APPROVAL SHEET

Title of Thesis: Architecture Exploration for Low-Power Wearable Stress Detection Processor

Name of Candidate: Nasrin Attaran
Master of Science, 2017

Thesis and Abstract Approved: ______________________
Tinoosh Mohsenin
Assistant Professor
Department of Computer Science and
Electrical Engineering

Date Approved: 05/01/2017

ABSTRACT

Title of Thesis: Architecture Exploration for Low-Power Wearable Stress Detection Processor

Name of Candidate: Nasrin Attaran, Master of Science, 2017

Thesis directed by: Professor Tinoosh Mohsenin
Department of Computer Science and
Electrical Engineering

Personal monitoring systems can offer effective solutions for human health. These systems require sampling and significant processing on multiple streams of physiological signals to extract meaningful knowledge. The processing typically consists of feature extraction, data fusion, and classification stages, which require a large number of digital signal processing and machine learning kernels. In order to be used in a wearable environment, the processing system needs to be low-power, real-time and light-weight. In this thesis, we present a personalized stress monitoring processor that can meet these requirements. A dataset provided by Army Research Laboratory (ARL) that contains multi-physiological signals is used for design exploration. Various physiological features are explored to maximize stress detection accuracy using two machine learning classifiers including Support Vector Machine (SVM) and K-Nearest Neighbors (KNN).

Among different extracted features from four physiological sensors, heart rate and accelerometer features have 96.8% and 95.8% detection accuracy for SVM and KNN classifiers, respectively.

Two fully flexible and multi-modal processing hardware were designed which consist of feature extraction and classification algorithms using SVM and KNN for stress monitoring. We demonstrate the ASIC post-layout implementation of both designs in 65 nm CMOS technology. The proposed SVM processor occupies 0.17 mm^2 and dissipates 39.4 mW at 250 MHz. The KNN processor has an area of 0.3 mm^2 and consumes 76.69 mW at 250 MHz.

Next, we explore the choice of low-power programmable embedded processors for energy-efficient processing of physiological signals for a wearable multi-modal stress detection system. The entire system which consists of feature extraction and classification for all 15 participants' data, which is implemented on a number platforms including Artix-7 FPGA, NVIDIA TK1 ARM-A15 CPU and Kepler GPU, and a domain-specific many core named Power Efficient Nano Clusters (PENC). The comparison of performance metrics among all platforms shows that PENC architecture has the highest throughput (decision/sec) over all platforms due to existence of task-level and data-level parallelism present in its architecture. PENC improves the throughput by 4.6x and 4.05x over the Artix FPGA for the KNN and SVM implementations respectively. The experimental results also indicate that for a larger design such as KNN with 16K training data, PENC accelerator is the most energy efficient platform. For KNN implementation, PENC manycore improves the energy efficiency by 4.7x and 268x over the FPGA and GPU, respectively. However, for the SVM implementation with 6000 support vectors as a smaller design, the FPGA improves the energy efficiency by 1.2x and 630x over the PENC and GPU, respectively. These findings suggest that the PENC manycore can be used as an energy-efficient, programmable and real-time platform for biomedical applications with large amount of data and similar computation-intensive parallel processing.

Architecture Exploration for Low-Power Wearable Stress Detection Processor

by

Nasrin Attaran

Thesis submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
Master of Science
2017

I would like to dedicate this thesis to my family. My kind and lovely husband Ali, and my supportive parents have been very encouraging throughout my graduate career. Without their unconditional love and support I would not have been able to come this far.

ACKNOWLEDGMENTS

I owe gratitude to all those who have made this thesis possible. Firstly, I would like to thank my advisor, Dr. Tinoosh Mohsenin, for her guidance and support. Her experience and knowledge has helped me in my research work and I would like to thank her for giving me this opportunity to work with her. I would also like to thank my committee members, Dr. Tulay Adali, Dr. Tim Oates, and Dr. Justin Brooks for all their feedback and guidance. Their feedback has been very valuable to me and my research work. I would like to thank Debbie Patton and Justin Brooks from the army research laboratory for providing the dataset in this work and their great help. I would also thank my lab mates, Abhilash and Sri for their contributions. Finally, I would like to thank all my family, friends for their support. Without their encouragement, I would not have been able to accomplish this.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Motivation

In daily life, stress is a normal reaction of the human body to external events of different kinds. However, if this reaction is too great or if it lasts too long, there is a risk of it resulting in physical or mental disorders. To avoid the critical effects of stress, continuous estimation of the stress level of individuals could provide an early warning.

With the vast improvement of semiconductor technology, the usage of personalize wearable biomedical devices has become significantly more popular. Personalized biomedical systems provide the ability to monitor multi-physiological signals, perform data fusion and processing and real time analysis. They have potential impacts on long-term health assessment and medical intervention.

In wearable health monitoring devices, accuracy and the individuals' convenience are the main priorities. To fulfill the patient's convenience requirement, the power consumption, which directly translates to the battery lifetime and size, must be kept as low as possible. Meanwhile, adopted improvements in power consumption should not impact accuracy. Therefore, reducing the energy consumption of these devices has already been the subject of a significant amount of research in the past few years [AHR] [LAC] [Viseh2014]. Transmitting all raw data to a computer to execute the required processing needs vast amount of communication, which leads to considerable power consumption and the need for a bulky battery (which hinders the device's practicality and the individuals' convenience).

Wearable devices need local processors to process the raw data and either transmit the important extracted features or the final classification result. Applying a local processor to the system helps reduce the power consumption and consequently, the size and weight of the device. Naturally, designing a very energy efficient on-board processor becomes challenging.

Figure1.1 compares power consumption and battery life of raw data transmission through local processing with the post layout implementation of a personalized stress detection system. This system uses the heart rate and three accelerometer signals with 100 Hz sampling rate and resolution of 16 bits. The local processing with one low-power ASIC processor decreases the total power consumption by 81% compared to raw data transmission. Furthermore using the local processing increases the battery life by 66% over raw data transmission.

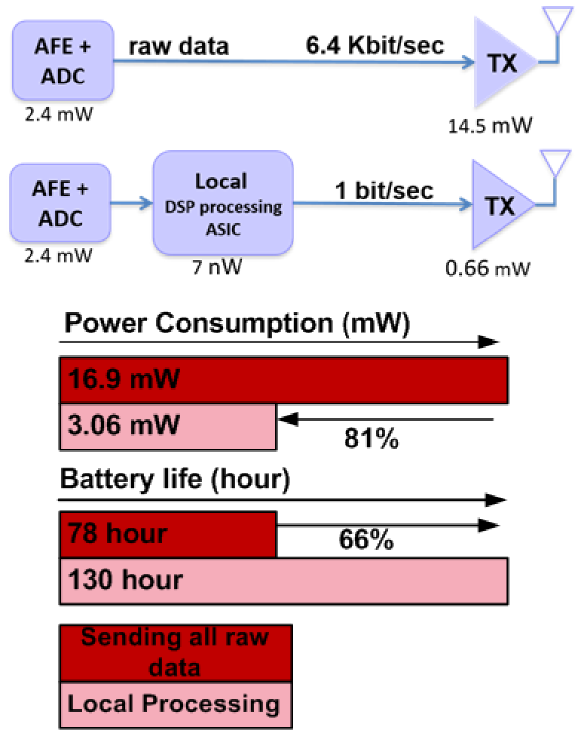


FIG. 1.1: The comparison between raw data transmission and local processing in aspect of power consumption (mW) and battery life ($hour$). For the entire system power consumption decreases by 81% and battery life increases by 66%.

1.2 Contributions

In this thesis, we propose a low-power reconfigurable processor which performs all signals processing at the sensor level for the case of wearable stress detection. For this purpose we investigated different physiological sensors and their corresponding features. We choose the most appropriate features set based on classification accuracy to design and implement the accurate stress detection processor.

We evaluated two popular machine learning classifiers, SVM and KNN, for stress monitoring in terms of accuracy, execution time and power consumption requirements. We implemented efficient personalized stress monitoring processors for two classifier using standard cell post-place and route flow in 65 nm CMOS technology.

We explored the programmable accelerators for generalized models of KNN and SVM versions of stress detection application using Artix FPGA and NVIDIA TK1 CPU and GPU as well. In response to all computing challenges of biomedical application, we utilized the a programmable energy-efficient, domain-specific accelerator named Power Efficient Nano Cluster (PENC) to map both machine learning kernels for the case study of stress monitoring application. This research provides analysis in terms of performance and resource utilization of the DSP and machine learning kernels on ASIC, FPGA, PENC manycore and multicore CPU and GPU based state-of-the-art embedded computing platforms.

1.3 Organization of Thesis

In the second chapter, we review concepts and previous research on stress monitoring. We discuss and analyze the different sections of the multi-modal stress detection in chapter 3. Chapter 4 describes the implementation results of wearable stress detection using several hardware-based and software-based platforms including ASIC, FPGA, PENC manycore, CPU and GPU. Chapter 5 concludes the thesis with the discussion of future work.

Chapter 2

BACKGROUND

2.1 Stress

Stress is a physiological response to the mental, emotional, and physical challenges that everyone encounters in daily life [Sun *et al.*2010]. There are strong links between stress and overall health, concentration and ability to perform tasks. In modern society, more and more people are suffering from stress. More than 56% of Americans reported stress as a source of personal health problems [Deng *et al.*2012]. 94% of adults believe that stress can contribute to the development of major illnesses, such as heart disease, depression and obesity, and that some types of stress can trigger heart attacks, arrhythmias and even sudden death (particularly in people who already have cardiovascular disease) [Ollander2015]. Identifying the stress of human being by using physiological sensors has been a hot research topic in recent years [Deng *et al.*2012].

