean impute and then utilizes the resulting complete dataset to construct a KDTree. The KDTree is then utilized to compute the nearest neighbors (NN). Following the determination of k-NNs, it calculates their weighted average. This algorithm is deemed to be more accurate than the commonly utilized imputation methods such as mean, median, and mode, as it incorporates the similarity between the features into consideration.

5.3.4 Stress Detection

We evaluate the efficacy of our proposed stress detection system through binary classification. In this classification, instances of “no stress” (represented by a stress level of 1) are assigned a value of 0, while instances of “a little bit,” “some,” “a lot,” and “extremely” (represented by stress levels greater than or equal to 2) are assigned a value of 1.

The main reason for classifying the samples into “stress” and “no stress” is to have a more balanced distribution of labels since some classes such as “extremely” and “a lot” are rare. As shown in Figure 5.2, with this categorization, there will be 288 samples with label 0 (“no stress”) and 303 samples with label 1 “stress”.

Machine Learning Algorithms

To build a stress detection algorithm, we used machine learning-based methods. Three classification techniques were employed, namely the K-Nearest-Neighbor [178] with k values ranging from 1 to 20, Random Forest [179] with a depth range of 1 to 10, and the XGBoost classifier [180]. The K-Nearest-Neighbor technique predicts the outcome based on a majority vote using the k number of closest data points. The Random Forest classifier is an ensemble learning approach that employs averaging to increase predictive accuracy and reduce overfitting. It fits a number of decision tree classifiers to different subsamples of the dataset. The XGBoost is an efficacious open-source implementation of the gradient-boosted trees algorithm. Gradient boosting is a supervised learning process that combines the predictions of a number of weaker, simpler models to predict a target variable properly.

Feature Selection

In this work, feature selection constitutes a pivotal stage, and its inclusion can significantly enhance the performance of our model. This is largely due to the inherent constraints associated with the AWARE features. Specifically, some features may exhibit consistent values over time, thereby rendering them less reliable and less critical for classification. Additionally, certain features may present significant quantities of missing data due to the challenges encountered in the collection of contextual data, such as sensor malfunction. Furthermore, the filtering and cleaning of the PPG signals, necessary to eliminate motion artifact noise, may result in the loss of critical information from the signals. As a consequence, the extraction of features from PPG signals may be rendered less dependable. It is, therefore, imperative to identify and eliminate these aforementioned features from the model to mitigate their adverse effects on the model's efficacy.

Given that a tree-based machine learning classification algorithm is employed in our prediction models, we have elected to adopt a tree-based feature selection algorithm. Random forest classifiers offer the mean decrease impurity and mean decrease accuracy feature selection approaches. In this study, we chose features using a mean decrease impurity technique. Gini importance is another name for mean decrease impurity. Random forest uses numerous different decision trees. Every node in the decision tree represents a condition on one of the qualities, and it is a model of decisions that resembles a tree. These nodes divide the data into two sets, with the goal of having the data with the same labels end up in the same set in the best case. The criterion used to determine the best condition for each node is impurity. The total decrease in node impurity averaged over all ensemble trees is what is meant by "mean decrease impurity" for each feature. This metric is used to order the features.

Performance Metrics

In order to evaluate the performance of our stress monitoring system, we use F1-score as a quality metric. The F1-score is a measurement of a test's accuracy used in statistical analyses of binary categorization. It is derived from the test's precision, and recall, where precision is the proportion of "true positive" results to "all positive results," including those incorrectly identified as positive, and recall is the proportion of "true positive" results to "all samples that should have been identified as positive." In diagnostic binary classification, recall is also referred to as sensitivity, while precision is also referred to as positive predictive value. The F1-score is calculated as the weighted average of precision and recall:

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

5.4 Results

In our research, a cross-validation technique [181] was utilized to evaluate the performance of our classification models. Cross-validation is a widely employed algorithm for accurately estimating the performance of a machine-learning model on unknown data. The process involves training a model using different subsets of the data and then testing the average accuracy of the remaining data. To assess the effectiveness of our research findings, we employed a 5-fold cross-validation method. To ensure that there is no overlap of user data in the train and test splits, the splits were created based on user IDs.

In order to ensure objectivity and prevent any potential biases, we adopted a fresh start approach for each iteration of the stress detection model. We disregarded any prior knowledge or information from previous stress models or the data from the current test users. The ultimate performance of the model was computed by calculating the mean of the individual stress models' performances generated.

The summary of the performance achieved for our stress assessment algorithm utilizing solely PPG data with a 5-fold cross-validation technique is presented in Table 5.4. The table shows that KNN with k=7 exhibited the best performance with an F1-score of 56. However, the result of 56 is not promising for binary classification.

As a subsequent measure, we resolved to augment our model with contextual information to enhance its performance. The results of the stress assessment algorithm that incorporates both PPG and contextual data are summarized in Table 5.5. This assessment was carried out

Table 5.4: Validation accuracy of our stress assessment algorithm using only PPG data

Classifiers	**F1**	**Selected Features**
Random Forest (depth=9)	52	SD1, IBI, S, SD2, BPM, SDSD
KNN (k=7)	56	SD1, IBI, S, SD2, BPM, SDSD
XGBoost	51	S, SD2, BPM, SDSD

using a 5-fold cross-validation technique. According to the table, the Random Forest model with a depth of 5, employing the top five features chosen by the GINI index algorithm, attained the best performance. This outcome indicates a 14% increase in performance, underscoring the noteworthy role of contextual data in stress detection techniques.

Table 5.5: Validation accuracy of our stress assessment algorithm using both PPG and contextual data

Classifiers	**F1**	**Selected Contextual Features**	**Selected PPG Features**
Random Forest (d=3)	70	weather, wind_speed, device_off, location	BPM
KNN (k=9)	62	weather, wind_speed, device_off, location, speed	BPM, S, SD2, SDSD, SDNN, BR
XGBoost	64	weather, wind_speed, device_off, location, speed	BPM, S, SD2, SDSD, SDNN, BR, IBI

The most important contextual features, as determined by our analysis, are weather, wind_speed, device_off, and location. Regarding the PPG signal, beat per minute (BPM) was identified as the most relevant feature for stress detection. However, the performance of the KNN and XGBoost classifiers was found to be lower. For these two classifiers, the top 11 and 12 features were selected, respectively.

5.4.1 Explainability of the Model

This section employs the stress detection model with the most optimal performance, which is the Random Forest model detailed in Table 5.5. This model employs the five most important features, namely bpm, weather, wind_speed, device_off, and location, for the predictions.

To explain how our machine learning model predicts stress in terms of extracted features, we use the SHAP method [182]. SHAP (SHapley Additive exPlanations) is a game theoretic approach to explain the output of any machine learning model. It utilizes the traditional Shapley values from game theory and their related extensions to correlate optimal credit allocation with local explanations.

The fundamental concept of Shapley value-based interpretations of machine learning models is to allocate credit for a model's output among its input features. While computing SHAP values can be quite intricate, as they are generally NP-hard, this is not the case for linear models which are more straightforward. In such cases, we can extract the SHAP values directly from a partial dependence plot. Having a prediction $f(x_i)$, the SHAP value for a particular feature x_i is the difference between the anticipated model output and the partial dependence plot at the feature's value x_i.

We apply SHAP method to our proposed stress detection model with the best performance, which is the Random Forest model presented in Table 5.5. Figure 5.3 shows the bar plot providing the absolute SHAP values calculated for each feature. This bar plot takes the mean absolute value of each feature over all the instances (rows) of the dataset (test data). According to this Figure, the BPM and location have the lowest impact on the model compared to the other features. The device_off feature, which denotes the duration the phone is not in use, is a contextual feature that has the greatest impact on the model's outcome. Subsequently, the weather and wind_speed features exhibit the highest influence on the model output, with a mean absolute SHAP value of approximately 0.15.

To observe the impact of each feature on the model's output based on the feature values, we employ the beeswarm plot. Figure 5.4 presents a beeswarm plot that summarizes the complete distribution of SHAP values for each feature. Utilizing SHAP values, this plot showcases the effects of each feature on the model output. Features are sorted by the total sum of SHAP value magnitudes over all samples. The color of the plot demonstrates the

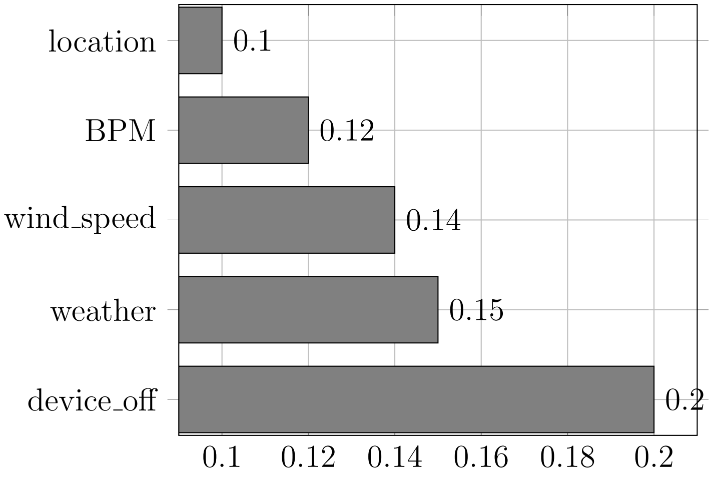


Figure 5.3: SHAP mean absolute values

feature value, with red indicating high and blue indicating low. This analysis shows that a high value for the device_off feature (more time the device is not in use) results in a lower predicted stress value. As expected, a higher BPM increases the predicted stress value. For the weather feature, our findings suggest that when the weather condition is in the mid-range, such as mist, clouds, or rain, it increases the probability of stress, whereas when the weather condition is clear, it reduces the likelihood of stress. Furthermore, higher wind speed also increases the predicted stress value.

One notable finding that has been made here is that a lower value for the location feature indicates a higher predicted stress level while higher values indicate lower predicted stress levels. According to our designated ranges for the location feature, a lower value corresponds to the presence inside the university premises. On the other hand, higher values of the location feature correspond to UCI housing or an outdoor location, which results in decreased predicted stress levels. This observation suggests that the location of an individual can play a significant role in their stress levels, with certain locations associated with higher levels of stress.

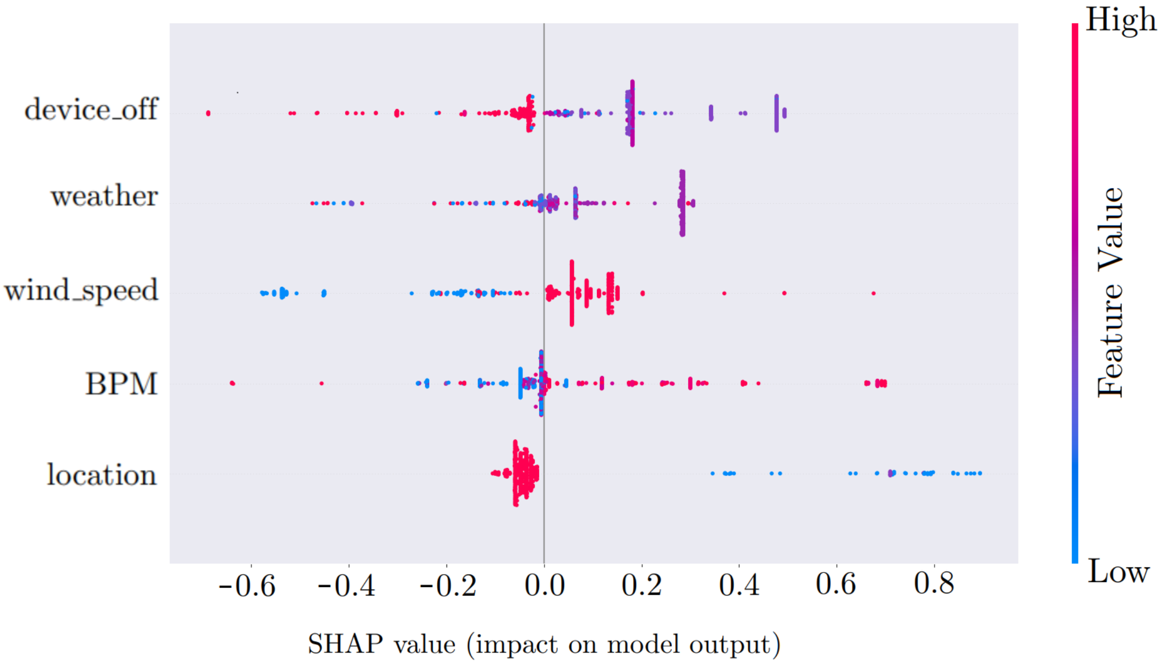


Figure 5.4: Feature impacts on the model output

5.4.2 Personalization

In order to evaluate the impact of personalization on our stress detection algorithm, we conducted an experiment using data from three subjects with substantial amounts of data (S111, S912, and S731). To achieve this, we trained our model on data from all other subjects in the first stage, and then tested it on half of the data from one of the selected subjects (e.g., S111). In the second stage, we customized the model using the second half of the S111 data for training (in addition to data from other subjects) and the first half of the S111 data for testing. We utilized the Random Forest classifier to demonstrate the performance changes. Table 5.6 shows the results, indicating that personalization improves the prediction performance by approximately 10%. All the extracted PPG and contextual features were used in our models for this experiment.

Table 5.6: F1-score before and after personalization

User ID	F1 (Before)	F1 (After)	Selected Features
S111	74.1	81.0	Random Forest (depth=5)
S912	76.0	83.0	Random Forest (depth=5)
S731	36.7	47.1	Random Forest (depth=19)

5.5 Discussion

Capturing real-time features and signals while collecting labels from participants in daily life is challenging. We propose a real-time monitoring system using a three-level architecture. It includes a sensor layer with a Samsung Gear Sport watch 2, an edge layer with mobile apps using the AWARE framework, and a cloud layer to store data securely in a database.

Our real-time multi-tier system architecture was able to achieve an F1-score of 70% for the task of stress detection. In addition, we have shown that personalization has a positive impact on our stress detection models, resulting in approximately a 10% improvement. This observation suggests that having a personalized model for the participants could result in improved performance for stress prediction models.

Despite the successful implementation of a multi-tier system, one limitation of our work is the occasional absence of contextual data for certain timing windows. The contextual data is captured by the AWARE framework through different phone sensors, services, and APIs. Therefore, the limitations and potential inaccuracies of these sensors, services, and APIs, may result in missing features for some of the captured context data.

The second limitation of our research is associated with our label query system. Specifically, our labeling system functions on a time-event paradigm, wherein it solicits EMAs from study participants at pre-determined intervals of T hours. As a result, there exists a possibility that our system may not trigger an EMA request during instances when an individual is undergoing stress. Conversely, the system may initiate an EMA during moments when participants are resting or occupied with work, leading to an unsatisfactory experience and subsequently, an increase in missing EMA submissions. In future work, we intend to implement a smarter query system to overcome the mentioned challenges with the purpose of (1) accurately identifying the time frames during which the participant experiences stress and (2) not resting or engaged in work or activities.