2.2 Physiological Signals for Monitoring Stress

Various physiological signals are used for continuously measuring stress, including galvanic skin resistance (GSR), skin temperature, electrocardiogram (ECG), electroencephalography (electrical activity of brain, EEG), respiration (RESP), electromyography (EMG), oxygen saturation (SpO_2), blood pressure, pupil diameter and accelerometer (ACC) [Zhai & Barreto2006] [Riera *et al.*2012] [Wac & Tsiourti2014]. Thus there is a large choice of possible signals that one can acquire from the human body including measuring properties of the eye, the face, the brain, the muscles, the skin, the heart and even movement of the body as a whole (Figure 2.1).

2.2.0.1 Electrocardiogram (ECG) An ECG records the electrical activity of the heart using electrodes placed upon the body (Figure 2.2). The ECG signal is usually periodic, consisting of three parts: the P wave, the QRS complex and the T wave. The graphical representation of ECG is shown in Figure 2.3. The primary purpose of the ECG is to calculate the heart rate (HR), normally done through the inter-beat intervals (IBI) of the R waves. The heart rate variability (HRV) is a denotation that combines all measures related to how the heart rate varies, e.g. its standard deviation or the difference between successive HR values. An alternative to ECG is measuring the blood volume pulse (BVP), from which HR also can be derived.

2.2.0.2 Electromyogram (EMG) An EMG records the electrical potential generated by skeletal muscle cells. Needle electrodes are used in this purpose, usually placed on an arm, a leg or a shoulder. Facial electromyography is also possible. In this case the electrodes are placed upon various facial muscles. Several studies have shown the significant changes of EMG features in stressful conditions [Healey & Picard2005] [Wijsman *et al.*2011].

2.2.0.3 Electrodermal Activity (EDA) also Known as Galvanic Skin Response (GSR) or Skin Conductance Response (SCR) The sweat glands and the skin blood vessel are only connected to the sympathetic nervous system, not the parasympathetic. Sweat secretion increases the conductance of the skin proportionally, thus the EDA is measured by the conductivity of the skin. The density of sweat glands is highest around the palms of the hands or the feet, so this is usually where it is measured. Two systems for measuring the GSR are presented in Figure 2.4 including finger electrodes and the Empatica E4 wristband, with wrist electrodes [Ollander2015].

2.2.0.4 Respiration It is possible to measure the respiration (RESP) of a person by recording chest expansion. This can be done using a resistor, by measuring its impedance. The respiration of a person might influence the ECG signal, by causing peaks in the low frequencies (< 0.3 Hz) of the ECG spectrogram [Shi *et al.*2010].

2.2.0.5 Acceleration An accelerometer (ACC) can be used mainly in combination with other sensors to record whether an individual has been moving or not. In this way it is possible to distinguish between physiological reactions caused by movement, and those caused by other means (e.g. psychological stress).

2.3 Multi-Modal Stress Detection

Effectively detecting the stress of a human being provides a helpful way for people to better understand their stress condition and provide physicians with more reliable data for intervention and stress management. Existing findings have indicated that psychological stress can be recognized by the physiological sensors [Deng *et al.*2012]. The physiological information can be acquired by multiple biological or physiological sensors. Figure 2.5 shows the multi-modal stress monitoring system which uses various physiological sensors including respiration, EMG and EDA.

2.3.1 Structure of Personalized Stress Detection System

A stress detection system, such as a personalized bio-medical device, typically consists of three main circuit blocks: 1.Signal Acquisition block including analog to digital converter, 2.Digital Signal Processing blocks which typically contain a feature extractor and ML classifiers 3.Radio transmitter to transmit the processed data or prediction to the user or medical personnel (Figure 2.6).

2.4 Related Prior Work

Several research studies have been undertaken in driver stress detection. Rigas et al. proposed a method based on Bayesian Network for the estimation of vehicle driver stress production due to specific driving events [Rigas *et al.*2008]. Shiwu et al. proposed a system that can actively monitor the driver's fatigue level in real time using the SVM based on physiological features [Shiwu *et al.*2011]. Jeong et al. developed a device which detects the ECG signal of driver in real time as well as degree of stress through analysis of Heart Rate Variability (HRV) [Jeong *et al.*2007]. Healry et al. [Healey & Picard2005] collected and analyzed physiological signals like ECG, GSR, EMG, and RR continuously during real-world driving tasks and determined drivers' relative stress level. Deng [Deng *et al.*2012] proposed a feature selection approach based on Principal Component Analysis (PCA) and evaluated their effectiveness in terms of correct rate and computational time using five classification algorithms including linear discriminant analysis (LDA), C5.4 induction tree, SVM, Naïve Bayes and KNN. Shi [Shi *et al.*2010] built a personalized stress detection based on SVM and evaluated it on the collected data and results showed that their model can detect stress with high precision.

Zhai [Zhai *et al.*2005] focused on the use of physiological signals including blood volume pulse (BVP), GSR and pupil diameter to automatically monitor the stress state of computer users. Several other studies used multi-physiological signals to detect the stress [Sun *et al.*2010] [Wagner, Kim, & Andre2005] [Salai, Vassányi, & Kósa2016].

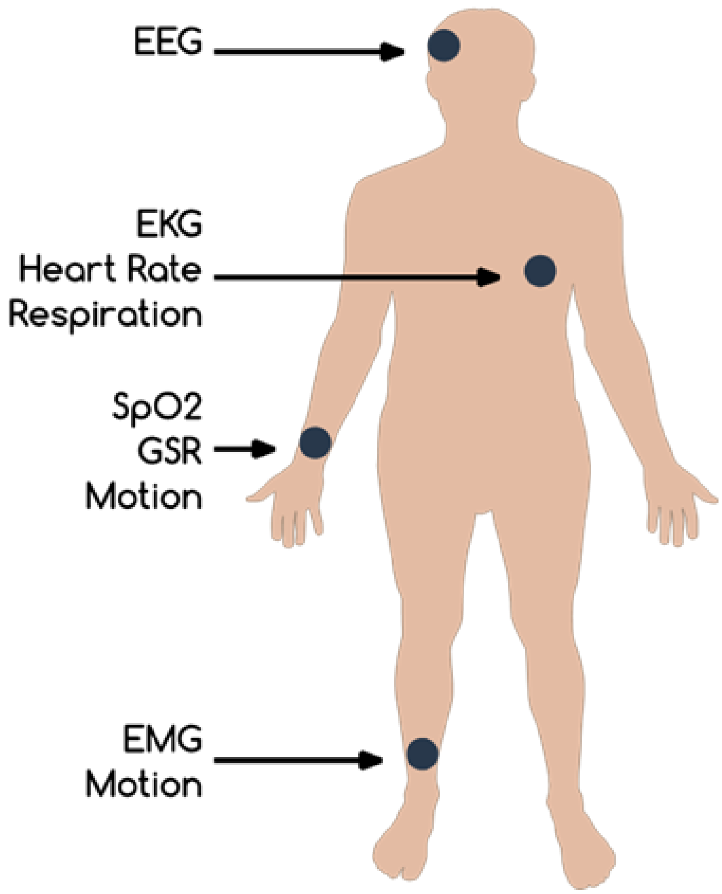


FIG. 2.1: Common physiological signals that might be used for stress detection. EKG = electrocardiogram, EEG = electroencephalogram, EMG = electromyogram , GSR = Galvanic Skin Response.

FIG. 2.2: ECG electrodes (figure obtained from [Ollander2015])

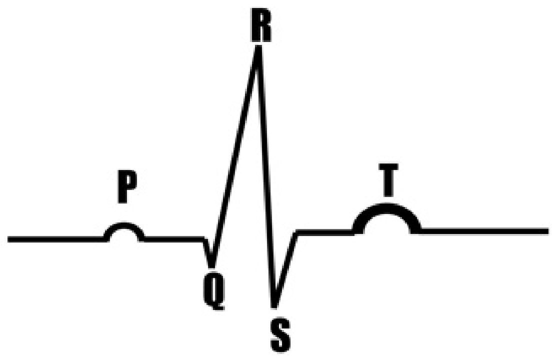


FIG. 2.3: An ECG signal representing a heartbeat, with the usual elements: P wave, QRST complex and T wave.

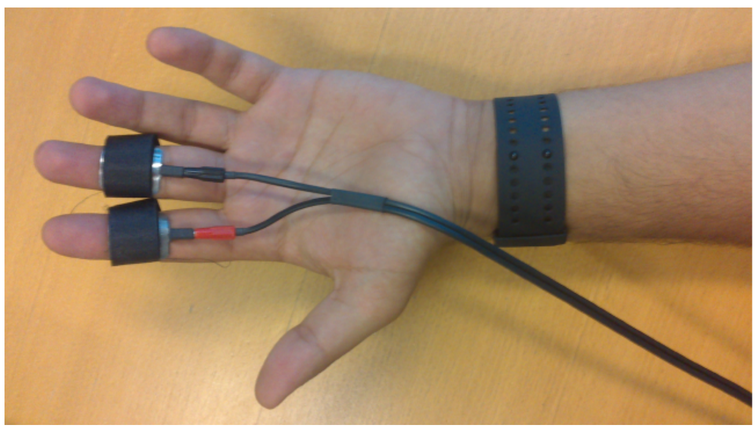


FIG. 2.4: GSR electrodes and Empatica E4 wristband [Ollander2015]

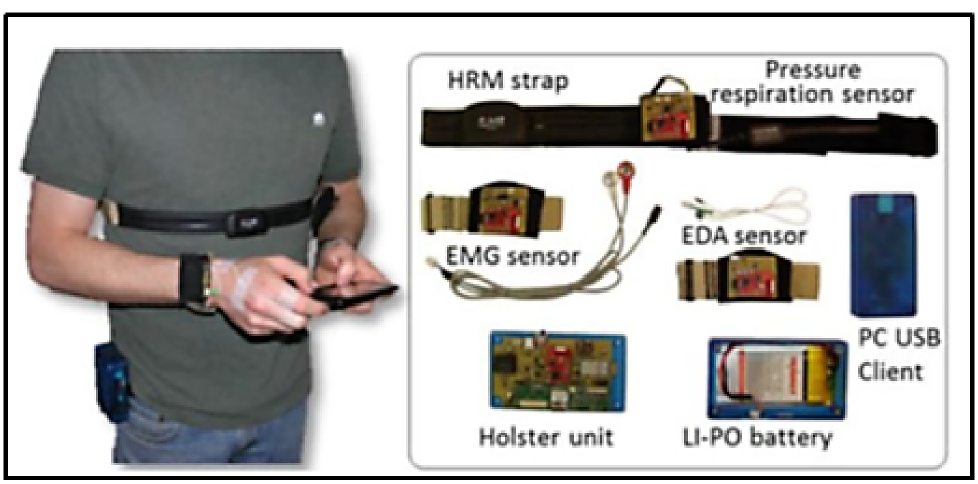


FIG. 2.5: Monitoring the stress level using various physiological sensors

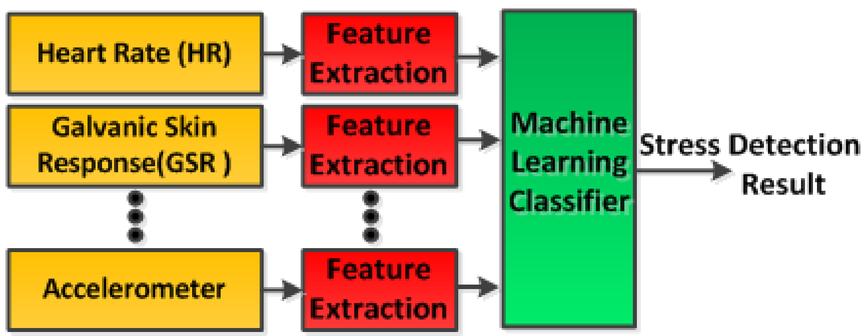


FIG. 2.6: Block diagram of a multi-physiological stress detection system containing data acquisition by sensor, feature extraction, and machine learning classifier to generate result. The application also shows inherent data-level and task-level parallelism.

Chapter 3

MULTI-MODAL STRESS DETECTION SYSTEM

3.1 ARL Dataset

We used the dataset that was collected by Army Research Laboratory (ARL) in this project. This dataset consists of multi-physiological and behavioral recordings from 15 individuals.

Participants performed a shooting task in a simulator in which they had to discriminate enemy versus friendly targets and decide to shoot or refrain respectively. Each participant, one by one, participated in several pre-programmed scenarios. Each scenario lasted approximately 10-22 minutes and provided 300 degree of visibility in the simulator.

Target pairs were presented in various locations within each simulated scene (e.g. behind a car, wall, building, natural terrain, rocks). The foe targets pointed and fired a M-9 pistol at the participant. The friend targets performed actions like handing over a soda, pulling out a wallet, or showing the “I surrender” hands. The target pairs were either two friends or one enemy and one friend (two enemy targets were not presented). The interval between display target pairs varied between two, four and six seconds. This inter-trial interval was used to minimize a pattern effect. There were a total of 128 target pairs that could be presented. There were 64 friend target pairs and 64 foe target pairs.

Three types of feedback were induced to participants during the experiments based on their success or failure in performing the shooting task. These feedback types included shock (tactile feedback on belt), lifebar (visual feedback display, turning a green light to red on errors) and nofeedback conditions. In shock and lifebar conditions, a shock or change in the life bar occurred when a foe was not hit and a minimum of 30 seconds had passed since the last shock or change in life bar respectively [Patton2013]. Target pairs were presented on the same screen for two seconds.

During the experiment, participants wore a physiological monitor (Equivital EQ02^{TM}), which captured 3 axis accelerometry data, ECG, chest expansion (Respiration), peripheral capillary oxygen saturation (SpO2), and EDA [Attaran, Brooks, & Mohsenin2016]. Figure 3.1 shows the 300-degree simulator and the embedded sensors that were used to collect physiological signals.

3.2 Labeling the Data based on the Stress Measurements

The ARL team used two different approaches for stress measurement: salivary amylase and the Multiple Affect Adjective Check-list Revised (MAACL-R) (analyzing the questioners' responses).

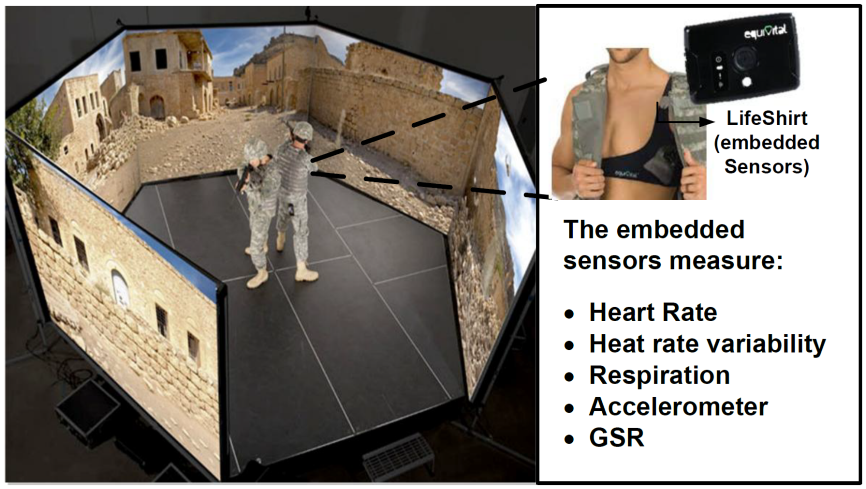


FIG. 3.1: 300-degree simulator to collect the multi-physiological data during different levels of stress using the embedded sensors in wearable life-shirt (Figure obtained from [Patton2013]).

3.2.1 Salivary Amylase

The salivary amylase test was performed to derive a quantifiable level of stress. Saliva samples were collected during the shooting task for all participants in different conditions. The saliva samples were used to calculate physiological stress levels. For each sample collected from the specified times, the same saliva sample was used for amylase assays. Figure 3.2 illustrates the amylase level of each condition.

3.2.2 Multiple Affect Adjective Check-list Revised (MAACL-R)

The Multiple Affect Adjective Checklist-Revised [Lubin & Zuckerman1999] consists of a list of 132 adjectives in which participants are instructed to check all the words describing how they “feel right now” or “during the simulation”. The MAACL has six validated subscales, including anxiety, depression, hostility, sensation seeking, positive affect, and

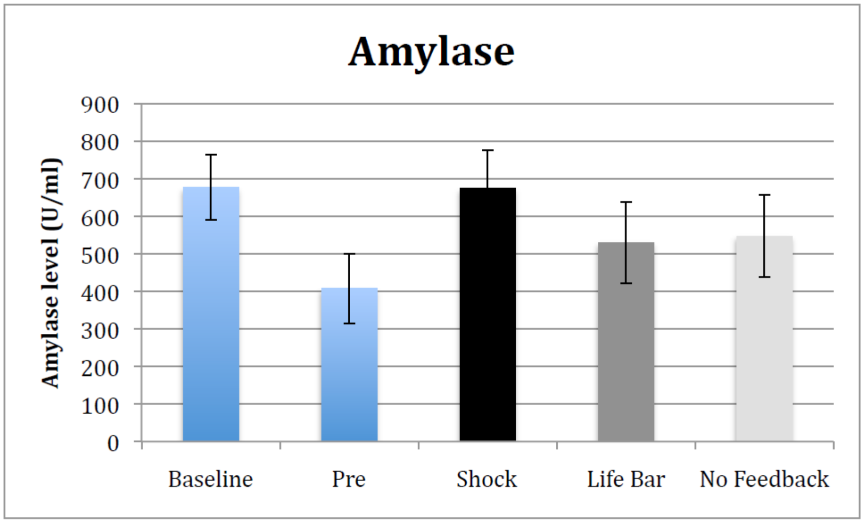


FIG. 3.2: Mean+SEM (standard error of mean) salivary amylase. Baseline indicates the initial condition. Pre represents the pre-shock condition (figure obtained from [Patton2013]).

dysphoria. A univariate ANOVA was used to identify specific measures with significant differences among conditions. Finally, paired t-tests were used to identify the differences between conditions [Patton2013]. The anxiety level in the five different conditions is shown in Figure 3.3.

3.3 Preprocessing of the Data

First, we needed to narrow down the raw data. In this research, we are interested in digging into different physiological recordings of 15 participants. As was mentioned in the previous section, various physiological signals were recorded during the shooting experiment. The sampling frequencies of collected signals are very diverse. The ECG channel has the highest sampling frequency (200 HZ) among all signals in the dataset and is used as a base timeline. There are many NaN values for other sensor data to fill out the

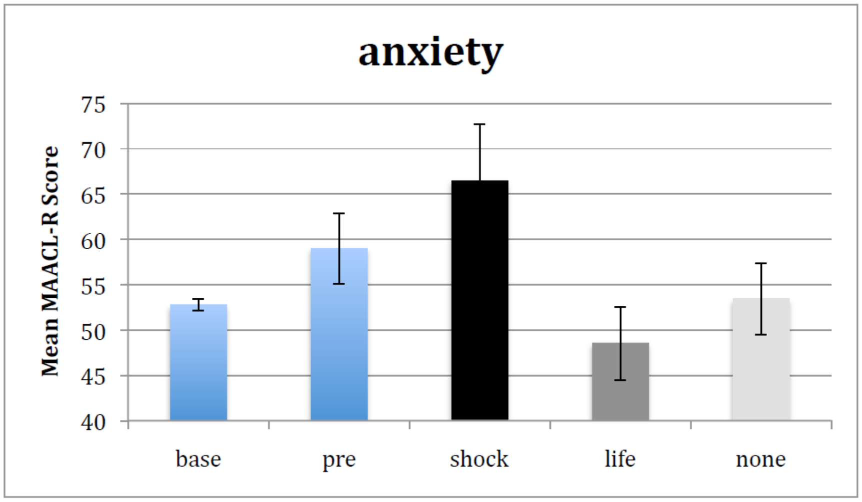


FIG. 3.3: Mean+SEM MAACL-R Anxiety subscale by condition including baseline, pre-shock, shock, lifebar and no-feedback. The Mean MAACL-R Score for anxiety parameter is highest in shock condition. (figure obtained from [Patton2013])

time periods that have no data. In this study, we focused on three periods of the shooting experiment: shock, lifebar and nofeedback. Thus we needed to remove the unnecessary portions from the recordings.