To efficiently identify these timing windows, it is imperative to establish a real-time data processing system in the cloud, which can receive and process data from the edge layer in real-time. By utilizing the processed physiological signals and contextual features, it becomes possible to detect circumstances that may result in stressful timing windows. The accelerometer signals obtained from the Samsung Gear Sport Watch can be utilized to construct a machine-learning approach for identifying various daily activities, such as sleeping, walking, and sitting. Daily-life activities alongside context features such as location could help us to build a smart high-level context recognition system detecting the most efficient timing windows to send the EMAs.

5.6 Conclusion

In this work, we proposed a context-aware daily-life stress monitoring system using physiological and smartphone data. A smart query module, which uses accelerometer signals collected from a watch, is implemented in order to capture sufficient and high-quality labels. To the best of our knowledge, this is the first work presenting a daily-life stress monitoring system employing both physiology and context data with a smart query system to capture a sufficient number of EMAs throughout the day. According to our results, we were able to achieve an F1-score of up to 70% using a Random Forest classifier.

Chapter 6

Incorporating Personalization to Elevate User Experience and Engagement

In today's fast-paced world, accurately monitoring stress levels is crucial. Sensor-based stress monitoring systems often require large datasets to train effective models. However, individual-specific models are essential for personalized and interactive scenarios. Traditional methods like Ecological Momentary Assessments (EMAs) can assess stress but often struggle with efficient data collection without burdening users.

The challenge lies in sending EMAs at the right moments, especially during periods of stress, balancing monitoring efficiency and user convenience. In this chapter, we introduce a novel context-aware active reinforcement learning (RL) algorithm for enhanced affect detection using Photoplethysmography (PPG) data from smartwatches and contextual data from smartphones. Our approach dynamically selects optimal times for deploying EMAs, utilizing the user's immediate context to maximize label accuracy and minimize intrusiveness. We aim to detect stress as a case study, which is one of the main types of affect.

Initially, our study was conducted in an offline environment to refine the label collection process, aiming to increase accuracy while reducing user burden. We later integrated a real-time label collection mechanism, transitioning to an online methodology, which resulted in an 11% improvement in stress detection efficiency. Incorporating contextual data further improved model accuracy by 4%. Personalization studies showed a 10% enhancement in AUC-ROC scores, indicating better differentiation of stress levels.

This chapter represents a significant advancement towards personalized, context-driven, real-time stress monitoring within our proposed closed-loop real-time system architecture.

6.1 Introduction

As per data from the American Institute of Stress [183], approximately 55% of individuals in the United States encounter stress throughout their day. The American population stands as one of the most stressed globally, with their current stress levels surpassing the global average by 20 percentage points. The impact of stress extends to the physical body, cognitive processes, emotions, and behavior [184]. Unaddressed stress can contribute to several health issues, including high blood pressure, heart ailments, obesity, and diabetes [184]. Consequently, daily life monitoring of stress has garnered significant significance within our society, and the advancement of techniques for diagnosing human stress holds paramount importance.

The presence of stress within the human body can be diagnosed by analyzing psychophysiological signals such as Photoplethysmography (PPG) [185]. PPG is a simple optical sensing technique for detecting blood volume alterations within peripheral circulation [97]. With the rapid advancements in technology and the development of the Internet of Things (IoT)

[25, 186–188], the acquisition of PPG signals has been greatly facilitated, primarily through the utilization of wearable devices like smart rings or smartwatches [189]. Consequently, monitoring stress levels in everyday life becomes an attainable endeavor by analyzing the PPG signals garnered from the aforementioned wearable devices. Furthermore, the evolution of mobile apps designed for context logging has provided a means to consistently observe and record a user's contextual data, including elements like their location, activities, weather conditions, and other relevant variables, all in real-time [190]. Prior studies have already demonstrated the significance of this contextual information in understanding and identifying stressful experiences encountered by individuals [68, 191, 192]. In everyday situations, biosignals vary widely among individuals due to physiological and lifestyle differences, as well as the diverse activities one might partake in. Additionally, the perception of stress levels differs from person to person, leading to biases in the data collected. Therefore, there is a significant need to tailor predictive models to each individual across their various activities. These adjustments, which are essential at the time of deployment, present both conceptual and technical challenges.

The evolution of stress monitoring in daily settings has seen a significant transformation. Originally, stress monitoring was largely confined to controlled environments such as laboratories, which allowed researchers to closely observe and study physiological responses under stress [193, 194]. Such controlled studies laid the groundwork for understanding stress responses in a structured setting. Over time, advancements in technology enabled the transition to more naturalistic and dynamic environments. This shift paved the way for methods like Ecological Momentary Assessments (EMAs) [195]. EMAs revolutionized stress monitoring by allowing real-time data collection about participants' stress levels throughout their day-to-day activities. Using self-reported stress (EMA) as the labeling source, users are asked to respond to real-time queries that link the data gathered by sensors to stress labels in everyday situations. One of the main challenges is optimally collecting these labels (stress levels) from individuals in their daily lives [196]. Frequently triggering EMAs or sending them at inappropriate times, such as when a user is busy with work or sleeping, could burden the user. This could lead to a significantly lower number of reported labels. Moreover, selecting the optimal moments to send the EMAs, especially during instances when an individual is experiencing stress, poses a considerable challenge and holds the utmost importance.

To tackle these challenges, in this work, we introduced a contextual variant of active learning, based on Deep Q-Learning, which incorporates the contextual information pertaining to an individual into the decision-making process. In the initial phase of our research, a context-aware active reinforcement learning algorithm was utilized in an offline setting [197]. This approach was implemented to thoroughly evaluate the effectiveness of our proposed method. We demonstrated that the utilization of such an algorithm in a stress detection task can lead to a reduction of up to 88% in the required EMAs when compared to a random selection approach, and up to 32% when compared to traditional active learning methods. Furthermore, we observed that employing such an algorithm can increase the performance of stress detection tasks by up to 21% compared to a random selection method, and up to 8% when compared to traditional active learning approaches. However, an offline context-aware active reinforcement learning algorithm abstains from employing active learning to initiate EMAs. However, this approach may still entail user burdens and result in triggering EMAs at inappropriate times.

In this article, an extension of our previous work [197], we have improved our proposed algorithm for application in an online setting, leveraging active learning to initiate EMAs. In the online setting, our algorithm initiates EMAs at various points during the study, guided by the contextual information pertaining to the user. Within the context of obtaining participant labels such as stress levels, the active learning algorithm analyzes contextual information in real-time to ascertain the most appropriate timing for posing questions. This adaptive approach serves to reduce participant burden while simultaneously enhancing label accuracy. To comprehensively assess the efficacy of our online algorithm, we compare it to the prior offline one, we conducted two distinct analyses on the same dataset: one employing the offline context-aware active reinforcement learning algorithm and the other utilizing the online variant. Our findings unequivocally demonstrate that the online algorithm yields a substantial enhancement in the performance of the stress detection task when contrasted with its offline counterpart. Lastly, we employ a personalization technique to investigate the effects of personalized customization in enhancing the model's performance.

In summary, the key contributions of this work are as follows:

- Propose a new form of active learning, utilizing Deep Q-learning, aimed at enhancing interaction with the monitored individual during data collection.
- Develop a sensor-edge-cloud layered system architecture for the acquisition and labeling of the data aimed at real-time stress detection. We further demonstrate the effectiveness of our proposed system through the utilization of actual real-time data and comparing it with an identical offline variant of the algorithm.
- Incorporate the contextual features into the stress detection models for the purpose of systematically monitoring the influence of contextual factors within the context of stress detection.
- Examine how the performance of our algorithm is enhanced with the inclusion of subject-specific data during the training phase in order to explore the influence of personalization on stress detection.
- Conduct a two-stage IRB-approved study on 54 individuals across undergraduate and graduate student populations over two periods: June 2020 to June 2021 (offline method) and March 2022 to May 2023 (online method), generating a total of 132,598 filtered samples.

This chapter is structured as follows: Section 2 offers a comprehensive review of stress assessment methods and associated research, highlighting the importance of personalizing models and the crucial role of user behavior and contextual data in real-time labeling. It also emphasizes the need to enhance user engagement in everyday settings. Section 3 details the platform developed for data collection and analysis. Section 4 outlines the dataset we gathered and our data processing methods. Section 5 introduces our proposed context-aware active learning approach, which incorporates user behavior and context into its query mechanism, along with temporal data correlations, to enhance query scheduling. Section 6 reports on the outcomes of various querying techniques in relation to personalization. Finally, Section 7 concludes the chapter and includes a discussion.

6.2 Related Work

Stress-related research often delineates its origins from both exogenous factors—such as lifestyle, interpersonal relationships, and financial stability—and endogenous factors like individual psychological constitution and thought processes. These factors act as progenitors for negative affective states, including anxiety and fear, and instigate corresponding physiological responses. The physiological aspect of stress, denoted as a stress response, is a series of bodily reactions to environmental stimuli or stressors. Within the scientific discourse, the construct of stress is categorized into psychological, behavioral, and physiological dimensions. Historically, self-report measures such as the Perceived Stress Scale (PSS), formulated by Cohen et al. [198], and the stress inventory by Holmes and Rahe [199], have been the standard for gauging stress levels retrospectively.

Nonetheless, the accuracy of survey-based assessments of stress is compromised by measurement biases, including response bias, which reflects the influence of the query's framing on the participant's responses. Additionally, while some behavioral expressions of stress—like facial expressions—are spontaneous, they may also be subject to volitional control, thus potentially skewing data accuracy. Consequently, recordings of such behaviors must be critically examined for systematic errors that may misrepresent the actual stress magnitude.

Given these constraints, and paralleling the evolution of high-precision sensor technology, there is an augmented demand for veritable detectors of physiological stress markers. Biosignal attributes of stress episodes are typically involuntary, and such data can be acquired through methodologies like electrocardiography (ECG), photoplethysmography (PPG), electromyography (EMG), skin conductance (SC) or electrodermal activity (EDA), respiratory rate (RSP), skin temperature (ST), pupil dilation (PD), and cerebral activity as captured by electroencephalography (EEG) [200].

The current methodologies for monitoring stress in daily life utilize EMAs to inquire about participants' stress levels throughout the day [196]. The task of effectively gathering accurate stress level indicators from individuals in the context of their everyday activities presents a significant challenge [201]. The over-frequent activation of EMAs or their issuance at times that clash with a user's schedule, such as during work hours or rest periods, can be an imposition. This may culminate in a reduced quantity of reported stress labels. Additionally, pinpointing the precise moments for sending EMAs, particularly in moments when stress levels are elevated, is of paramount significance and presents a notable challenge.

Existing methods in the literature for the deployment of EMAs in daily life stress monitoring studies can be categorized into three distinct categories: 1. Random, 2. Time-based, and 3. Statistical-based.

Within the random triggering methods, EMAs are dispatched at random intervals throughout the course of the study. Random sampling is often the preferred approach in situations where the research topic's indicators cannot be reliably ascertained [202]. However, when the research topic possesses specific objectives and focus, alternative methods tend to yield more favorable outcomes.

Time-based triggering methods involve sending EMAs at fixed pre-defined intervals throughout the day. The majority of existing research endeavors in daily life stress monitoring in the literature employ this algorithm for their label querying system, as documented in previous studies [203–207]. While this algorithm boasts simplicity of implementation and uniform coverage of the study period, it imposes a considerable burden on participants due to untimely EMA deliveries. Consequently, this may result in an increased prevalence of missing data and a reduction in the utility of collected labels.

In the context of statistical-based triggering methods, EMAs are dispatched based on the distribution of samples [208]. Under this triggering algorithm, a label is requested for a specific sample based on the number of unlabeled samples within its vicinity. Although this policy can effectively mitigate the incidence of undesired EMA deliveries, it may still impose a burden on users and lead to missing data, as it fails to consider contextual information about users, which is pivotal in determining the optimal times for EMA deployment.

In this study, we propose a context-aware active reinforcement learning approach to effectively trigger EMAs throughout the day.

Initially we conducted an offline study where we employed a statistical-based triggering method to send EMAs throughout the day. In this phase, the probability of selecting each sample for labeling is proportionate to the quantity of prior unlabeled samples in its proximity. This approach increases the likelihood of requesting a user label for a sample situated in a region with a substantial number of unlabeled samples. Upon accumulating a sufficient number of labeled samples for each region, the data collection process is terminated. We implemented three distinct algorithms offline for optimal label selection in model development: 1. Random, 2. Traditional Active Reinforcement Learning, and our novel approach 3. Context-Aware Active Reinforcement Learning. Our findings revealed that using a context-aware active reinforcement learning algorithm in stress detection significantly decreases the necessity for EMAs enhances the effectiveness of stress detection over random or traditional active learning methodologies. As previously noted, statistical-based triggering algorithm may still impose a user burden and result in missing data due to its failure to incorporate user contextual information into the label-querying decision-making process.

In the next phase, we propose an online context-aware active reinforcement learning algorithm to utilize RL agent for decision making in real time to further improve the performance. This algorithm actively utilizes a context-aware active learning approach in real-time based on Deep Q-Learning to determine whether an EMA should be triggered for a given sample. By considering real-time contextual information related to each user in the decision-making process of whether to trigger an EMA for a specific sample, our approach is poised to significantly reduce the user burden associated with untimely EMA deliveries, consequently leading to an increase in the acquisition of high-utility labels. Table 6.1 provides a summary of the comparison between our work (offline and online studies) and existing literature on this subject.

6.3 Offline Study

In the initial stage of this work, we target to evaluate the effectiveness of our proposed label triggering method [197]. The EMAs are dispatched to participants' phones on a statistical-based basis. Once data collection was completed, we applied our proposed method of context-aware active reinforcement learning for the labeling process. Our offline study involved an Institutional Review Board (IRB)-sanctioned study on human subjects, during which we gathered over 2,629 days of data in everyday environments from college students.

The collected dataset encompasses PPG and various motion metrics (such as acceleration,

Table 6.1: Comparison of our study vs existing works

Study	**Triggering Method**	**EMA Frequency**	**Real-time Analysis**	**Data-based queries**	**Context-based queries**	**Online triggering method**
Yu et al. [203]	Time-based	1.5 hours apart, 10/day	✗	✗	✗	✗
Yu et al. [204]	Time-based	1/day	✗	✗	✗	✗
Mundnich et al. [205]	Time-based	1/day	✗	✗	✗	✗
Wang et al. [206]	Time-based	Every three months	✗	✗	✗	✗
Battalio et al. [207]	Random and Time-based	End of the day, varies	✓	✓	✗	✗
Our Offline Study	Statistical-based	A cap of seven EMAs per day	✓	✓	✗	✗
Our Online Study	**Active Reinforcement Learning**	**A cap of seven EMAs per day**	✓	✓	✓	✓

gyroscope, and gravity readings), and is partially annotated with information on stress levels, emotional states, and physical activities, determined through EMAs conducted at statistical-decided intervals.