The other preprocessing steps is to visualize various physiological signals which are affected mostly by stress. After evaluating various available sensor data in the ARL dataset, we chose HR, RESP, ACC, and SpO_2 physiological signals for stress monitoring.

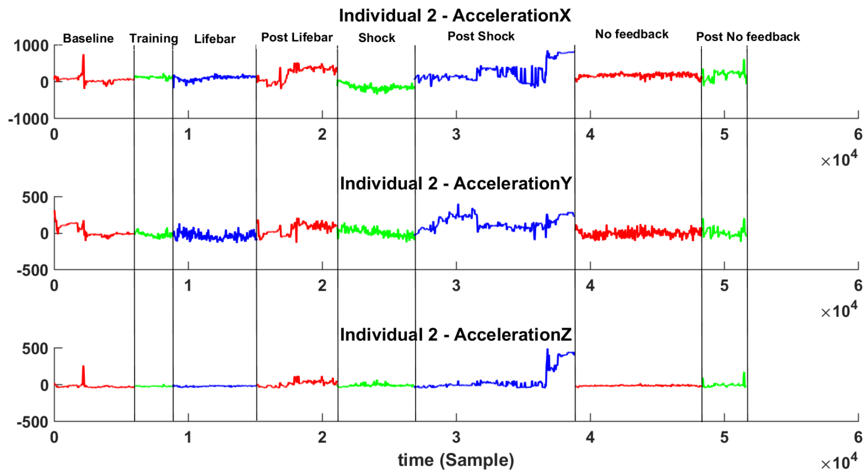
3.4 Feature Extraction

When working with large data quantities (e.g., physiological signals over a large time duration), it is essential to utilize feature extraction and feature selection approaches. Using raw physiological data as inputs for classifiers or clustering models results in low accuracy. Thus, we need to extract an effective feature set from raw data. Feature extraction means reducing the raw data into more comprehensive measures. Computing features of signals by statistical methods is one way of feature extraction. Some features are signal-specific, e.g. rise time of the GSR after a stressful event, while others are more general, e.g. the mean value of a signal during a time window. To choose a feature set, we can search the raw data for patterns in the signals, especially between different classes [Ollander2015]. Feature extraction is a critical step in most data analysis applications. Working with recognizable properties in the signals rather than raw data makes the models easier to understandable, while they are more likely to be generalizable.

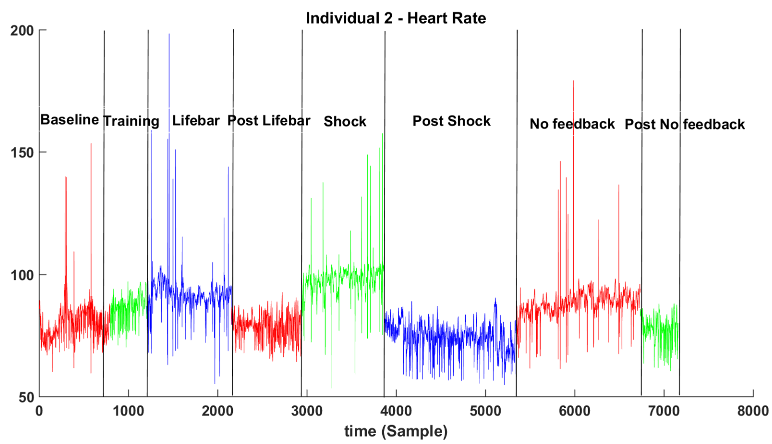
Fig 3.4 shows the raw accelerometer, heart rate and oxygen saturation signals in all different situations during the shooting experiment for the individual 2 in the given dataset. Table 3.1 shows the physiological signals and their extracted features which are most correlated to the stress level in different previous studies.

3.4.1 Data Segmentation

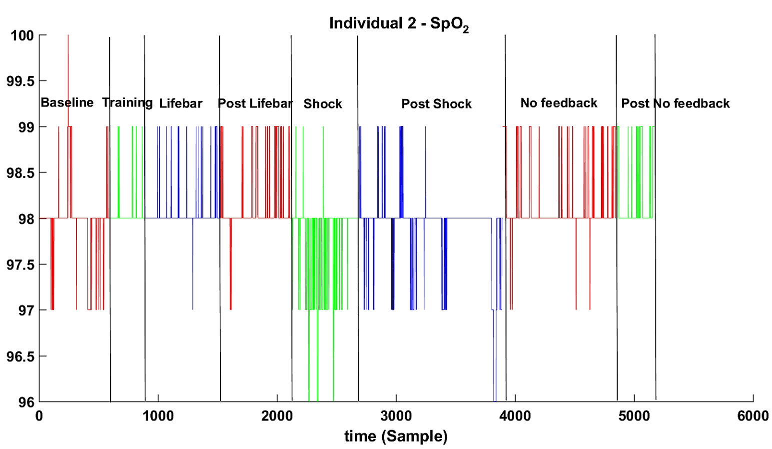
During the shooting task, each participant took part in several trials in which a target pair appeared in the simulated shooting task. We divided the physiological recordings based on the average length of trials. The trial duration (and the corresponding windows) is around 6 seconds. We extracted several time-domain and frequency domain features from different physiological signals for all 6 second windows of data in each of the three feedback conditions. Each of these windows were assumed to represent a period of low, high or medium stress depending on the feedback modality provided during that time period. For



(a)



(b)



(c)

FIG. 3.4: Raw signal representation: (a) Accelerometer (3-axes), (b) HR and (c) SpO_2

Source	Signals	#Feat.	Important feat.
[Holzinger, Bruschi, & Eder2013]	BVP, EDA	12	GSR: mean, sum
[Frank *et al.*2013]	ACC, ECG, HR, RESP, ST	16	
[Shi *et al.*2010]	ECG, GSR, RESP, ST	26	
[Sharma & Gedeon2012]	BP, BVP, EMG, GSR, HR, ST, RESP	13	ECG features, GSR features, HRV
[Setz *et al.*2010]	ECG, EDA, RESP	16	EDA: mean peak height, slope
[Riera *et al.*2012]	EEG, EMG, face	5	EEG: alpha asymmetry, alpha/beta ratio
[McDuff, Gontarek, & Picard2014]	BVP, EDA, PPC, RESP	5	HR: mean, mean RESP rate, HRV: LF power, HF power, LF/HF

Table 3.1: Features of physiological signals commonly used for stress detection. BP= Blood pressure, face: face measurement, HRV= heart rate variability [Ollander2015]

each participant, we segment each channel of data into a 6-second window to obtain the data windows w_1, w_2, ..., w_n. We denote F_i as the feature vector extracted from the data window w_i. We created a set of feature vectors F for each participant.

3.4.2 Feature Set

For monitoring the stress level during the shooting experiment we extracted 16 features from ECG, respiration rate, oxygen saturation and accelerometer physiological signals. We will explain the extracted features from these four physiological signals with more details as follows.

HRV Analysis: Heart rate (HR) was determined as the duration between peaks of the QRS complex in ECG. RR indicates the interval between two consecutive QRS complex in the ECG signals. Both HR and RR were recorded for all participants during the shooting experiment. HRV analysis can be categorized into time domain and frequency (spectral) domain analysis. Time domain analysis is calculated directly from RR-intervals and HR over the 6 second window. In this research, we found the mean value of the HR (mean HR), standard deviation of HR (Std HR), mean value of RR-interval (mean RR) and Standard deviation of the RR-interval (Std RR).

Moreover, in the spectral domain method, a power spectrum density (PSD) estimate is calculated for the RR interval series. We applied a Fast Fourier Transform (FFT) to convert the time-domain RR-interval sequence to the power spectrum. The low frequency band (0-0.08 Hz) represents sympathetic nervous system activity. The high frequency band (0.15-0.5 Hz) which is modulated by the parasympathetic system activity. The LF to HF ratio (LF/HF) is used as an index of automatic balance (increase in stress level will increase this ratio) [Healey & Picard2005] [Sun *et al.*2010]. Thus we extracted three frequency domain features from RR-interval including LF, HF and LF/HF ratio.

Accelerometer Analysis: Six accelerometer features (mean and standard deviation) were derived from each of the 3-axis accelerometer signals. The magnitude feature is calculated based on (3.1) as well.

$$magnitude = \frac{1}{n}\sum_{i=1}^{n} \sqrt{x_i^2 + y_i^2 + z_i^2} \tag{3.1}$$

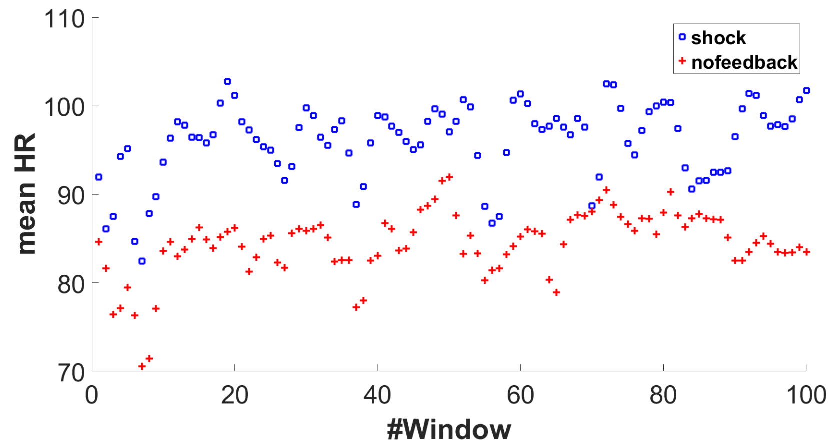
Feature No.	Sensors	Features
1 to 7	ECG	Mean HR, Std HR, Mean RR, Std RR LF-HRV, HF-HRV, LF/HF ratio
8 to 14	Accelerometer	Mean of X,Y and Z axis Standard deviation of X,Y and Z axis Magnitude of three axes
15	Resp. Rate	Mean RR
16	SpO_2	Mean SpO_2

Table 3.2: 16 extracted features from four physiological sensors per each 6 second window.

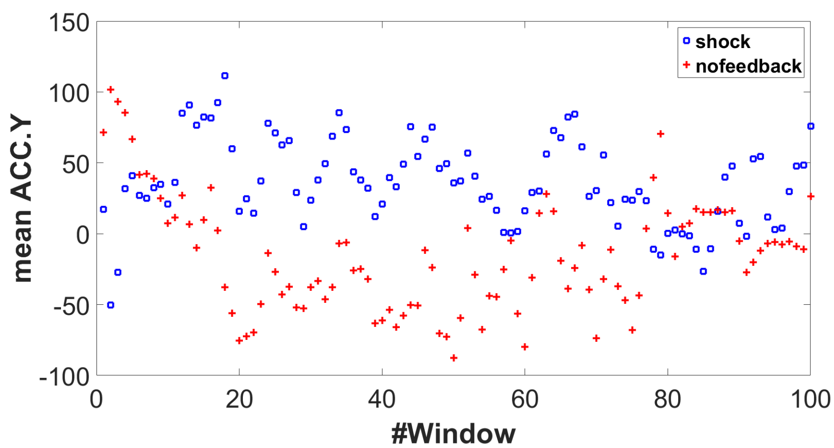
Respiration and oxygen saturation analysis: The mean value of respiration rate (mean RR) and the mean value of SpO_2 per each window were also derived as two other time domain features. The List of extracted features from the four physiological channels are elaborated in Table 3.2. Figure 3.5 indicates the representation of three different features including mean HR, mean ACC.Y and SpO_2 for participant 2 in the shock and nofeedback conditions.

3.4.3 Feature Normalization

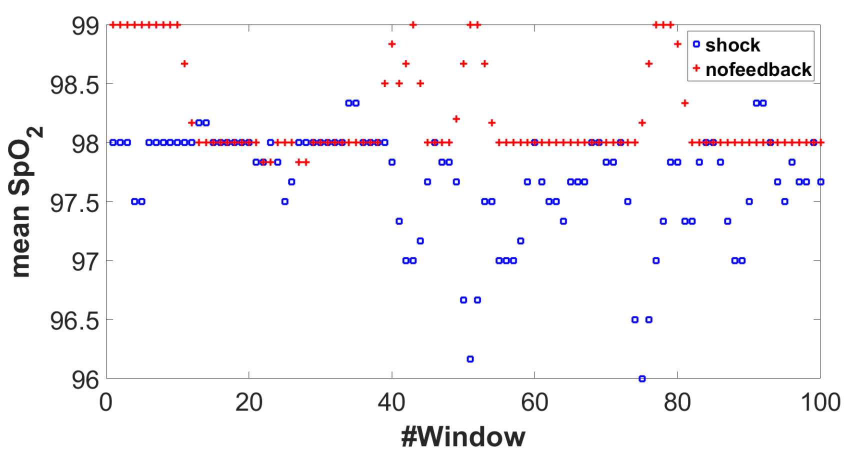
Physiological signals are dependent on each individual's initial physiological level. Even if we measure physiological signals such as HR and GSR baseline levels for a single individual, these signals may still vary due to mental state and the sensor's connectivity. To eliminate these factors, we normalized each feature. There are different methods for features normalization. In this study, we utilized the min-max scaling to normalize the



(a)



(b)



(c)

FIG. 3.5: Representation of three features for participant 2 in shock and nofeedback conditions : (a) mean HR, (b) mean ACC.Y and (c) SpO_2

feature set (3.2).

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{3.2}$$

3.5 Machine Learning Classification and Feature Selection

One of the most important parts of the stress monitoring systems the same as other health monitoring systems is classification. The performance of classification algorithm has a significant role on the effectiveness of the entire system. In this research, we utilized the SVM and KNN machine learning algorithms as two well-known classifiers to find the best feature set for stress monitoring.

First, we reviewed these two binary classifiers, then we shown how to choose the best features from 16 extracted features.

3.5.1 KNN Machine Learning Classifier

The KNN algorithm uses the K nearest samples to "vote" for the class membership of a new sample (Figure 3.6). K is usually chosen as a small number, and different weighting of each neighbor is sometimes used. A small K is more sensitive to noise, but a large K makes the algorithm computationally expensive. For binary classification, odd K is a good idea since this prevents ties in the voting process. Note that the KNN usually works in the feature space. If new data points are introduced, relearning the KNN is simple, since it simply means rechecking the neighbors. The most commonly used distance to decide which class is nearest is the Euclidean distance (3.3) ([Ollander2015]).

$$d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{3.3}$$

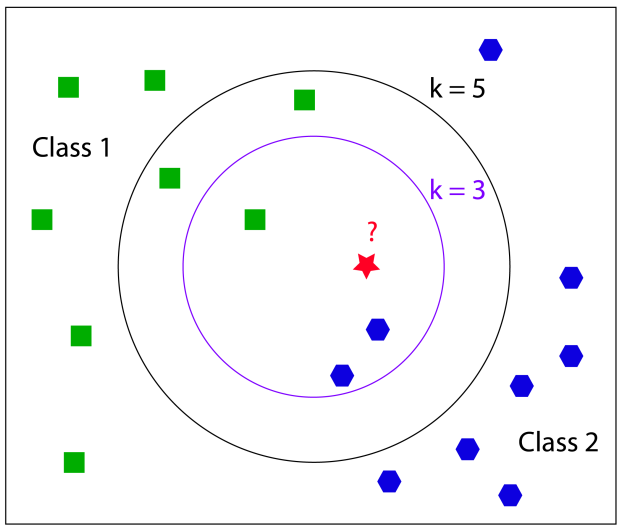


FIG. 3.6: KNN classifier with the two classes: blue (hexagonal) and green (square) symbols. The class of the new point (marked by "?") is shown by voting using its K = 3 and K=5 nearest neighbors.

3.5.2 SVM Machine Learning Classifier

Support vector machines work by finding the maximum margin hyperplane, i.e., the linear separator that is as far as possible from the closest positive and negative training instances (3.7). Kernel functions can be used to project the data into a high dimensional space such that the linear separator is highly non-linear in the input space. The default linear and polynomial kernels are in (3.4) and (3.5), respectively [Page *et al.*2015].

$$K(S_i, x) = S_i \cdot x \tag{3.4}$$

$$K(S_i, x) = (\gamma \times (S_i \cdot x) + b)^d \tag{3.5}$$

Samples are then classified using the function shown in (3.6), where S_i is a support vector, x is a test vector, $K(s_i, x)$ is the kernel function, α_i and γ_i are the weight and label of the support vector, and b is the bias.

$$f(x) = sign(\sum_{i=1}^{NUM\_SV} \alpha_i \gamma_i K(S_i, x) + b) \tag{3.6}$$

3.5.3 Cross Validation

Cross validation is a model evaluation method. In this technique, some of the data is removed before training begins. When training is done, the data that was removed can be used to test the performance of the learned model. This is the basic idea for the whole class

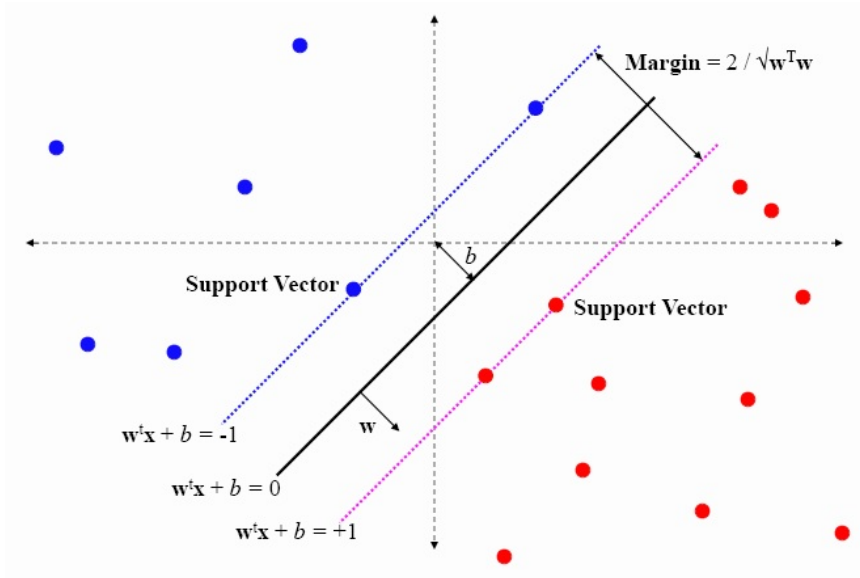


FIG. 3.7: The representation of the SVM classifier with the two classes: blue and red symbols.

of model evaluation methods called cross validation.

3.5.3.1 k-fold Cross Validation In k-fold cross-validation, the original sample is randomly partitioned into k equal sized subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k-1 subsamples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds can then be averaged to produce a single estimation. In this study, we used this validation technique to evaluate the accuracy of the general model (when we use all participants' data).

3.5.3.2 Leave-one-out Cross Validation Leave-one-out cross validation (LOOCV) is k-fold cross validation taken to its logical extreme, with k equal to N, the number of data points on the set. That means that N separate times, the function approximator is trained on all the data except for one point and a prediction is made for that point.

In this research, due to limited samples for each individual, we use this technique to evaluate the accuracy of personalized stress detection model.

3.5.4 Features Selection based on Accuracy

When the features are extracted, one needs to examine which ones contain the most useful information, and remove those that are not contributing to improving the model.