Our data collection initiative, spanning from June 2020 to June 2021, involved 20 volunteers selected from undergraduate and graduate student populations. The demographic breakdown of the participants included 13 male and 7 female students, ranging in age from 19 to 29 years. During the study, we gathered 109,586 samples over a period of 2,629 days. Participants contributed to the study for periods ranging from 11 to 287 days, with an average participation duration of 131 days. On average, each subject contributed 5,479 instances to

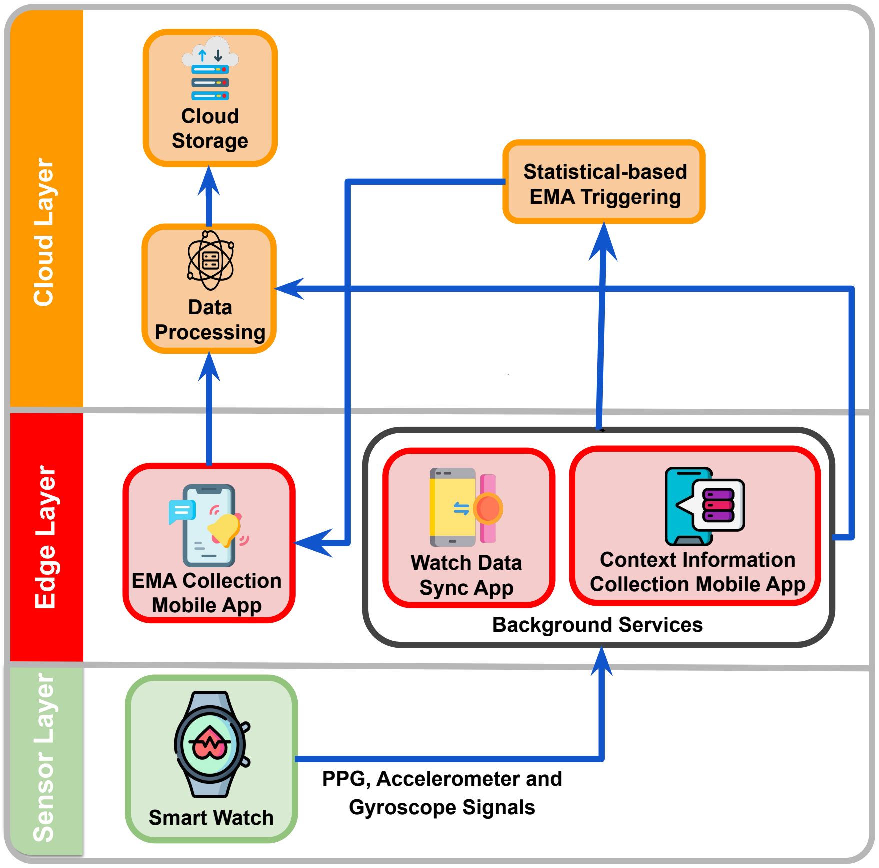

Cloud Layer
Edge Layer
Sensor Layer
Cloud Storage
Data Processing
Statistical-based EMA Triggering
EMA Collection Mobile App
Watch Data Sync App
Context Information Collection Mobile App
Background Services
Smart Watch
PPG, Accelerometer and Gyroscope Signals

Figure 6.1: System Architecture - Offline Study

the dataset.

6.3.1 Proposed System Architecture

Creating a reliable system to gather real-time physiological and contextual data while using active learning from participants is challenging. Wearable devices like smartwatches can be affected by motion artifacts, requiring extensive processing for stress detection [209]. Timing label requests is crucial to ensure participant engagement and label reliability. Figure 6.1 depicts our offline proposed three-layer system including a sensor layer for data acquisition, an edge layer for data transmission and user interaction, and a cloud layer for data processing and decision-making respectively. This system architecture illustrates the architectural composition of our proposed three-layer system, called ZotCare [210].

6.3.2 Sensor layer

This study utilizes Samsung Galaxy Active 2 Watches, equipped with PPG (20Hz), accelerometer, and gyroscope sensors [211]. We developed a Tizen-based smartwatch app to collect these signals [212]. Data is sent to the cloud via Wi-Fi or Bluetooth to a smartphone when Wi-Fi is unavailable. The raw signal acquisition program consists of two services and a user interface (UI). The first service sends sensor data to the cloud every 15 minutes at 2-minute intervals.

6.3.3 Edge layer

We employ the AWARE framework [213] to collect contextual data in everyday scenarios. AWARE is an open-source mobile tool designed for recording, sharing, and reusing context-related information on mobile devices. It utilizes the built-in sensors of smartphones to capture various aspects of daily life, including battery status, weather conditions, location, screen activity, and more. In situations where Wi-Fi connectivity is unavailable, we utilize an alternative smartphone application installed on the edge of our network. This application collects raw PPG signals and accelerometer data from the sensor layer through Bluetooth and subsequently transmits this data to cloud storage. To obtain stress level ratings from our study participants, we have developed an additional smartphone application. This application employs an EMA approach to request stress level assessments from the participants.

6.3.4 Cloud layer

This layer comprises two distinct modules:

- **Data Processing:** This module focuses on processing data retrieved from the edge layer, with its primary objectives being encryption and the storage of data in the appropriate format on ZotCare servers.

- **Statistical-based EMA Triggering:** Our label triggering method is consists of two phases:

  - **Initial Stage:** To obtain an initial approximation of the sample distribution within the sample space, we start the procedure with observation. During the first N samples (100 samples in our configuration, equivalent to approximately 25 hours of wearing the watch), no EMAs are initiated. By the conclusion of this phase, an estimation of the sample distribution in the sample space is obtained.
  - **Query Stage:** Subsequent to the initial phase (from N+1 onward), EMAs are triggered (labels requested) for a subset of samples. The selection probability for labeling each sample is proportionate to the number of preceding samples (unlabeled) in its proximity. This approach ensures that samples in regions with a substantial number of unlabeled counterparts are more likely to be queried for labels. Once a sufficient number of labeled samples are acquired for a particular region, the label collection for that region ceases. Nonetheless, the minimum probability of triggering an EMA for a sample is set at P = 0.1. Consequently, even if a sample is situated in an area with few or no previous samples, the probability of a query remains nonzero. This design enables exploration of unseen regions as well as regions with higher densities while maintaining a balanced approach.

6.3.5 Preprocessing

Following the conclusion of study and data collection, the collected raw PPG signals are preprocessed in order to extract relevant features for model building.

Data Cleaning

The bio-signals from wearable devices, which are inherently noisy, are stored directly in the cloud. The goal of this step is to remove clearly erroneous data points. We use the motion data to remove noises and artifacts in the bio-signals. In our study, we primarily focus on refining PPG signals and heart-rate data. For PPG signals, we utilize a bandpass Butterworth filter with cutoff frequencies between 0.7 Hz and 3.5 Hz. To ensure consistency, the filter is of the third order, and we adopt a sampling rate of 20 Hz, which aligns with our data collection parameters. Additionally, we implement a moving average over a 1-second window to smoothen the PPG data, mitigating artifacts commonly induced by body gestures and movements in daily environments.

Data Normalization

Normalization is indispensable when aiming to minimize variances specific to individual participants and countering the repercussions of subpar bio-signal samples. A standout method in this context, particularly in statistical analyses and machine learning, is the min-max normalization. This technique scales feature values consistently within a predetermined range, commonly [0, 1]. By doing so, it ensures the inherent structure of the data remains intact, which becomes crucial for algorithms that might be affected by the magnitude of the features. For our PPG bio-signals, we've employed the min-max normalization as our primary estimator to ensure that each feature aligns appropriately within a range dictated by the training set.

Feature Extraction

In our investigation, a feature extraction module was employed to analyze PPG data in 2-minute intervals. This facilitated the identification of PPG peaks and the derivation of key metrics such as heart rate. Utilizing the HeartPy library [214] for comprehensive PPG signal processing, we extracted 12 features from both electrical activations and pressure waveforms in the dataset. The extracted PPG features are presented in Table 6.3.

6.3.6 Context-aware Active Reinforcement Learning Algorithm

Our study utilizes the Context-aware Active Reinforcement Learning algorithm to label the collected data in our offline study. In this section, we provide a detailed explanation of this method and compare it with the random selection method and traditional active reinforcement learning methods for label querying.

In traditional supervised learning, the entire labeled dataset was utilized for training [215]. However, within our personalized data collection approach, we accumulated labeled data from diverse sources over time. To leverage this, we iteratively queried our users for new annotations. Commencing with a subset of labeled data, we employed a query mechanism to identify the most informative unlabeled instances with the assistance of human experts.

Active learning played a pivotal role in this process, strategically selecting data samples for labeling to enhance model accuracy while minimizing data usage. Strategies encompassed uncertainty sampling (opting for ambiguous data) and diversity sampling (choosing unique and indicative data). Nevertheless, real-world queries incurred costs, necessitating a delicate balance between query expenses and model improvement.

In our scenario, we encountered distinctive challenges. User behavior influenced data quality and availability, contingent on factors such as activity, time, query frequency, and phone interaction. A lack of response resulted in the denial of labeling and delayed responses, diminishing data quality alignment. Consequently, our active learning approach needed to consider not only data quality but also future label accessibility. We proposed the use of Deep Q-Learning, where an agent modeled user behavior to ensure sustained user engagement.

Deep Q-learning

Deep Q-Learning (DQN) [216] is a model-free, online, off-policy reinforcement learning method. At its core, DQN seeks to estimate the action-value function, denoted as $Q(s, a)$, which predicts the expected return after taking an action a in state s. The Bellman equation, which is fundamental to Q-learning, is given by:

$$Q(s, a) = r + \gamma \max_{a'} Q(s', a') \tag{6.1}$$

Where r is the immediate reward, γ is the discount factor, and s' is the subsequent state after taking action a in state s. The primary distinction between traditional Q-learning and DQN is the utilization of deep neural networks to approximate the Q-values. This is paramount for tasks with large state spaces. The loss $\mathcal{L}$ during training is defined as:

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,a,r,s') \sim U(D)} \left[\left(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \right)^2 \right] \tag{6.2}$$

Where D is the replay buffer, $U(D)$ is a uniform random sample from D, θ are the network parameters, and θ^- are the target network parameters.

In the following part, we will elaborate the detailed definitions in our DQN.

State Representation

The state vector, denoted by s, encodes crucial information leading to decision-making. The intention is to refine and personalize the stress detector to optimize accuracy while minimizing user queries. The state comprises:

- **Uncertainty Factor**: Originating from the raw output of a pre-trained classifier. This factor measures the distance from the decision boundary, effectively quantifying the confidence of the prediction.
- **Time-aware Response Rate**: This accounts for the time of the day (in hourly intervals) and embodies the user's responsiveness across different hours.
- **Time since Last Query**: To enhance user experience and prevent excessive querying in short time frames.
- **Time of Day**: Represents the current hour and is used to model potential variations in stress levels throughout the day.

Reward Formulation

The reward function integrates components from the 'n_state' vector for holistic decision-making. It is formulated as:

$$r_0 = \frac{1}{1 + e^{-20(n_{\text{state}[0]} - 0.5)}} \tag{6.3}$$

$$r_1 = \text{reward\_F}(n_{\text{state}[1]}) \tag{6.4}$$

$$r_2 = \frac{1}{1 + e^{-10(n_{\text{state}[2]} - 0.5)}} \tag{6.5}$$

The overall reward function, R, based on the action taken, is:

$$R(\text{action}) = \begin{cases} \text{reward\_p} & \text{if action is True} \\ 3 - \text{reward\_p} & \text{if action is False} \end{cases} \tag{6.6}$$

Q-Network Design

The Q-network constitutes the backbone of our framework. The structure and features are enumerated below:

- The core is a densely connected neural network geared towards estimating Q-values.
- Input: The network takes in 4 nodes, matching the count of state variables.
- Hidden Layers: The architecture consists of variable hidden layers, as specified by the list h. In the provided example, four hidden layers are employed with 5, 9, 7, and 5 nodes, respectively. Each of these nodes uses the ReLU activation function and incorporates both $l1$ and $l2$ kernel regularizers, with the $l2$ regularization strength set at $1e^{-2}$.
- Output: The network furnishes 2 output nodes, indicative of the duo of feasible actions, with a linear activation function.

Additionally, in our experiments, the agent employed an ϵ-greedy policy accompanied by a linear annealing schedule for its exploration factor. This strategy ensures a gradual transition from exploration to exploitation during the learning process, thereby enhancing convergence and robustness in diverse environments. For our implementation, we leveraged the Keras-RL library [217]. To elucidate, the ϵ-greedy policy in reinforcement learning can be characterized as follows:

$$\pi(a|s) = \begin{cases} \epsilon + \frac{1-\epsilon}{|A|} & \text{if } a = \text{argmax}_{a' \in A} Q(s, a') \\ \frac{1-\epsilon}{|A|} & \text{otherwise} \end{cases} \tag{6.7}$$

Where:

- $\pi(a|s)$ is the probability of taking action a in state s.
- ϵ is the exploration probability.
- A is the set of possible actions.
- $Q(s, a')$ is the estimated value of taking action a' in state s.

For our experiment, we employed a sequential memory architecture with a capacity limited to 50,000 instances and a window length set at one. The DQN agent was initialized with parameters set as follows: a discount factor γ at 0.95, a warm-up phase consisting of 100 steps, and a learning rate of $1e^{-2}$ for the target model update. Optimization was carried out using the Adam optimizer, and the performance was gauged using the Mean Absolute Error (MAE) metric.

Policy Strategy

The decision strategy, symbolized by $\pi_\theta(s, a)$, selects the action that corresponds to the optimum Q-value for a specified state s. To guarantee a complete traversal of the state space:

- On an estimated 5% of occasions, a query is initiated randomly. This procedure considers the potentially undiscovered areas within the state space.
- Following every $K = 100$ occasions, which ideally occurs once in a 24-hour span, the user's response frequency metrics are re-calibrated according to their interaction patterns. This adjustment acknowledges individual variability and revitalizes the query selection methodology tailored for each user.
- The stress detection module undergoes periodic retraining. This process assimilates the most recent subjectively labeled information, amalgamated with the prior objective data procured from diverse subjects.

6.3.7 Evaluation and Results

We initiated the pre-training of the stress detector, which was subsequently utilized to derive the state and reward in constructing the active learning framework. A random forest classifier with n = 500 estimators (number of trees) and max depth = 5 for each tree was employed. Participants were requested to assess their stress levels on a five-point scale: (1) not at all, (2) a little bit, (3) some, (4) a lot, and (5) extremely. We translated the stress labels into two categories: a lot and extremely as 1 (stressed), and the remaining three labels as 0 (not stressed).

We conducted training for the model using labeled data from 14 different subjects, reserving one subject for personalization. The newly trained model on this subject exhibited a recall value of recall = 0.238 for the minority class (class stressed). The Q-learning agent underwent pre-training with an offline sequential objective dataset. Upon completing a sequence (referred to as one episode), we restarted from the beginning until reaching the total number of steps. The model underwent training for K=200,000 steps. The total reward achieved during the episodes reached a saturation point before this stage, indicating the model's convergence.