Feature selection means choosing a subset among the extracted features that provides a good prediction performance and a small generalization error. The generalization of a machine learning measures its capacity to predict unseen data. A high generalization error means that the model does not perform well on new data.

Too many features might lead to overfitting (overly complex models) while including too few features means a risk of losing useful information (underfitting). One must also keep in mind that some features can perform poorly alone, but can prove very useful in combination. Thus, one must be careful while analyzing features one by one. In this research, as we mentioned previously, we defined two classes.

- Class 1: “less stress (nofeedback mode)” **label 0**
- Class 2: “high stress (shock mode)” **label 1**

The automated method to choose the most appropriate features is classification accuracy. We find the best feature set by adding or removing features based upon if they are making prediction easier or harder (by evaluation of the classification accuracy).

In order to find the best combination of features, we examined the classification accuracy of each feature for all individuals independently. We utilized two popular binary classifiers, SVM and KNN, to analyze the accuracy of each feature for all participants.

We developed custom MATLAB scripts to train the classifiers and determine the accuracy of the classification for both KNN and linear SVM algorithms. In the KNN classifier the k parameter was set to 3 and Euclidean distance measurement was used to find the nearest neighbors. We also utilized the linear kernel for SVM model implementation. For both models, we used The LOOCV technique to measure the accuracy for each participant.

Figure 3.8 shows the average accuracy for all 15 individuals for each feature when using the SVM and KNN classifiers. The mean heart rate, mean RR-interval, mean accelerometer data in all 3 axes, accelerometer magnitude, and mean SpO_2 achieve the highest accuracy across all individuals.

3.5.4.1 Forward and Backward Feature Selection Algorithms Forward and backward feature selection algorithms are the two main approaches to choose the most important features from a given feature set. As it was shown it the previous section, after evaluating the single-user accuracy results for all features using both SVM and KNN, we selected seven features. Since our final goal is to implement one low power and low area processor for stress detection, we are interested in applying some feature selection method

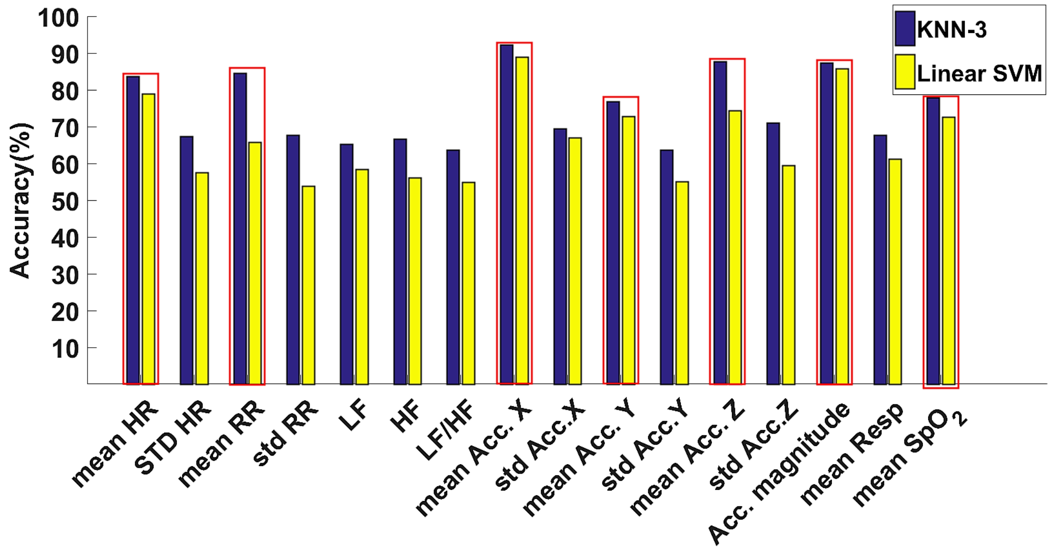


FIG. 3.8: Average of two-level stress detection accuracy using multi-physiological sensors and corresponding features for 15 individuals. The features with highest accuracy (more than 65% for both SVM and KNN) are highlighted.

on this feature set to find the most important features.

The forward algorithm starts with an empty feature space, successively adding the feature that increases the classifier performance the most. This is done until all the features have been added and afterward one can observe the performance to decide which combination of features was most efficient. The backward algorithms work identically, except that they start with all the features, successively deleting the features that decrease the classifier performance the least.

We utilized the "Attribute Selection" function from Weka tool and chose the backward feature selection to find the best feature set for both SVM and KNN classifier. The Weka's returned 4 features from HR and accelerometer including mean HR, mean ACC.X, mean ACC.Y and mean ACC.Z as the most effective features to detect two level of stress.

Classifier	mean HR	mean ACC.X	mean ACC.Y	mean ACC.Z	Combined feature set
SVM	78.99%	88.79%	72.83%	74.38%	96.7%
KNN	83.06%	92.13%	76.66%	87.56%	95.81%

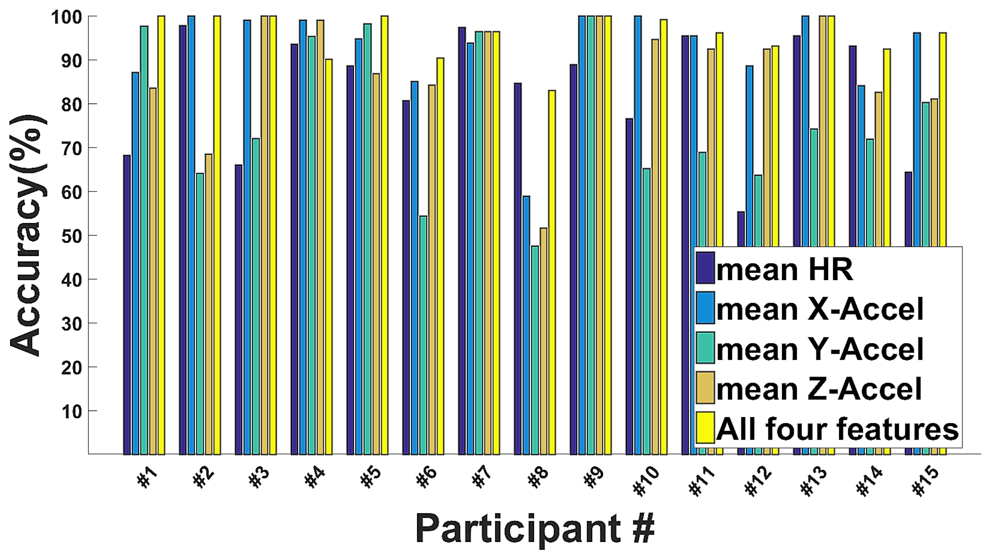
Table 3.3: The average accuracy of the four most important features plus combined feature set across all 15 participants.

Figure 3.9 shows the average accuracy for individual features including mean HR, mean ACC.X, mean ACC.Y and mean ACC.Y, as well as the combined feature set when using the KNN and SVM classifiers across all 15 participants. The average accuracy of KNN and SVM classifiers for the combined features set are 95.81% and 96.7% respectively. Table 3.3 shows the average accuracy of each separate feature and the combined feature set.

3.6 Hardware Complexity Analysis of SVM and KNN

We examined the supervised machine learning algorithms including linear SVM and KNN for stress detection case study. In this section, we perform complexity analysis of these supervised machine learning classifiers. For the KNN algorithm, the training step cannot be performed offline. With the SVM, the training step can be performed offline to find the required support vectors, which are much smaller in size compared to training data in the KNN data. Table 3.4 compares the memory and arithmetic operation requirement by KNN and SVM. The memory requirement and arithmetic operations are larger for the KNN classifier. Given the increased computational and memory requirement which leads to larger footprint and increased power requirements for KNN implementation.

(a)



(b)

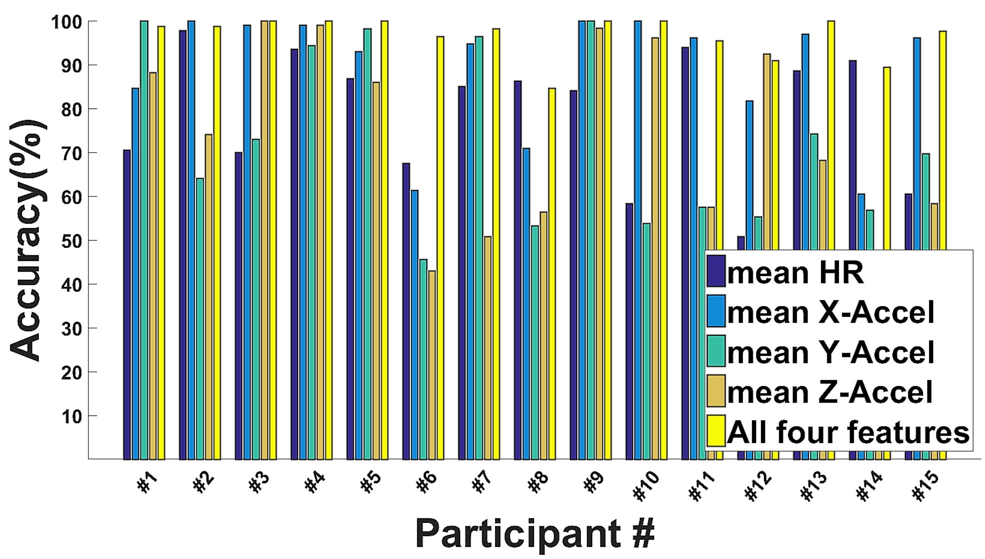


FIG. 3.9: The comparison of accuracy level for multi-modal feature set with separate features from HR and accelerometer for two classifiers: (a) KNN-3 (b) Linear SVM

Table 3.4: Hardware complexity analysis of KNN and SVM, n: size of training data, m: size of test data, p: No. of features, and s: No. of support vectors.