The agent underwent training in two distinct modes. Initially, we employed a "traditional" approach, where the agent's attention was solely on the state and reward associated with the classifier's raw output. The agent received significant rewards for executing the 'submit query' action for instances within the classifier's uncertainty region, while being rewarded for the 'do not submit query' action for other instances. In this mode, the agent operated without capturing contextual information and functioned akin to a conventional active learning selection policy. Subsequently, a modification was introduced by incorporating contextual information into the reward function, as previously described. High rewards were assigned for the 'submit query' action for instances not only within the uncertainty region but also from a time interval when the user demonstrated increased responsiveness. Moreover, instances not in a short time distance from the preceding 'select' action were considered. For other instances, the agent received rewards for executing the 'do not submit query' action.

We conducted an analysis on the quantity of queries needed to achieve a particular level of personalization. The experiment was repeated N = 100 times to mitigate the influence of random selection. The average number of instances is presented in Figure 6.2. Notably, a substantial disparity exists between random selection and DQN agents, with the context-aware agent demonstrating the capability to attain high performance with a significantly lower number of queries. Specifically, the context-aware selection policy reduces the required queries by up to 88%, in contrast to the random selection method. While the number of necessary labels remains consistent for the two DQN agents throughout the analysis (as expected), the context-aware agent manages to reduce the required queries by up to 32%. These findings, derived from a subject with a higher number of labels, exhibit similar trends when extended to data from other subjects.

We examined the effectiveness of two agents and a random selection policy in achieving the primary objective of personalizing the classifier, and the findings are illustrated in Figure 6.3.

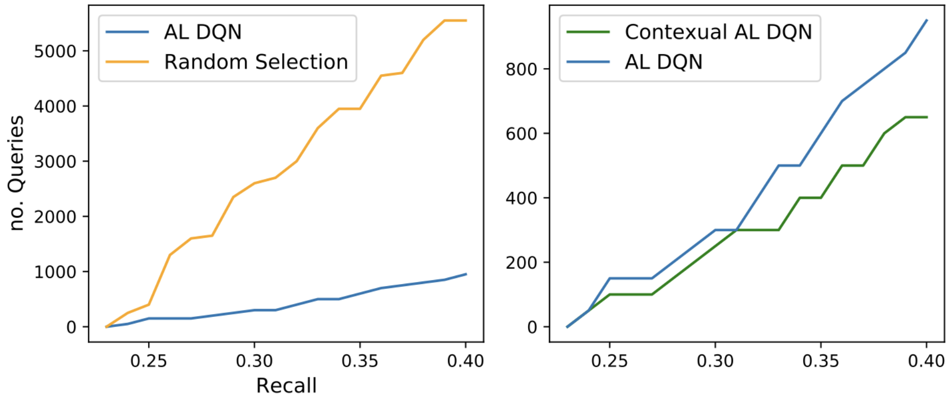


Figure 6.2: Number of queries needed to reach a certain performance level during personalization.

The number of instances selected for querying remained consistent across each step for all selection methods. However, a subset of these selected instances stayed unlabeled, resulting in different quantities of instances available for personalization depending on the selection policies. Aside from this, the selected instances had varying impacts on personalization under different policies, comparing random selection to the other two methods. From a single subject, we obtained a total of 12,700 instances, with 922 labeled. We reserved 230 labeled instances (from the end of the sequence) as test data from one subject, leaving the remainder for training (25% - 75%). The process started without any personalization, gradually progressing through partially labeled subjective data. A subset of this data was chosen for querying, and a part of the selected data was labeled, with the labeled data being utilized for personalization. At each step, the context-aware agent selected fewer instances than the non-context-aware agent. To ensure a fair comparison, we randomly down-sampled the number of queries from the larger group. Additionally, for random selection, we randomly picked a number of partially labeled samples equivalent to the number of queries from the agents. To mitigate the impact of random selection, at each step, we selected instances and personalized the models N = 100 times. Figure 6.3 displays the mean and standard deviation of recall (True Positive Rate) for the stressed class on test data for the three selection methods.

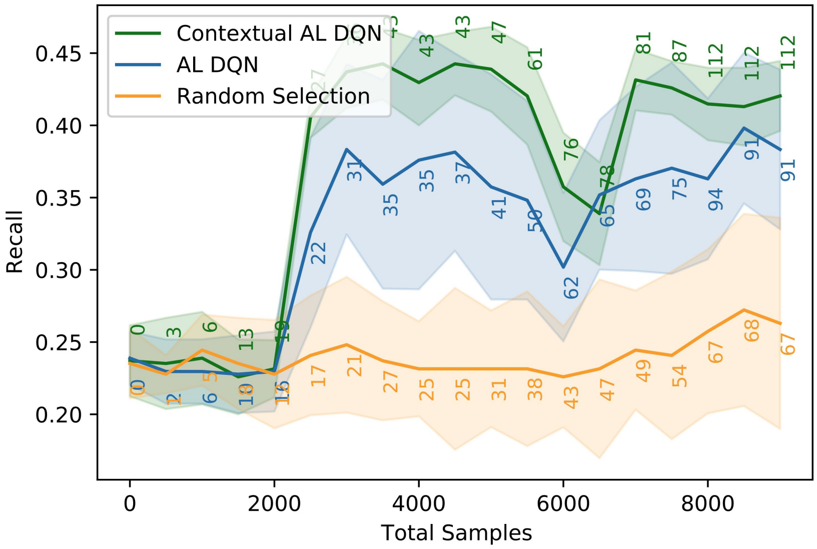


Figure 6.3: Presonalization Recall in Previous Work

Instances that are selected by the random agent do not improve the performance significantly since they include samples from the entire region of the input space of the classifier, including samples whose class is 'trivial' to be extracted. Instances that are selected by a non-contextual active learning method (blue curve) increase the performance. However, with an equal number of queries, the best result is achieved when the agent is context aware (green curve), since it results in a higher number of impactful instances which also have a higher chance to receive the label.

6.4 Online Study

In our online study, we employ the Context-aware Active Learning Deep Q-Network (Context-aware AL DQN) algorithm, aligned with our offline investigation, to assess the efficacy of our proposed algorithm in a real-time setting where users are actively involved in training the Reinforcement Learning (RL) agent for decision-making. This real-time approach significantly reduces the user burden and has the potential to enhance stress detection performance compared to its offline counterpart by leveraging a real-time smart RL agent that query labels.

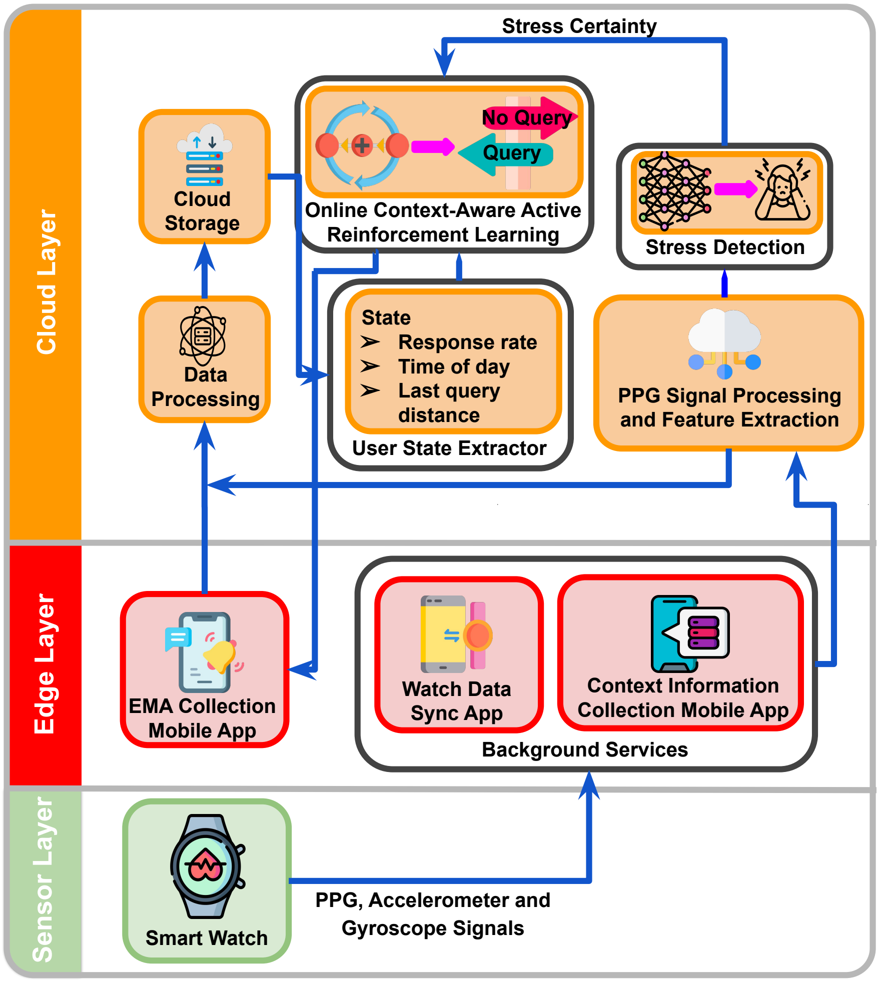


Figure 6.4: System Architecture - Online Study

We assessed data derived from a cohort of 34 individuals. This study spanned from March 2022 to May 2023. Participants, ranging in age from 19 to 29 years, provided a comprehensive dataset. After filtering out anomalous and noisy records, we aggregated 23,012 samples over a period of 420 days. On an average basis, each participant yielded 676 distinct samples. It is noteworthy to mention that the respective IRB granted approval for all aspects of this investigation.

6.4.1 Proposed System Architecture

The proposed system architecture is illustrated in Figure 6.4. Consistent with our offline study, we maintain a three-layer system, denoted as ZotCare. A comparison with the architecture presented in Figure 6.1 reveals that, although the sensor layer and the edge layer remain unchanged, substantial modifications are implemented in the cloud layer. These alterations are undertaken to render the system conducive to real-time label querying through the utilization of our proposed context-aware active reinforcement learning algorithm explained in 6.3.6.

Cloud layer mainly comprises of four distinct modules to replace the previous simple statistical-based triggering method with our proposed triggering algorithm.

- PPG Signal Preprocessing and Feature Extraction: To effectively identify moments conducive to experiencing stress, we continuously monitor participants' stress levels in real time using PPG signals from their watches. However, to make these signals suitable for stress prediction, several preprocessing steps are required. This module is dedicated to preparing the PPG signals for stress detection. More details can be found in Section 6.4.2.
- Stress Detection: We utilize the data from our previous study [197] to construct our stress detection module. The features extracted from the PPG signals are input into this module for stress detection. The level of certainty regarding stress is then forwarded to the context-aware active reinforcement learning module to aid in identifying stressful moments.
- Context Recognition: Within this module, we extract contextual information pertaining to each user, which is subsequently provided to our active learning module for decision-making purposes. This includes factors such as the time elapsed since the last query, the time of day, and the time-aware response rate. The time-aware response rate considers the user's responsiveness within the current hour based on their historical activity.
- Context-Aware Active Reinforcement Learning: The primary objective of this module

is to determine whether it is appropriate to trigger an EMA at any given moment. Stress certainty, time elapsed since the last query, time of day, and time-aware response rate are all input into this module to inform the decision-making process. If an EMA needs to be triggered, a notification is dispatched to the user's mobile device on the edge layer, prompting them to rate their current stress level. In the following section, we will delve deeper into the training process of the active reinforcement learning agent.

6.4.2 Preprocessing

PPG signals, contextual AWARE data, and user-reported stress levels are collected from the cloud for stress model construction. However, raw cloud-stored PPG and AWARE data need preprocessing before building the model. This section explains our data preparation steps.

Data Cleaning and Normalization

This study employs the same modules for data cleaning and normalization as discussed in the offline study section (see Section 6.3.5).

Feature Extraction

- PPG Features: This module has been previously discussed in 6.3.5. For the information regarding the PPG features, please refer to the Table 6.3.
- Contextual Features: The raw contextual information obtained from AWARE is not ready for building the stress detection models. We transform both categorical and numerical raw features into solely numerical features. We show the features extracted from raw AWARE data in Table 6.2.

Table 6.2: AWARE Features

Feature	Definition
Call	Call duration, type, and count
Notification	APP source and count
Screen & Touch	User screen interactions
Battery	Battery charge duration and level
Message	Message type and count
Time	Time of the day (24-hour format)
Location	Longitude, latitude, altitude

Table 6.3: PPG Features

Feature	Definition
BPM	Heart beats per minute
IBI	Inter-Beat Interval, the average time interval between two successive heartbeats (NN intervals)
SDNN	Standard deviation of NN intervals
SDSD	Standard deviation of successive differences between adjacent NNs
RMSSD	Root mean square of successive differences between the adjacent NNs
PNN20	The proportion of successive NNs greater than 20ms
PNN50	The proportion of successive NNs greater than 50ms
HR_mad	Median absolute deviation of NN intervals
SD1 and SD2	Standard deviations of the corresponding Poincare plot
S	Area of ellipse described by SD1 and SD2
BR	The number of breaths per minute (breathing rate)

Data Labeling

The EMA protocol is set to activate no more than seven times daily, prompting the participants to rate their stress levels on a five-point scale: (1) not at all, (2) a little bit, (3) some, (4) a lot, and (5) extremely. These self-reported stress levels, along with their associated timestamps, are archived in the cloud for future analysis. Each 15-minute interval of accumulated physiological and contextual data is labeled in accordance with the nearest subsequent EMA response. The distribution of these labels can be seen in Figure 6.5.

6.4.3 Evaluation and Results

In order to conduct a comprehensive comparison between our online context-aware active learning method for stress detection and previously offline variant, we have deliberately

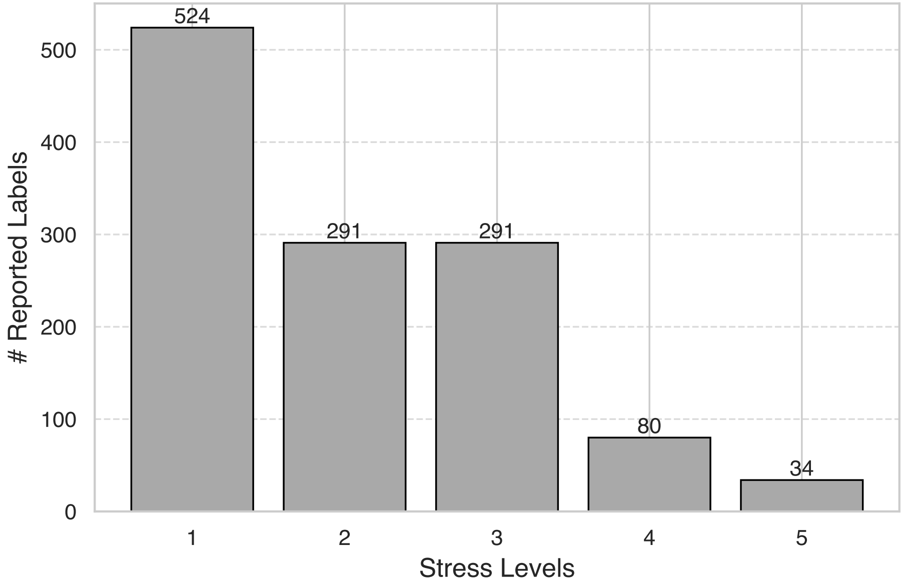


Figure 6.5: Distribution of Stress Labels

employed identical classification algorithms for both studies.

Three distinct classification techniques have been used: Support Vector Machines (SVM) [218], Random Forest [94], and XGBoost [219]. SVM finds a hyperplane in high-dimensional space to separate data classes. Random Forest uses multiple decision tree classifiers on dataset subsets, improving predictive accuracy while avoiding overfitting. Additionally, XGBoost is employed, providing an effective gradient-boosted trees implementation.

The utilization of these diverse classification techniques enables a comprehensive and robust evaluation of our proposed stress detection algorithm.

Our stress detection models are classified into two categories: single-modal and multi-modal algorithms. Within the single-modal algorithm, solely the PPG signal is employed for constructing the stress detection models. On the other hand, the multi-modal algorithm utilizes both the PPG signal and contextual information (AWARE data) in the development of the proposed models.

6.4.4 Classification Performance

Experiment Detail

In order to ensure a fair evaluation, we utilize the k-fold cross validation technique [220] with k equal to 4.

K-fold cross-validation involves splitting the data into multiple subsets for training and testing the model. It prevents overfitting, utilizes all available data, and improves model robustness against data variations. Averaging results across folds provides a reliable way to evaluate model performance, making it valuable for model selection and hyperparameter tuning.

Evaluation Metrics

To evaluate our stress monitoring system, we use three key metrics: F1-score, precision, and recall. The F1-score assesses binary categorization test accuracy, calculated from precision and recall, where precision measures correctly identified "true positive" results and recall identifies all "true positive" results. F1-score is a weighted average of precision and recall, important for binary classification tests.

Classification Performance Results

Table 6.4 presents a comprehensive performance analysis of our novel stress detection algorithm, incorporating an online context-aware active learning approach, compared with the offline variant.

The results clearly illustrate the substantial performance enhancements achievable with the online context-aware algorithm across all evaluated metrics when compared to the offline counterpart. Notably, for the Random Forest classifier, we observe a noteworthy 11% improvement in F1-score. The significant improvement in performance underscores the importance of employing intelligent real-time label triggering methods to identify optimal moments for sending Ecological Momentary Assessments (EMAs).

This outcome also underscores the considerable advantage of incorporating contextual awareness into our model, resulting in significant enhancements across various classification metrics and reaffirming the pivotal role of context in stress detection tasks.

Table 6.4: Classification Performance Results

		Classification Model								
		Random Forest			**XGBoost**			**SVM**		
Active Learning Method	**Data**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**
Offline Context-Aware	PPG	0.21	0.27	0.17	0.31	0.35	0.27	0.41	0.32	0.58
Online Context-Aware	PPG	0.32	0.43	0.25	0.39	0.43	0.35	0.5	0.41	0.64
Online Context-Aware	PPG and Context	**0.36**	**0.49**	**0.28**	**0.40**	**0.45**	**0.36**	**0.52**	**0.45**	0.61

6.4.5 Personalization Performance

Leveraging the unique physiological and behavioral variations in individuals can significantly enhance the efficacy of generic models. Inspired by the potential advantages of individualized prediction models, we hypothesize that personalizing reinforcement learning models might similarly elevate data quality. To validate this premise, we initially trained a generalized representation model using the aggregated training data from all users. Subsequently, we fine-tuned this model for each user individually, aiming to discern potential enhancements in prediction accuracy. Our evaluation centered on contrasting these generalized and personalized models to elucidate the tangible benefits of our individual-based personalization strategy in data collection.

Experiment Detail

To accurately evaluate the affect of personalization in our data collection mechanism, we implemented a unique train-test splitting strategy. Our dataset comprises data from multiple users. To ensure a robust evaluation, we adopted a leave-one-subject-out cross-validation scheme. In each round of this scheme, data for each user is divided temporally into two parts. The initial half serves the purpose of model personalization, while the latter half is reserved for testing.

Two distinct models were constructed for comparative assessment:

- **Plain Model**: This model is trained using the entire dataset except for the data of the user currently under consideration. For testing and evaluation, the latter half of this user's data is employed.
- **Personalized Model**: This model, on the other hand, is trained using the complete dataset (excluding the data of the current user) combined with the initial half of the current user's data. Again, the latter half of the user's data is utilized for testing.

Comparing these two models helps us gauge the effectiveness of our personalization strategy. By contrasting their performance, we can see how incorporating user-specific data for training improves accuracy significantly compared to using a generic global dataset.

Personalization Performance Results

Table 6.4 showcases the comparative performance of our stress detection model in both personalized and unpersonalized configurations. We present outcomes from both the ROC curve, as referenced in 6.4, and additional performance metrics. The ROC curve assesses binary classification model efficacy by illustrating the relationship between True Positive Rate

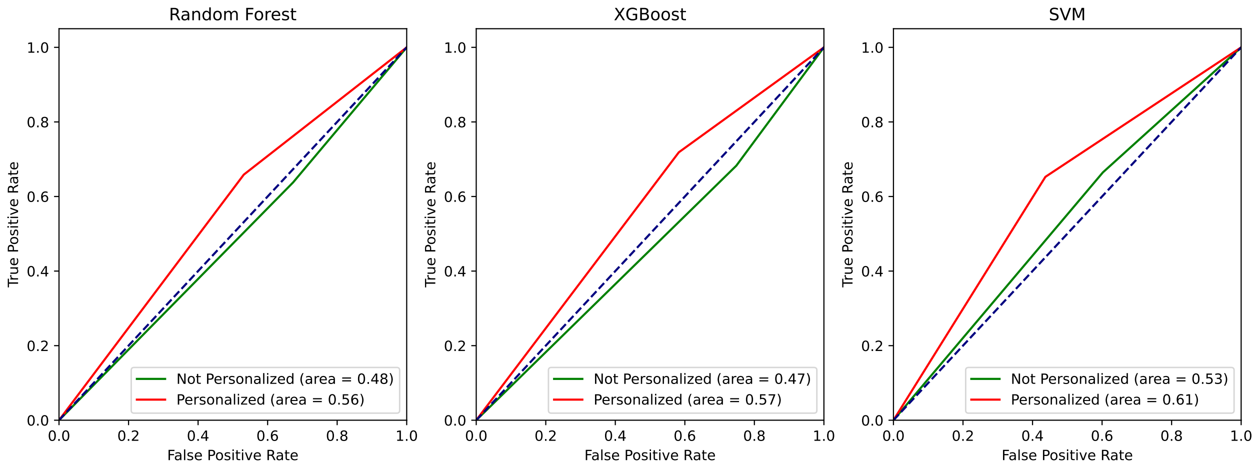


Figure 6.6: Presonalization ROC Curve

and False Positive Rate across various decision thresholds. An elevated Area Under the Curve (AUC) signifies superior model performance, underscoring its merit as a comparative measure. The findings, as depicted in the provided figure and table, reveal that adopting a personalized training approach markedly amplifies the efficacy of our stress detection strategy, as evidenced by the AUC-ROC score. Notably, when employing the XGBoost classifier, we observed a pronounced boost of approximately 10% in the AUC-ROC score.

Table 6.5: Personalization Results

		Classification Model							
		Random Forest		**XGBoost**			**SVM**		
Personalized Training Method	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**
Not Personalized	0.60	0.55	0.64	0.60	0.54	0.68	0.62	0.59	0.66
Personalized	**0.64**	**0.61**	**0.66**	**0.66**	**0.61**	**0.71**	**0.65**	**0.66**	0.65

6.5 Conclusion

In conclusion, this work introduced a novel contextual variant of active learning, leveraging Deep Q-Learning to incorporate individual contextual information into the decision-making process [197, 208]. In the initial phase, the implementation of a context-aware active reinforcement learning algorithm in an offline setting showcased its efficacy, resulting in a significant reduction of up to 88% in required EMAs compared to random selection and up to 32% compared to traditional active learning methods. Additionally, stress detection performance exhibited notable improvements, with up to a 21% enhancement compared to random selection and up to 8% compared to traditional active learning.

Moving to the second phase, our online implementation of the algorithm utilized active learning for EMA initiation, leveraging real-time contextual information to optimize question timings and reduce participant burden. Comparative analyses of the offline and online variants on the same dataset unequivocally demonstrated the superiority of the online algorithm, showcasing a potential improvement of up to 11% in stress detection performance. Incorporating contextual features further improved results by 4%, emphasizing the significance of personalization in enhancing model performance.

This study not only contributes a valuable advancement in stress detection methodologies but also underscores the pivotal role of context-awareness and online implementation in achieving superior results. The demonstrated reductions in participant burden and improvements in label accuracy signify the potential practical impact of this research in real-world applications. Future directions may explore additional personalization techniques and extend the application of context-aware active learning to diverse domains, fostering continued advancements in intelligent and user-centric systems.

Chapter 7

Summary and Conclusion

This thesis has delved into the complex interplay between affective states, mental health, and the autonomic nervous system (ANS), underscoring the significance of monitoring physiological signals to better understand human emotions and moods. By leveraging advancements in wearable IoT devices, we have developed a context-aware, personalized, and robust system for monitoring affect in daily life.

Traditional methods for monitoring affect, such as patient self-reporting and diary logging, present several limitations. These methods often rely heavily on the individual's subjective interpretation and memory, which can be inaccurate or biased. Additionally, they require significant effort and consistency from the individual, making long-term adherence challenging. These approaches also necessitate assistance from a therapist or provider, which can be resource-intensive and less practical for continuous monitoring. As a result, traditional methods may fail to capture real-time fluctuations in affective states and miss critical data points that are essential for comprehensive affect monitoring.

To address the limitations of traditional self-reporting methods, we proposed a novel system architecture integrating Context-Awareness, Personalization, and Robustness. This system collects diverse physiological and contextual data, tailors the monitoring process to individual differences, and ensures resilience against daily-life noise and motion artifacts.

Leveraging Deep Learning Techniques to Extract more Relevant Stress-Related Features: Chapter 2 introduced pyEDA, a tool designed for the automatic extraction of features from physiological signals such as photoplethysmography (PPG) and electrodermal activity (EDA). PyEDA demonstrated its capability to handle large datasets and extract meaningful features essential for affective monitoring, facilitating large-scale data processing and analysis.

Motion Artifact Removal for Ensuring System Robustness and Enhanced Performance: Chapter 3 focused on developing a PPG motion artifact removal module. This module is crucial for maintaining robust performance in everyday scenarios, where motion artifacts can significantly distort physiological data. Advanced signal processing techniques were employed to mitigate these artifacts, ensuring the reliability and accuracy of the collected data.

Integrating Respiratory Rate Extraction for Advanced Multimodal Analysis: In Chapter 4, we presented a novel module for extracting respiration rate from PPG signals. Incorporating respiratory rate as an additional modality enriched the affective monitoring system, providing deeper insights into physiological responses. The algorithms and validation processes used to accurately derive respiration rate were thoroughly discussed and demonstrated their significance in understanding physiological responses.

Developing a Context-Aware Closed-Loop System: Chapter 5 detailed the comprehensive context-aware closed-loop system architecture, integrating physiological data, contextual information, and stress labels. This integration is key to understanding affect in real-life settings. The system components and their interactions were elaborated upon, showcasing how the closed-loop design enhances the accuracy and relevance of affective monitoring.

Incorporating Personalization to Elevate User Experience and Engagement: Finally, Chapter 6 applied a context-aware active reinforcement learning approach to the closed-loop system. This approach aims to improve system performance and user engagement by dynamically adapting to the user's context and feedback. The reinforcement learning framework, experimental setup, and results were discussed, demonstrating the effectiveness of this approach in real-world applications.

Chapter 8

Future Directions

Enhanced Personalization: One potential avenue for future research is the further refinement of the system to improve its ability to adapt to individual differences in affective responses and physiological signals. By integrating more sophisticated algorithms and machine learning models, the system could offer a more personalized experience, adjusting its responses and recommendations based on the unique characteristics of each user. This would not only increase the system's effectiveness but also enhance user satisfaction by providing more tailored interventions.

Expanded Sensor Suite: Another direction for development is the exploration of additional wearable sensors that can capture a broader range of physiological and contextual data. Incorporating sensors that measure variables such as skin conductivity, muscle tension, or environmental factors could provide a more comprehensive understanding of the user's state and surroundings. This expanded sensor suite would enable the system to deliver more accurate and context-aware interventions, potentially improving outcomes in various settings.

User Experience Optimization: To maximize the system's adoption and effectiveness, future work should focus on optimizing the user interface and overall experience. This includes making the system more intuitive, engaging, and accessible to a diverse range of users. By streamlining the interface and incorporating user feedback, the system can become more user-friendly, encouraging sustained engagement and ensuring that users can easily integrate it into their daily routines.

Longitudinal Studies: Finally, conducting long-term studies is essential to assess the system's effectiveness and impact on mental health over extended periods. These studies would provide valuable insights into how the system performs in real-world conditions and its long-term benefits or challenges. Understanding the system's effects over time would inform future iterations and help establish its role in mental health interventions, ensuring it delivers sustained positive outcomes.

Bibliography

[1] Seyed Amir Hossein Aqajari, Ziyu Wang, Ali Tazarv, Sina Labbaf, Salar Jafarlou, Brenda Nguyen, Nikil Dutt, Marco Levorato, and Amir M Rahmani. Enhancing performance and user engagement in everyday stress monitoring: A context-aware active reinforcement learning approach. *arXiv preprint arXiv:2407.08215*, 2024.

[2] Seyed Amir Hossein Aqajari, Sina Labbaf, Phuc Hoang Tran, Brenda Nguyen, Milad Asgari Mehrabadi, Marco Levorato, Nikil Dutt, and Amir M Rahmani. Context-aware stress monitoring using wearable and mobile technologies in everyday settings. *arXiv preprint arXiv:2401.05367*, 2023.

[3] Amir Hosein Afandizadeh Zargari, Seyed Amir Hossein Aqajari, Hadi Khodabandeh, Amir Rahmani, and Fadi Kurdahi. An accurate non-accelerometer-based ppg motion artifact removal technique using cyclegan. *ACM Transactions on Computing for Healthcare*, 4(1):1–14, 2023.

[4] Seyed Amir Hossein Aqajari, Rui Cao, Emad Kasaeyan Naeini, Michael-David Calderon, Kai Zheng, Nikil Dutt, Pasi Liljeberg, Sanna Salanterä, Ariana M Nelson, and Amir M Rahmani. Pain assessment tool with electrodermal activity for postoperative patients: method validation study. *JMIR mHealth and uHealth*, 9(5):e25258, 2021.

[5] Negin Sattari, Milad Asgari Mehrabadi, Seyed Amir Hossein Aqajari, Jing Zhang, Katharine Simon, Elisabet Alzueta, Teji Dulai, Massimiliano de Zambotti, Fiona Baker, Amir Rahmani, et al. 079 sleep quality prediction during the menstrual cycle based on daily sleep diary reports. *Sleep*, 44:A33, 2021.