Algorithm	Multiplications	Additions	Memory Requirement
KNN	$p \times n \times m$	$(p-1) \times n \times m$	$n \times p$
Linear SVM	$p \times m \times s$	$m \times (p-1) \times s$	$p \times s$

Chapter 4

PROCESSOR ARCHITECTURE EXPLORATION FOR WEARABLE LOW-POWER STRESS DETECTION

In this section, we explore design and implementation of a low-power and low-area processor for wearable stress detection using several hardware-based and software-based platforms. In the personalized stress monitoring system, we use each participant's training data so that the size of memories to store the training data in KNN or support vectors in SVM are smaller. In contrast with the personalized model, in the generalized model for stress monitoring we use all participants' data to train the classifiers, so we need large memories. Artix FPGA, PENC manycore, and NVIDIA TK1 platforms provided the enough memory to design and implement generalized stress monitoring system.

For a hardware based solution, we will present the post-layout ASIC designs for both SVM and KNN processors for personalized stress detection in the 65 nm CMOS technology. We utilized the Artix 7 FPGA to implement these two processors as another choice for a hardware-based platform.

We will explore the choice of software-based processors for energy-efficient processing of physiological signals for a wearable multi-modal stress detection system as well. These platforms provide task-level parallelism for biomedical application with large amount of processing. To meet this goal, the entire stress detection system which consists of feature extraction and classifier is implemented on programmable accelerators including NVIDIA TK1 ARM A15 CPU and Kepler GPU, and PENC domain-specific many core.

4.1 Hardware-based Solutions

4.1.1 Proposing Scalable and Pipeline SVM Processor

Figure 4.1 shows a detailed architecture of the linear SVM processor used for a stress monitoring system based on four extracted features from the heart rate and accelerometer sensors. The proposed parallel pipelined architecture is flexible for variable number of features. The support vectors (S), bias (b) and other required coefficients were calculated offline using the SVMtrain MATLAB function. Each memory block is loaded with precomputed weighted support vectors from a trained model for each feature. There are sufficient registers in this design to store the intermediate results in the pipeline scheme. The classifier receives the features derived in 6 second windows as testing input. The dot product operation runs between the testing data and all supporting vectors available in RAM memory blocks. This is followed by the parallel dot product operation, which is added with bias parameter at the final stage to find the predication result.

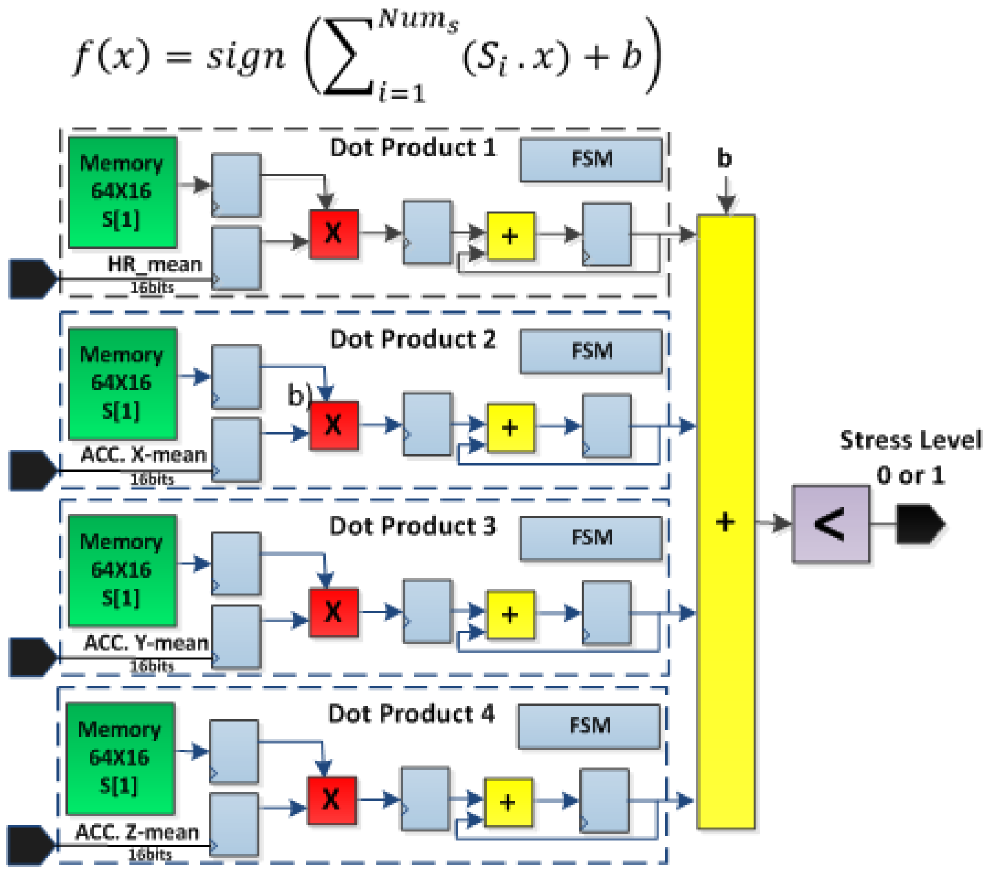
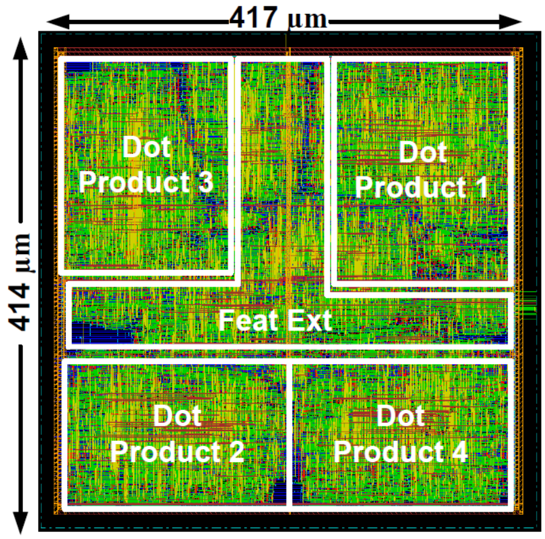


FIG. 4.1: Block diagram of the hardware implementation of SVM classifier for personalized stress detection system. Block RAMs on the left (1 Kbit each) store 64 support vectors.

4.1.2 ASIC Implementation Results for SVM Processor

In order to lower the overall power consumption, we use a standard-cell RTL to GDSII flow, using synthesis and automatic place and route. The hardware was developed using Verilog to describe the architecture, synthesized with RC compiler, and placed and routed using Cadence SOC Encounter in the 65 nm TSMC CMOS technology.

Figure 4.2 shows the post-layout view and results for the proposed SVM processor for a stress monitoring system. The SVM processor runs at 250 MHz and consumes the



SVM Processor + Feature Extraction Implementation Results

Technology	65 nm, 1v
Logic Utilization	91%
Area	0.17 mm^2
Frequency	250 MHz
Total Power	39.4 mW
Total Exe. Time	340 ns/170 cycles
Energy	13.4 nJ

FIG. 4.2: Layout view and post-layout implementation results of the proposed multi-modal SVM processor (64 support vectors) + feature extraction. The highlighted regions indicate the location of four dot product components and feature extraction on the chip.

power of 39.4 mW. The prediction for each window of data (6 sec) takes 0.3 us resulting in 13.4 nJ energy consumption per window.

4.1.3 Proposing Configurable KNN Processor

Figure 4.3 shows a high-level block diagram of the KNN-3 processor used for the stress monitoring system based on four extracted features from the heart rate and accelerometer sensors.

In this architecture we store the training samples (256 training samples for personalized stress monitoring) and their corresponding labels in the ROM block. The extracted feature set (4 features) from testing samples from the 6-second window was stored in the buffer component. The four subtractors, four multipliers and adder modules are used to find the Euclidean distance between the given test sample and training data.

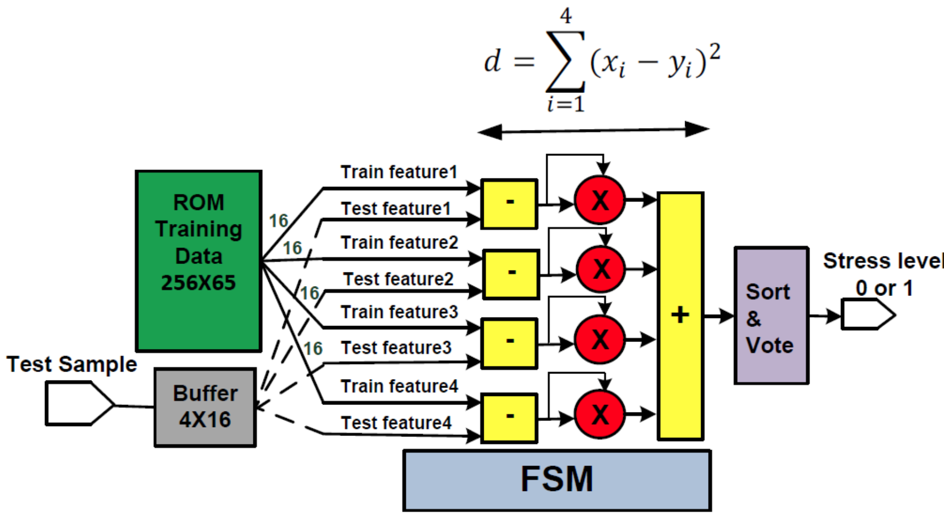


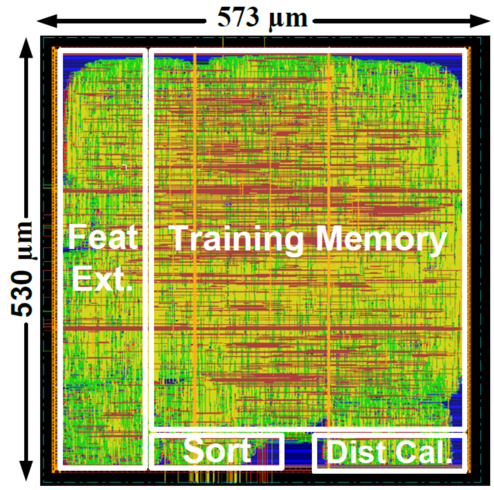
FIG. 4.3: Block diagram of the hardware implementation of configurable KNN processor. x represents the testing sample and y represents a training sample.