[6] Jung-Ah Lee, Seyed Amir Hossein Aqajari, Eunae Ju, Priscilla Kehoe, Lisa Gibbs, and Amir Rahmani. Home-visit intervention to reduce stress of underserved family caregivers for persons with dementia. *Innovation in Aging*, 5(Supplement_1):152–152, 2021.

[7] Mahyar Abbasian, Taha Rajabzadeh, Ahmadreza Moradipari, Seyed Amir Hossein Aqajari, Hongsheng Lu, and Amir M Rahmani. Controlling the latent space of gans through reinforcement learning: A case study on task-based image-to-image translation. In *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*, pages 1061–1063, 2024.

[8] Milad Asgari Mehrabadi, Seyed Amir Hossein Aqajari, Amir Hosein Afandizadeh Zargari, Nikil Dutt, and Amir M Rahmani. Novel blood pressure waveform reconstruction from photoplethysmography using cycle generative adversarial networks. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 1906–1909. IEEE, 2022.

[9] Seyed Amir Hossein Aqajari, Emad Kasaeyan Naeini, Milad Asgari Mehrabadi, Sina Labbaf, Nikil Dutt, and Amir M Rahmani. pyeda: An open-source python toolkit for pre-processing and feature extraction of electrodermal activity. *Procedia Computer Science*, 184:99–106, 2021.

[10] Seyed Amir Hossein Aqajari, Rui Cao, Amir Hosein Afandizadeh Zargari, and Amir M Rahmani. An end-to-end and accurate ppg-based respiratory rate estimation approach using cycle generative adversarial networks. *arXiv preprint arXiv:2105.00594*, 2021.

[11] Milad Asgari Mehrabadi, Seyed Amir Hossein Aqajari, Iman Azimi, Charles A Downs, Nikil Dutt, and Amir M Rahmani. Detection of covid-19 using heart rate and blood pressure: Lessons learned from patients with ards. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 2140–2143. IEEE, 2021.

[12] Rui Cao, Seyed Amir Hossein Aqajari, Emad Kasaeyan Naeini, and Amir M Rahmani. Objective pain assessment using wrist-based ppg signals: A respiratory rate based method. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 1164–1167. IEEE, 2021.

[13] Amir M Rahmani, Nikil Dutt, Kai Zheng, Ariana Nelson, Pasi Liljeberg, Sanna Salantera, Mingzhe Jiang, Arman Anzanpour, Elise Syrjala, Riitta Mieronkoski, et al. Pain assessment method and apparatus for patients unable to self-report pain, November 14 2019. US Patent App. 16/406,739.

[14] Paul Ekman. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200, 1992.

[15] David Watson and Lee Anna Clark. The panas-x: Manual for the positive and negative affect schedule-expanded form. *Unpublished manuscript, University of Iowa*, 1994.

[16] James A Russell. Core affect and the psychological construction of emotion. *Psychological review*, 110(1):145, 2003.

[17] National Institute of Mental Health. Mental illness, 2019.

[18] JM Guérit. The central nervous system. structure and function.-oxford university press (1998), 676 p. *Neurophysiologie Clinique/Clinical Neurophysiology*, 4(29):365–366, 1999.

[19] Julian F Thayer and Richard D Lane. A model of neurovisceral integration in emotion regulation and dysregulation. *Journal of affective disorders*, 61(3):201–216, 2000.

[20] Jajack Heikenfeld, Andrew Jajack, Jim Rogers, Philipp Gutruf, Lei Tian, Tingrui Pan, Ruya Li, Michelle Khine, Jintae Kim, and Juanhong Wang. Wearable sensors: modalities, challenges, and prospects. *Lab on a Chip*, 18(2):217–248, 2018.

[21] Hamidreza Alikhani, Anil Kanduri, Pasi Liljeberg, Amir M. Rahmani, and Nikil Dutt. Dynafuse: Dynamic fusion for resource efficient multimodal machine learning inference. *IEEE Embedded Systems Letters*, 15(4):222–225, 2023.

[22] Anil Kanduri, Sina Shahhosseini, Emad Kasaeyan Naeini, Hamidreza Alikhani, Pasi Liljeberg, Nikil Dutt, and Amir M. Rahmani. *Edge-Centric Optimization of Multimodal ML-Driven eHealth Applications*, pages 95–125. Springer Nature Switzerland, Cham, 2024.

[23] Ziyu Wang, Zhongqi Yang, Iman Azimi, and Amir M. Rahmani. Differential private federated transfer learning for mental health monitoring in everyday settings: A case study on stress detection, 2024.

[24] Hamidreza Alikhani, Ziyu Wang, Anil Kanduri, Pasi Liljeberg, Amir M. Rahmani, and Nikil Dutt. EA^2: Energy efficient adaptive active learning for smart wearables. In *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED)*, Newport Beach, California, USA, August 2024. IEEE.

[25] Ziyu Wang et al. Guardhealth: Blockchain empowered secure data management and graph convolutional network enabled anomaly detection in smart healthcare. *Journal of Parallel and Distributed Computing*, 142:1–12, 2020.

[26] Ahmad Reza Danesh, Mahyar Safiallah, Haoran Pu, and Payam Heydari. An isolated frequency compensation technique for ultra-low-power low-noise two-stage otas. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 71(1):6–10, 2024.

[27] Mahyar Safiallah, Ahmad Reza Danesh, Haoran Pu, and Payam Heydari. A current-adjusting auto-zeroing technique for dc-offset and flicker-noise cancellation. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 31(12):1950–1959, 2023.

[28] Ahmad Reza Danesh, Haoran Pu, Mahyar Safiallah, An H. Do, Zoran Nenadic, and Payam Heydari. A cmos bd-bci: Neural recorder with two-step time-domain quantizer and multi-polar stimulator with dual-mode charge balancing. *IEEE Transactions on Biomedical Circuits and Systems*, pages 1–15, 2024.

[29] Salar Hashemi, Amir M. Hajisadeghi, and Hamid R. Zarandi. EARL: an efficient approximate hardware framework for accelerating fault tree analysis. In *25th Euromicro Conference on Digital System Design, DSD 2022, Maspalomas, Spain, August 31 - Sept. 2, 2022*, pages 32–38. IEEE, 2022.

[30] Salar Hashemi, Amir M. Hajisadeghi, Hamid R. Zarandi, and Saadat Pour-Mozafari. A fast and efficient fault tree analysis using approximate computing. In *15th European Dependable Computing Conference, EDCC 2019, Naples, Italy, September 17-20, 2019*, pages 39–46. IEEE, 2019.

[31] Ahmadreza Moradipari and Mahnoosh Alizadeh. Pricing and routing mechanisms for differentiated services in an electric vehicle public charging station network. *IEEE Transactions on smart grid*, 11(2):1489–1499, 2019.

[32] Ahmadreza Moradipari, Sanae Amani, Mahnoosh Alizadeh, and Christos Thrampoulidis. Safe linear thompson sampling with side information. *IEEE Transactions on Signal Processing*, 69:3755–3767, 2021.

[33] Ahmadreza Moradipari, Nathaniel Tucker, and Mahnoosh Alizadeh. Mobility-aware electric vehicle fast charging load models with geographical price variations. *IEEE Transactions on Transportation Electrification*, 7(2):554–565, 2020.

[34] Ahmadreza Moradipari, Christos Thrampoulidis, and Mahnoosh Alizadeh. Stage-wise conservative linear bandits. *Advances in neural information processing systems*, 33:11191–11201, 2020.

[35] Nathaniel Tucker, Ahmadreza Moradipari, and Mahnoosh Alizadeh. Constrained thompson sampling for real-time electricity pricing with grid reliability constraints. *IEEE Transactions on Smart Grid*, 11(6):4971–4983, 2020.

[36] Ahmadreza Moradipari, Nathaniel Tucker, Tuo Zhang, Gustavo Cezar, and Mahnoosh Alizadeh. Mobility-aware smart charging of electric bus fleets. In *2020 IEEE Power & Energy Society General Meeting (PESGM)*, pages 1–5. IEEE, 2020.

[37] Ahmadreza Moradipari, Mahnoosh Alizadeh, and Christos Thrampoulidis. Linear thompson sampling under unknown linear constraints. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3392–3396. IEEE, 2020.

[38] Ahmadreza Moradipari, Cody Silva, and Mahnoosh Alizadeh. Learning to dynamically price electricity demand based on multi-armed bandits. In *2018 IEEE global conference on signal and information processing (GlobalSIP)*, pages 917–921. IEEE, 2018.

[39] Ahmadreza Moradipari and Mahnoosh Alizadeh. Pricing differentiated services in an electric vehicle public charging station network. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 6488–6494. IEEE, 2018.

[40] Ahmadreza Moradipari, Berkay Turan, Yasin Abbasi-Yadkori, Mahnoosh Alizadeh, and Mohammad Ghavamzadeh. Feature and parameter selection in stochastic linear bandits. In *International Conference on Machine Learning*, pages 15927–15958. PMLR, 2022.

[41] Ahmadreza Moradipari, Mohammad Ghavamzadeh, and Mahnoosh Alizadeh. Collaborative multi-agent stochastic linear bandits. In *2022 American Control Conference (ACC)*, pages 2761–2766. IEEE, 2022.

[42] Ahmadreza Moradipari, Yasin Abbasi-Yadkori, Mahnoosh Alizadeh, and Mohammad Ghavamzadeh. Parameter and feature selection in stochastic linear bandits. *arXiv preprint arXiv:2106.05378*, 2021.

[43] Ahmadreza Moradipari, Sina Shahsavari, Ashkan Esmaeili, and Farokh Marvasti. Using empirical covariance matrix in enhancing prediction accuracy of linear models with missing information. In *2017 International conference on sampling theory and applications (SampTA)*, pages 446–450. IEEE, 2017.

[44] Ahmadreza Moradipari, Sangjae Bae, Mahnoosh Alizadeh, Ehsan Moradi Pari, and David Isele. Predicting parameters for modeling traffic participants. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 703–708. IEEE, 2022.

[45] Ahmadreza Moradipari, Mohammad Ghavamzadeh, Taha Rajabzadeh, Christos Thrampoulidis, and Mahnoosh Alizadeh. Multi-environment meta-learning in stochastic linear bandits. In *2022 IEEE International Symposium on Information Theory (ISIT)*, pages 1659–1664. IEEE, 2022.

[46] Ahmadreza Moradipari, Mohammad Pedramfar, Modjtaba Shokrian Zini, and Vaneet Aggarwal. Improved bayesian regret bounds for thompson sampling in reinforcement learning. *Advances in Neural Information Processing Systems*, 36:23557–23569, 2023.

[47] Modjtaba Shokrian Zini, Mohammad Pedramfar, Matthew Riemer, Ahmadreza Moradipari, and Miao Liu. Coagent networks revisited. *arXiv preprint arXiv:2001.10474*, 2020.

[48] Amirhossein Afsharrad, Ahmadreza Moradipari, and Sanjay Lall. Convex methods for constrained linear bandits. In *2024 European Control Conference (ECC)*, pages 2111–2118. IEEE, 2024.

[49] Ahmadreza Moradipari, Sergei S Avedisov, and Hongsheng Lu. Benefits of intent sharing in cooperative platooning. In *2024 IEEE Vehicular Networking Conference (VNC)*, pages 195–202. IEEE, 2024.

[50] Lahiri Kanipakam, Ahmed Hamdi Sakr, Sergei S Avedisov, and Ahmadreza Moradipari. Cooperative adaptive cruise control based on intent sharing messages and reinforcement learning. In *2024 IEEE Vehicular Networking Conference (VNC)*, pages 188–194. IEEE, 2024.

[51] S Shiffman, AA Stone, and MR Hufford. Ecological momentary assessment. annual review of clinical psychology. *vol*, 4:1–32, 2007.

[52] Philip Schmidt, Attila Reiss, Robert Duerichen, and Kristof Van Laerhoven. Wearable affect and stress recognition: A review. *arXiv preprint arXiv:1811.08854*, 2018.

[53] Hamidreza Alikhani, Ziyu Wang, Anil Kanduri, Pasi Liljeberg, Amir M. Rahmani, and Nikil Dutt. Seal: Sensing efficient active learning on wearables through context-awareness. In *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1–2, 2024.

[54] Hillol Sarker, Matthew Tyburski, Md Mahbubur Rahman, Karen Hovsepian, Moushumi Sharmin, David H Epstein, Kenzie L Preston, C Debra Furr-Holden, Adam Milam, Inbal Nahum-Shani, et al. Finding significant stress episodes in a discontinuous time series of rapidly varying mobile sensor data. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pages 4489–4501, 2016.

[55] Elena Smets, Emmanuel Rios Velazquez, Giuseppina Schiavone, Imen Chakroun, Ellie D'Hondt, Walter De Raedt, Jan Cornelis, Olivier Janssens, Sofie Van Hoecke, Stephan Claes, et al. Large-scale wearable data reveal digital phenotypes for daily-life stress detection. *NPJ digital medicine*, 1(1):67, 2018.

[56] Andrea Gaggioli, Giovanni Pioggia, Gennaro Tartarisco, Giovanni Baldus, Daniele Corda, Pietro Cipresso, and Giuseppe Riva. A mobile data collection platform for mental health research. *Personal and Ubiquitous Computing*, 17:241–251, 2013.

[57] Yanxia Cheng, Saurabh Sharma, Prashant Sharma, and KMMCB Kulathunga. Role of personalization in continuous use intention of mobile news apps in india: Extending the utaut2 model. *Information*, 11(1):33, 2020.

[58] Akane Sano. *Measuring college students' sleep, stress, mental health and wellbeing with wearable sensors and mobile phones*. PhD thesis, Massachusetts Institute of Technology, 2016.

[59] Barbara Henker, Carol K Whalen, Larry D Jamner, and Ralph J Delfino. Anxiety, affect, and activity in teenagers: Monitoring daily life with electronic diaries. *Journal of the American Academy of Child & Adolescent Psychiatry*, 41(6):660–670, 2002.

[60] Hojoong Kim, Yun-Soung Kim, Musa Mahmood, Shinjae Kwon, Nathan Zavanelli, Hee Seok Kim, You Seung Rim, Fayron Epps, and Woon-Hong Yeo. Fully integrated, stretchable, wireless skin-conformal bioelectronics for continuous stress monitoring in daily life. *Advanced Science*, 7(15):2000810, 2020.

[61] Jochen Fahrenberg, Michael Myrtek, Kurt Pawlik, and Meinrad Perrez. Ambulatory assessment-monitoring behavior in daily life settings. *European Journal of Psychological Assessment*, 23(4):206–213, 2007.

[62] Jocelyn Lai, Amir Rahmani, Asal Yunusova, Alexander P Rivera, Sina Labbaf, Sirui Hu, Nikil Dutt, Ramesh Jain, and Jessica L Borelli. Using multimodal assessments to capture personalized contexts of college student well-being in 2020: Case study. *JMIR formative research*, 5(5):e26186, 2021.

[63] Rossana Castaldo, Luis Montesinos, Paolo Melillo, Sebastiano Massaro, and Leandro Pecchia. To what extent can we shorten hrv analysis in wearable sensing? a case study on mental stress detection. In *EMBEC & NBC 2017: Joint Conference of the European Medical and Biological Engineering Conference (EMBEC) and the Nordic-Baltic Conference on Biomedical Engineering and Medical Physics (NBC), Tampere, Finland, June 2017*, pages 643–646. Springer, 2018.

[64] Yuan Shi, Minh Hoai Nguyen, Patrick Blitz, Brian French, Scott Fisk, Fernando De la Torre, Asim Smailagic, Daniel P Siewiorek, Mustafa Al'Absi, Emre Ertin, et al. Personalized stress detection from physiological measurements. In *International symposium on quality of life technology*, volume 1, pages 28–29, 2010.

[65] BIOMEDIKAL.IN, [Online]. `http://biomedikal.in/2011/05/important-physiological-signals-in-the-body/`. Accessed: Oct-2020.

[66] Lan Li and Ji-hua Chen. Emotion recognition using physiological signals. In *International Conference on Artificial Reality and Telexistence*, pages 437–446. Springer, 2006.

[67] Lin Shu, Jinyan Xie, Mingyue Yang, Ziyi Li, Zhenqi Li, Dan Liao, Xiangmin Xu, and Xinyi Yang. A review of emotion recognition using physiological signals. *Sensors*, 18(7):2074, 2018.

[68] Hee Jeong Han, Sina Labbaf, Jessica L Borelli, Nikil Dutt, and Amir M Rahmani. Objective stress monitoring based on wearable sensors in everyday settings. *Journal of Medical Engineering & Technology*, 44(4):177–189, 2020.

[69] Ali Rostami, Vaibhav Pandey, Nitish Nag, Vesper Wang, and Ramesh Jain. Personal food model. In *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, page 4416–4424, New York, NY, USA, 2020. Association for Computing Machinery.

[70] Milad Asgari Mehrabadi, Iman Azimi, Fatemeh Sarhaddi, Anna Axelin, Hannakaisa Niela-Vilén, Saana Myllyntausta, Sari Stenholm, Nikil Dutt, Pasi Liljeberg, and Amir M Rahmani. Sleep tracking of a commercially available smart ring and smartwatch against medical-grade actigraphy in everyday settings: Instrument validation study. *JMIR mHealth and uHealth*, 2020.

[71] Mohammad Soleymani, Frank Villaro-Dixon, Thierry Pun, and Guillaume Chanel. Toolbox for emotional feature extraction from physiological signals (teap). *Frontiers in ICT*, 4:1, 2017.

[72] John Edison Muñoz, Elvio Rubio Gouveia, Mónica S Cameirão, and Sergi Bermúdez i Badia. Physiolab-a multivariate physiological computing toolbox for ecg, emg and eda signals: a case of study of cardiorespiratory fitness assessment in the elderly population. *Multimedia Tools and Applications*, 77(9):11521–11546, 2018.

[73] Jens Blechert, Peter Peyk, Michael Liedlgruber, and Frank H Wilhelm. Anslab: Integrated multichannel peripheral biosignal processing in psychophysiological science. *Behavior Research Methods*, 48(4):1528–1545, 2016.

[74] Dominique Makowski, Tam Pham, Zen, Jan C. Brammer, Duy Le, Hung Pham (Pham Tien Hùng), François Lesspinasse, Chuan-Peng Hu, and Christopher Schölzel. neuropsychology/neurokit: 0.0.6, January 2020.

[75] Giulio Gabrieli, Atiqah Azhari, and Gianluca Esposito. Pysiology: A python package for physiological feature extraction. In *Neural Approaches to Dynamics of Signal Exchanges*, pages 395–402. Springer, 2020.

[76] Galvanic Skin Response. The complete pocket guide. *Imotions–Biometric Research, Simplified*, 2017.

[77] Pamela Zontone, Antonio Affanni, Riccardo Bernardini, Alessandro Piras, and Roberto Rinaldo. Stress detection through electrodermal activity (eda) and electrocardiogram (ecg) analysis in car drivers. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5. IEEE, 2019.

[78] Busra T Susam, Murat Akcakaya, Hooman Nezamfar, Damaris Diaz, Xiaojing Xu, Virginia R de Sa, Kenneth D Craig, Jeannie S Huang, and Matthew S Goodwin. Automated pain assessment using electrodermal activity data and machine learning. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 372–375. IEEE, 2018.

[79] Dian Yu and Shouqian Sun. A systematic exploration of deep neural networks for eda-based emotion recognition. *Information*, 11(4):212, 2020.

[80] Beanbonyka Rim, Nak-Jun Sung, Sedong Min, and Min Hong. Deep learning in physiological signal data: A survey. *Sensors*, 20(4):969, 2020.

[81] Amir Hosein Afandizadeh Zargari, Manik Dautta, Marzieh Ashrafiamiri, Minjun Seo, Peter Tseng, and Fadi Kurdahi. Newertrack: Ml-based accurate tracking of in-mouth nutrient sensors position using spectrum-wide information. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(11):3833–3841, 2020.

[82] Milad Asgari Mehrabadi, Seyed Amir Hossein Aqajari, Iman Azimi, Charles A Downs, Nikil Dutt, and Amir M Rahmani. Detection of covid-19 using heart rate and blood pressure: Lessons learned from patients with ards. *arXiv preprint arXiv:2011.10470*, 2020.

[83] Marzieh Ashrafiamiri, Sai Manoj Pudukotai Dinakarrao, Amir Hosein Afandizadeh Zargari, Minjun Seo, Fadi Kurdahi, and Houman Homayoun. R2ad: Randomization and reconstructor-based adversarial defense on deep neural network. In *Proceedings of the 2020 ACM/IEEE Workshop on Machine Learning for CAD*, pages 21–26, 2020.

[84] pyEDA, [Online]. `https://github.com/HealthSciTech/pyEDA`. Accessed: Oct-2020.

[85] Heera Lee and Andrea Kleinsmith. Public speaking anxiety in a real classroom: Towards developing a reflection system. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2019.

[86] Philip Schmidt, Attila Reiss, Robert Duerichen, Claus Marberger, and Kristof Van Laerhoven. Introducing wesad, a multimodal dataset for wearable stress and affect

detection. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, pages 400–408, 2018.

[87] Alberto Greco, Gaetano Valenza, Antonio Lanata, Enzo Pasquale Scilingo, and Luca Citi. cvxeda: A convex optimization approach to electrodermal activity processing. *IEEE Transactions on Biomedical Engineering*, 63(4):797–804, 2015.

[88] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[89] Quoc V Le et al. A tutorial on deep learning part 2: Autoencoders, convolutional neural networks and recurrent neural networks. *Google Brain*, pages 1–20, 2015.

[90] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

[91] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[92] David J Hand and Keming Yu. Idiot's bayes—not so stupid after all? *International statistical review*, 69(3):385–398, 2001.

[93] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[94] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[95] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[96] empatica-e4, [Online]. `https://www.empatica.com/research/e4/`. Accessed: Oct-2020.

[97] John Allen. Photoplethysmography and its application in clinical physiological measurement. *Physiological measurement*, 28(3):R1, 2007.

[98] Shao Hanyu and Chen Xiaohui. Motion artifact detection and reduction in ppg signals based on statistics analysis. In *2017 29th Chinese control and decision conference (CCDC)*, pages 3114–3119. IEEE, 2017.

[99] Syed Khairul Bashar, Dong Han, Apurv Soni, David D McManus, and Ki H Chon. Developing a novel noise artifact detection algorithm for smartphone ppg signals: Preliminary results. In *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, pages 79–82. IEEE, 2018.

[100] Wei-Jheng Lin and Hsi-Pin Ma. A physiological information extraction method based on wearable ppg sensors with motion artifact removal. In *2016 IEEE international conference on communications (ICC)*, pages 1–6. IEEE, 2016.

[101] M Raghuram, Kosaraju Sivani, and K Ashoka Reddy. Use of complex emd generated noise reference for adaptive reduction of motion artifacts from ppg signals. In *2016 international conference on electrical, electronics, and optimization techniques (ICEEOT)*, pages 1816–1820. IEEE, 2016.

[102] Shinsuke Hara, Takunori Shimazaki, Hiroyuki Okuhata, Hajime Nakamura, Takashi Kawabata, Kai Cai, and Tomohito Takubo. Parameter optimization of motion artifact canceling ppg-based heart rate sensor by means of cross validation. In *2017 11th international symposium on medical information and communication technology (ISMICT)*, pages 73–76. IEEE, 2017.

[103] Stefan CB Mannsfeld, Benjamin CK Tee, Randall M Stoltenberg, Christopher V Chen, Soumendra Barman, Beinn VO Muir, Anatoliy N Sokolov, Colin Reese, and Zhenan Bao. Highly sensitive flexible pressure sensors with microstructured rubber dielectric layers. *Nature materials*, 9(10):859–864, 2010.

[104] Alireza Nikzamir and Filippo Capolino. Highly sensitive coupled oscillator based on an exceptional point of degeneracy and nonlinearity. *arXiv preprint arXiv:2206.04031*, 2022.

[105] Khawaja Taimoor Tanweer, Syed Rafay Hasan, and Awais Mehmood Kamboh. Motion artifact reduction from ppg signals during intense exercise using filtered x-lms. In *2017 IEEE international symposium on circuits and systems (ISCAS)*, pages 1–4. IEEE, 2017.

[106] Chih-Chin Wu, I-Wei Chen, and Wai-Chi Fang. An implementation of motion artifacts elimination for ppg signal processing based on recursive least squares adaptive filter. In *2017 IEEE biomedical circuits and systems conference (BioCAS)*, pages 1–4. IEEE, 2017.

[107] Alessandro Baca, Giorgio Biagetti, Marta Camilletti, Paolo Crippa, Laura Falaschetti, Simone Orcioni, Luca Rossini, Dario Tonelli, and Claudio Turchetti. Carma: a robust motion artifact reduction algorithm for heart rate monitoring from ppg signals. In *2015 23rd European signal processing conference (EUSIPCO)*, pages 2646–2650. IEEE, 2015.

[108] Rozhin Yasaei, Felix Hernandez, and Mohammad Abdullah Al Faruque. Iot-cad: context-aware adaptive anomaly detection in iot systems through sensor association. In *2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pages 1–9. IEEE, 2020.

[109] Rozhin Yasaei, Luke Chen, Shih-Yuan Yu, and Mohammad Abdullah Al Faruque. Hardware trojan detection using graph neural networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2022.

[110] Farzam Ebrahimnejad and James R Lee. Multiscale entropic regularization for mts on general metric spaces. *arXiv preprint arXiv:2111.10908*, 2021.

[111] Sina Shahsavari, Pulak Sarangi, and Piya Pal. Kr-lista: Re-thinking unrolling for covariance-driven sparse inverse problems. In *2021 55th Asilomar Conference on Signals, Systems, and Computers*, pages 1403–1408. IEEE, 2021.

[112] Kushal Joshi, Alireza Javani, Joshua Park, Vanessa Velasco, Binzhi Xu, Olga Razorenova, and Rahim Esfandyarpour. A machine learning-assisted nanoparticle-printed biochip for real-time single cancer cell analysis. *Advanced Biosystems*, 4(11):2000160, 2020.

[113] Milad Asgari Mehrabadi, Seyed Amir Hossein Aqajari, Amir Hosein Afandizadeh Zargari, Nikil Dutt, and Amir M Rahmani. Novel blood pressure waveform reconstruction from photoplethysmography using cycle generative adversarial networks. *arXiv e-prints*, pages arXiv–2201, 2022.

[114] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang. Image blind denoising with generative adversarial network based noise modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3155–3164, 2018.

[115] Linh Duy Tran, Son Minh Nguyen, and Masayuki Arai. Gan-based noise model for denoising real images. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[116] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[117] Monalisa Singha Roy, Rajarshi Gupta, Jayanta K Chandra, Kaushik Das Sharma, and Arunansu Talukdar. Improving photoplethysmographic measurements under motion artifacts using artificial neural network for personal healthcare. *IEEE Transactions on Instrumentation and Measurement*, 67(12):2820–2829, 2018.

[118] Marco AF Pimentel, Alistair EW Johnson, Peter H Charlton, Drew Birrenkott, Peter J Watkinson, Lionel Tarassenko, and David A Clifton. Toward a robust estimation of respiratory rate from pulse oximeters. *IEEE Transactions on Biomedical Engineering*, 64(8):1914–1923, 2016.

[119] Empatica — medical devices, ai and algorithms for remote patient monitoring. `https://www.empatica.com/`. Accessed: 2021-05-24.

[120] Mert Sevil, Mudassir Rashid, Iman Hajizadeh, Mohammad Reza Askari, Nicole Hobbs, Rachel Brandt, Minsun Park, Laurie Quinn, and Ali Cinar. Discrimination of simultaneous psychological and physical stressors using wristband biosignals. *Computer Methods and Programs in Biomedicine*, 199:105898, 2021.

[121] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[122] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[123] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[124] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European conference on computer vision*, pages 702–716. Springer, 2016.

[125] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[126] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[127] David Pollreisz and Nima TaheriNejad. Detection and removal of motion artifacts in ppg signals. *Mobile Networks and Applications*, pages 1–11, 2019.

[128] Mohammad Reza Askari, Mudassir Rashid, Mert Sevil, Iman Hajizadeh, Rachel Brandt, Sediqeh Samadi, and Ali Cinar. Artifact removal from data generated by nonlinear systems: Heart rate estimation from blood volume pulse signal. *Industrial & Engineering Chemistry Research*, 59(6):2318–2327, 2019.

[129] Analog devices — adxl343. `https://www.analog.com/media/en/technical-documentation/data-sheets/adxl343.pdf`.

[130] Smartpower2 5vdc power supply. `https://ameridroid.com/products/smartpower2-5vdc-power-supply`.

[131] my.clevelandclinic.org. `https://my.clevelandclinic.org/health/articles/`. [Online; accessed Mar-2021].

[132] Sandy Rolfe. The importance of respiratory rate monitoring. *BJN*, 2019.

[133] Simon John Rankin Cooper et al. Respiratory rate records: the repeated rate? *JCN*, 2014.

[134] Lisa Dougherty et al. *The Royal Marsden manual of clinical nursing procedures.* John Wiley & Sons, 2015.

[135] Michelle A Cretikos et al. Respiratory rate: the neglected vital sign. *MJA*, 2008.

[136] Andrea Nicolò et al. The importance of respiratory rate monitoring: From healthcare to sport and exercise. *Sensors*, 2020.

[137] Mikko Pirhonen et al. Acquiring respiration rate from photoplethysmographic signal by recursive bayesian tracking of intrinsic modes in time-frequency spectra. *Sensors*, 2018.

[138] Peter H Charlton et al. Breathing rate estimation from the electrocardiogram and photoplethysmogram: A review. *IEEE RBME*, 2017.

[139] Peter H Charlton et al. An assessment of algorithms to estimate respiratory rate from the electrocardiogram and photoplethysmogram. *Physiological measurement*, 2016.

[140] Amir Hosein Afandizadeh Zargari et al. An accurate non-accelerometer-based ppg motion artifact removal technique using cyclegan. *arXiv preprint arXiv:2106.11512*, 2021.

[141] Dayi Bian et al. Respiratory rate estimation using ppg: A deep learning approach. In *2020 42nd Annual International Conference of the IEEE EMBC*, 2020.

[142] Jun-Yan Zhu et al. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *2017 IEEE ICCV*, 2017.

[143] Marco AF Pimentel et al. Toward a robust estimation of respiratory rate from pulse oximeters. *IEEE TBME*, 2016.

[144] Drew A Birrenkott et al. A robust fusion model for estimating respiratory rate from photoplethysmography and electrocardiography. *IEEE TBME*, 2017.

[145] Delaram Jarchi et al. Accelerometry-based estimation of respiratory rate for post-intensive care patient monitoring. *IEEE Sensors Journal*, 2018.

[146] Ian J Goodfellow et al. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[147] Carlos Carreiras. BioSPPy: Biosignal processing in Python, 2015–.

[148] Justin Johnson et al. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*. Springer, 2016.

[149] Kaiming He et al. Deep residual learning for image recognition. In *Proceedings of the IEEE CVPR*, 2016.

[150] Phillip Isola et al. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on CVPR*, 2017.

[151] Chuan Li et al. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *ECCV*. Springer, 2016.

[152] Christian Ledig et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE CVPR*, 2017.

[153] Torben Noto et al. Automated analysis of breathing waveforms using breathmetrics: a respiratory signal processing toolbox. *Chemical senses*, 2018.

[154] Walter Karlen et al. Multiparameter respiratory rate estimation from the photoplethysmogram. *IEEE TBME*, 2013.

[155] Alexander M Chan et al. Ambulatory respiratory rate detection using ecg and a triaxial accelerometer. In *2013 35th Annual International Conference of the IEEE EMBC*.

[156] Russell Li and Zhandong Liu. Stress detection using deep neural networks. *BMC Medical Informatics and Decision Making*, 20:1–10, 2020.

[157] Yekta Said Can, Niaz Chalabianloo, Deniz Ekiz, and Cem Ersoy. Continuous stress detection using wearable sensors in real life: Algorithmic programming contest case study. *Sensors*, 19(8):1849, 2019.

[158] Syem Ishaque, Alice Rueda, Binh Nguyen, Naimul Khan, and Sridhar Krishnan. Physiological signal analysis and classification of stress from virtual reality video game. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 867–870. IEEE, 2020.

[159] Peter H Charlton, Patrick Celka, Bushra Farukh, Phil Chowienczyk, and Jordi Alastruey. Assessing mental stress from the photoplethysmogram: a numerical study. *Physiological measurement*, 39(5):054001, 2018.

[160] John Allen. Photoplethysmography and its application in clinical physiological measurement. *Physiological measurement*, 28(3):R1, 2007.

[161] Panagiotis Kostopoulos, Athanasios I Kyritsis, Michel Deriaz, and Dimitri Konstantas. Stress detection using smart phone data. In *eHealth 360°: International Summit on eHealth, Budapest, Hungary, June 14-16, 2016, Revised Selected Papers*, pages 340–351. Springer, 2017.

[162] Dongyeol Seok, Sanghyun Lee, Minjae Kim, Jaeouk Cho, and Chul Kim. Motion artifact removal techniques for wearable eeg and ppg sensor systems. *Frontiers in Electronics*, 2:685513, 2021.

[163] Fanny Larradet, Radoslaw Niewiadomski, Giacinto Barresi, Darwin G Caldwell, and Leonardo S Mattos. Toward emotion recognition from physiological signals in the wild: approaching the methodological issues in real-life data collection. *Frontiers in psychology*, 11:1111, 2020.

[164] Hee Jeong Han, Sina Labbaf, Jessica L Borelli, Nikil Dutt, and Amir M Rahmani. Objective stress monitoring based on wearable sensors in everyday settings. *Journal of Medical Engineering & Technology*, 44(4):177–189, 2020.

[165] Seyed Amir Hossein Aqajari, Emad Kasaeyan Naeini, Milad Asgari Mehrabadi, Sina Labbaf, Nikil Dutt, and Amir M Rahmani. pyeda: An open-source python toolkit for pre-processing and feature extraction of electrodermal activity. *Procedia Computer Science*, 184:99–106, 2021.

[166] Hyun-Myung Cho, Heesu Park, Suh-Yeon Dong, and Inchan Youn. Ambulatory and laboratory stress detection based on raw electrocardiogram signals using a convolutional neural network. *Sensors*, 19(20):4408, 2019.

[167] Han Yu and Akane Sano. Semi-supervised learning and data augmentation in wearable-based momentary stress detection in the wild. *arXiv preprint arXiv:2202.12935*, 2022.

[168] Ramesh Kumar Sah, Michael McDonell, Patricia Pendry, Sara Parent, Hassan Ghasemzadeh, and Michael J Cleveland. Adarp: A multi modal dataset for stress and alcohol relapse quantification in real life setting. In *2022 IEEE-EMBS International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pages 1–4. IEEE, 2022.

[169] Ali Tazarv, Sina Labbaf, Stephanie M Reich, Nikil Dutt, Amir M Rahmani, and Marco Levorato. Personalized stress monitoring using wearable sensors in everyday settings. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 7332–7335. IEEE, 2021.

[170] Samuel L Battalio, David E Conroy, Walter Dempsey, Peng Liao, Marianne Menictas, Susan Murphy, Inbal Nahum-Shani, Tianchen Qian, Santosh Kumar, and Bonnie Spring. Sense2stop: a micro-randomized trial using wearable sensors to optimize a just-in-time-adaptive stress management intervention for smoking relapse prevention. *Contemporary Clinical Trials*, 109:106534, 2021.

[171] Yekta Said Can, Niaz Chalabianloo, Deniz Ekiz, and Cem Ersoy. Continuous stress detection using wearable sensors in real life: Algorithmic programming contest case study. *Sensors*, 19(8):1849, 2019.

[172] Weichen Wang, Shayan Mirjafari, Gabriella Harari, Dror Ben-Zeev, Rachel Brian, Tanzeem Choudhury, Marta Hauser, John Kane, Kizito Masaba, Subigya Nepal, et al. Social sensing: assessing social functioning of patients living with schizophrenia using mobile phone sensing. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–15, 2020.

[173] Han Yu and Akane Sano. Passive sensor data based future mood, health, and stress prediction: User adaptation using deep learning. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 5884–5887. IEEE, 2020.

[174] Karel Mundnich, Brandon M Booth, Michelle l'Hommedieu, Tiantian Feng, Benjamin Girault, Justin L'hommedieu, Mackenzie Wildman, Sophia Skaaden, Amrutha Nadarajan, Jennifer L Villatte, et al. Tiles-2018, a longitudinal physiologic and behavioral data set of hospital workers. *Scientific Data*, 7(1):354, 2020.

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

[176] Paul Van Gent, Haneen Farah, Nicole Van Nes, and Bart Van Arem. Heartpy: A novel heart rate algorithm for the analysis of noisy signals. *Transportation research part F: traffic psychology and behaviour*, 66:368–378, 2019.

[177] Gustavo EAPA Batista, Maria Carolina Monard, et al. A study of k-nearest neighbour as an imputation method. *His*, 87(251-260):48, 2002.

[178] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

[179] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[180] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[181] Selmer C Larson. The shrinkage of the coefficient of multiple correlation. *Journal of Educational Psychology*, 22(1):45, 1931.

[182] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[183] The american institute of stresss, 2021.

[184] Mayo clinic, 2023.

[185] Peter H Charlton et al. Assessing mental stress from the photoplethysmogram: a numerical study. *Physiological measurement*, 39(5):054001, 2018.

[186] Yuanfan Yao et al. Privacy-preserving and energy efficient task offloading for collaborative mobile computing in iot: An admm approach. *Computers & Security*, 96:101886, 2020.

[187] Anil Kanduri, Sina Shahhosseini, Emad Kasaeyan Naeini, Hamidreza Alikhani, Pasi Liljeberg, Nikil Dutt, and Amir M Rahmani. Edge-centric optimization of multimodal ml-driven ehealth applications. In *Embedded Machine Learning for Cyber-Physical, IoT, and Edge Computing: Use Cases and Emerging Challenges*, pages 95–125. Springer, 2023.

[188] Hamidreza Alikhani, Anil Kanduri, Pasi Liljeberg, Amir M Rahmani, and Nikil Dutt. Dynafuse: Dynamic fusion for resource efficient multi-modal machine learning inference. *IEEE Embedded Systems Letters*, 2023.

[189] Denisse Castaneda et al. A review on wearable photoplethysmography sensors and their potential future applications in health care. *International journal of biosensors & bioelectronics*, 4(4):195, 2018.

[190] Giovanna Sannino et al. A mobile system for real-time context-aware monitoring of patients' health and fainting. *International journal of data mining and bioinformatics*, 10(4):407–423, 2014.

[191] Marija Stojchevska et al. Assessing the added value of context during stress detection from wearable data. *BMC Medical Informatics and Decision Making*, 22(1):268, 2022.

[192] Yekta Said Can et al. Real-life stress level monitoring using smart bands in the light of contextual information. *IEEE Sensors Journal*, 20(15):8721–8730, 2020.

[193] Jochen Fahrenberg, Michael Myrtek, Kurt Pawlik, and Meinrad Perrez. Ambulatory assessment-monitoring behavior in daily life settings. *European Journal of Psychological Assessment*, 23(4):206–213, 2007.

[194] Andrew Steptoe, Sabine Kunz-Ebrecht, Natalie Owen, Pamela J Feldman, Gonneke Willemsen, Clemens Kirschbaum, and Michael Marmot. Socioeconomic status and stress-related biological responses over the working day. *Psychosomatic medicine*, 65(3):461–470, 2003.

[195] Lora E Burke, Saul Shiffman, Edvin Music, Mindi A Styn, Andrea Kriska, Asim Smailagic, Daniel Siewiorek, Linda J Ewing, Eileen Chasens, Brian French, et al. Ecological momentary assessment in behavioral research: addressing technological and human participant challenges. *Journal of medical Internet research*, 19(3):e77, 2017.

[196] Fanny Larradet et al. Toward emotion recognition from physiological signals in the wild: approaching the methodological issues in real-life data collection. *Frontiers in psychology*, 11:1111, 2020.

[197] Ali Tazarv et al. Active reinforcement learning for personalized stress monitoring in everyday settings. *arXiv preprint arXiv:2305.00111*, 2023.

[198] Sheldon Cohen, Tom Kamarck, Robin Mermelstein, et al. Perceived stress scale. *Measuring stress: A guide for health and social scientists*, 10(2):1–2, 1994.

[199] Thomas H Holmes and Richard H Rahe. The social readjustment rating scale. *Journal of psychosomatic research*, 1967.

[200] Giorgos Giannakakis, Dimitris Grigoriadis, Katerina Giannakaki, Olympia Simantiraki, Alexandros Roniotis, and Manolis Tsiknakis. Review on psychological stress detection using biosignals. *IEEE Transactions on Affective Computing*, 13(1):440–460, 2019.

[201] Burr Settles. Active learning literature survey. 2009.

[202] Gulin Dogan et al. Stress detection using experience sampling: A systematic mapping study. *International Journal of Environmental Research and Public Health*, 19(9):5693, 2022.

[203] Han Yu et al. Semi-supervised learning and data augmentation in wearable-based momentary stress detection in the wild. *arXiv preprint arXiv:2202.12935*, 2022.

[204] H Yu et al. Passive sensor data based future mood health and stress prediction: User adaptation using deep learning; passive sensor data based future mood health and stress prediction: User adaptation using deep learning. 2020.

[205] Karel Mundnich et al. Tiles-2018, a longitudinal physiologic and behavioral data set of hospital workers. *Scientific Data*, 7(1):354, 2020.

[206] Weichen Wang et al. Social sensing: assessing social functioning of patients living with schizophrenia using mobile phone sensing. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–15, 2020.

[207] Samuel L Battalio et al. Sense2stop: a micro-randomized trial using wearable sensors to optimize a just-in-time-adaptive stress management intervention for smoking relapse prevention. *Contemporary Clinical Trials*, 109:106534, 2021.

[208] Ali Tazarv et al. Personalized stress monitoring using wearable sensors in everyday settings. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 7332–7335. IEEE, 2021.

[209] Dongyeol Seok et al. Motion artifact removal techniques for wearable eeg and ppg sensor systems. *Frontiers in Electronics*, 2:685513, 2021.

[210] Labbaf Sina et al. Zotcare: a flexible, personalizable, and affordable mhealth service provider. *Front. Digit. Health*, 2023.

[211] Fatemeh Sarhaddi et al. A comprehensive accuracy assessment of samsung smartwatch heart rate and heart rate variability. *PloS one*, 17(12):e0268361, 2022.

[212] Geetika Vashisht et al. A study on the tizen operating system. *International Journal of Computer Trends and Technology*, 12(1):14–15, 2014.

[213] Aware framework, 2023.

[214] Paul Van Gent et al. Heartpy: A novel heart rate algorithm for the analysis of noisy signals. *Transportation research part F: traffic psychology and behaviour*, 66:368–378, 2019.

[215] Sotiris B Kotsiantis et al. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160(1):3–24, 2007.

[216] Todd Hester et al. Deep q-learning from demonstrations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.1, 2018.

[217] Matthias Plappert. keras-rl. `https://github.com/keras-rl/keras-rl`, 2016.

[218] Marti A. Hearst et al. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

[219] Tianqi Chen et al. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4, 2015.

[220] Daniel Berrar et al. Cross-validation., 2019.

Appendix A

Abbreviations

ANS	Autonomic Nervous System
APS	Acute Psychological Stress
ARDS	Acute Respiratory Distress Syndrome
BPM	Beats Per Minute
BR	Breath Rate
DC	Direct Current
DWT	Discrete Wavelet Transform
ECG	Electrocardiogram
EDA	Electrodermal Activity
EMBC	Engineering in Medicine & Biology Society
EMA	Ecological Momentary Assessment
HR	Heart Rate
HRV	Heart Rate Variability
IRB	Institutional Review Board
KNN	K-Nearest Neighbors
LMS	Least Mean Squares
MAE	Mean Absolute Error
MEMS	Micro-Electro-Mechanical Systems
N/A	Not Applicable
NLMS	Normalized Least Mean Squares
PPG	Photoplethysmography
PPE	Peak-Peak Error
PRT	Pulse Respiration Time
RLS	Recursive Least Squares
RR	Respiratory Rate
S-EMA	Smart EMA
SC	Skin Conductance
SD	Standard Deviation
SD1	Standard Deviation 1
SD2	Standard Deviation 2
SDNN	Standard Deviation of Normal to Normal
SVD	Singular Value Decomposition
UI	User Interface
VFCDM	Variable Frequency Complex Demodulation
WESAD	Wearable Stress and Affect Detection