The sorting block is responsible for finding the K smallest distance between the testing sample with all training data. The vote module generates the label of testing sample based on the majority voting at the final step.

The Finite state machine (FSM) module is responsible for syncing and controlling all modules. This design is fully configurable for variable number of features and variable size of training.

4.1.4 ASIC Implementation Results for KNN Processor

The post place and route results of KNN implementation is shown in Fig 4.4. The KNN processor runs at 250 MHz and consumes 76.69 mW. The prediction for each window of data (6 sec) takes 4.1 us resulting in 0.31 uJ energy consumption per window.



KNN Processor + Feature Extraction Implementation Results	
Technology	65 nm, 1v
Logic Utilization	94%
Area	0.3 mm^2
Frequency	250 MHz
Total Power	76.69 mW
Total Exe. Time	4.1 µs / 1025 cycle
Energy	0.31 µJ

FIG. 4.4: Layout view and post-layout implementation results of the proposed multi-modal KNN processor (256 training data) + feature extraction. The highlighted regions indicate the location of training memory, sorting, distance calculation and feature extraction on the chip.

4.1.5 Xilinx Artix-7 FPGA

FPGAs are highly flexible, allowing on-the-fly configuration to optimize bit resolution, clock frequency, and parallelization for a given application. In addition, modern FPGAs provide accelerators to boost the performance for operations such as multipliers, generic DSP cores, and embedded memories.

The main disadvantages of FPGAs, however, are that they have substantially higher leakage power and require writing low level logic blocks in HDL. For stress detection case study, complete FPGA hardware solution which consists of machine learning classifier and feature extraction was developed in Verilog that utilized highly parallel, highly pipelined DSP and ML kernels. Both real-time and simulated projections using commercial tools were used to perform timing and power analysis when running test stimulus. For stress detection, the smallest Artix-100T, is targeted on the Nexys platform. Table 4.1 summarizes

Design	SVM	KNN
FPGA package	XC7A100T	XC7A100T
Registers (#)	340	446
LUTs (#)	199	705
Memory (Kb)	512	1040
Operating Freq. (MHz)	166	166
Latency (cycles)	6084	64084
Dynamic Power (mW)	26	149
Leakage Power (mW)	82	83
Total Energy (uJ)	3.9	89.58

Table 4.1: Artix-7 FPGA performance for Stress detection case study consisting of features extraction (4 features) for KNN-3 and linear SVM classifier. Both Dynamic power and total results are presented for FPGA core only.

the results of implementation of stress detection for both SVM and KNN kernels on the Artix FPGA.

4.2 Software Solutions for Stress Detection

In order to study how embedded off-the-shelf processors can efficiently process biomedical applications, we conducted the experiments on NVIDIA ARM A15 CPU and GPU to implement the entire stress detection system.

To address the need for programmability, low power consumption, area efficiency and parallel computing platform, we adopted the PENC many-core platform. In order to determine the best platform for stress detection, this case study with both SVM and KNN classifiers is implemented on various software-based platforms.

4.2.1 Power Efficient Nano Cluster (PENC) Manycore

PENC manycore accelerator is a homogeneous architecture that consists of in-order tiny processors with a 6-stage pipeline, a RISC-like DSP instruction set and a Harvard Architecture model [Bisasky, Chandler, & Mohsenin2012], [Bisasky *et al.*2013], [Tavana *et al.*2014], [Kulkarni *et al.*2016c], [Kulkarni *et al.*2016b] which is designed and implemented in the EEHPC laboratory [EEH]. The core operates on a 16-bit data-path with minimal instruction and data memory suitable for task-level and data-level parallelism. Furthermore, these cores have a low complexity, minimal instruction set to further reduce area and power footprint. The lightweight cores also help to ensure that all used cores are fully utilized. The processor can support up to 128 instructions, 128 data memory, and provides 16 quick-access registers [Page *et al.*2016].

In the network topology, a cluster consists of three cores that can perform intra-cluster communication directly via a bus and inter-cluster communication through a hierarchical routing architecture. Each cluster also contains a shared memory. Figure 4.5 shows the block diagram of a 16 cluster version of the design, highlighting the processing cores in a bus-based cluster. Each core, bus, shared memory and router was synthesized and fully placed and routed in a 65 nm CMOS technology using Cadence SoC Encounter and results for one cluster are summarized in Figure 4.5.E. The processing core contains additional buffering on the input in the form of a 32-element content-addressable memory (CAM). It is used to store packets from the bus and allow a finite state machine (FSM) to find a

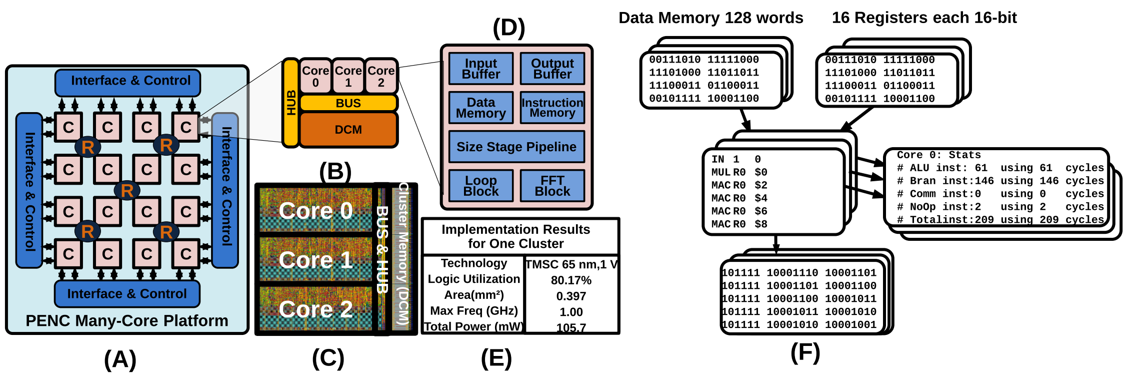


FIG. 4.5: (A) Power Efficient Nano Clusters (PENC) Manycore Architecture (B) Bus-based Cluster Architecture (C) Post-layout view of bus-based cluster implemented in 65 nm, 1V TSMC CMOS technology (D) Block Diagram of core architecture (E) Post Layout implementation results of optimized bus-based cluster (consisting of 3 cores + bus + cluster Memory) (F) PENC simulator and compiler flow high-level diagram.

word where the source core field corresponds to that in the IN instruction itself, where the IN instruction is used to communicate between the cores. For example, if the core is executing `IN 3`, the FSM searches through the CAM to find the first word whose source core is equal to three. This word is then presented to the processing core and processing continues. PENC manycore architecture has 3 light-weight processing cores and a shared memory in a single cluster. Our initial manycore architecture design had 4 processing cores without shared memory, which was ideal for DSP kernels for minimal data storage. Since bio-medical applications use ML kernels which often require large amount of memory for their model data, the proposed PENC manycore architecture replaces the fourth core with a shared memory to accommodate the memory requirements.

4.2.1.1 Bus based Cluster Cores use the IN and OUT instructions to communicate with each other. When a core executes an OUT instruction, the data and relevant addressing information is packetized and sent to its output FIFO through a bus. When data is present in a core's output FIFO, it requests to use the cluster bus. The bus then arbitrates between requests, only granting those whose transactions can be completed. The bus treats each transmission of data as a single transaction since it behaves with a simple push or data-driven protocol. The bus is used for intra-cluster communication. This includes a round-robin arbiter which chooses the next node to grant access based on round-robin scheme. Once the node gets access, it wraps the processing core pipeline with layers of buffering and is the main level in the PENC architecture that interacts with the bus. The destination core is used by the bus to forward the packet to the appropriate location, and the source core is used by the requesting node to satisfy its corresponding IN instruction. Based on the destination address and the data fields, the recipient core stores the address of the data.

4.2.1.2 Efficient Cluster Memory Access Architecture While the lightweight cores are ideal for DSP kernels that require minimal static data [Bisasky, Chandler, & Mohsenin2012], [Bisasky *et al.*2013], ML kernels often require larger amounts of memory for their model data. This is addressed with the distributed cluster-level shared memory (DCM), that is interfaced to the bus. The shared memory within a cluster consists of 3 instances of SRAM cells of memory size 1024x16 bits making up a total of 3072 words and can be accessed within the cluster using the bus and from other clusters through the router. To access the memory, cores use two memory instructions: `LD` and `ST` [Kulkarni *et al.*2016a]. Using data memory as operands for instructions is still beneficial to using `LD` and `ST` from an efficiency standpoint because of the one-cycle read/write capability. Referencing data from the cluster memory has latency and requires a separate instruction, which reduces the overall instructions per cycle that the pipeline can complete. However, the `LD` and `ST` instructions enable the use of a much larger addressable space, which allows the PENC to support many applications. The next section provides empirical results showing how these manycore specific features are well suited for biomedical applications. PENC manycore architecture is ideally suited for most biomedical applications as it addresses the inherent characteristics and challenges. As previously discussed, biomedical applications process a large number of physiological signals. Each signal can be processed in parallel in different designated clusters.

4.2.1.3 PENC Platform Evaluation Setup For the PENC manycore, we developed a stand-alone simulator and compiler that take user's code and post-layout hardware results (Figure 4.5.F) [Page *et al.*2017]. Careful attention was paid to this hardware simulator for a fair comparison. The simulator provides cycle accurate results including completion time, instructions, and memory usage per core. It also serves as a reference implementation of the architecture; its purpose is to make testing, refining, and enhancing the architecture easier. Each task of algorithm is first implemented in assembly language on every processing core using manycore simulator. The simulator reads in the code as well as initializes the register file and data memory in each core. It then models the functionality of the processor and calculates the final state of register files and data memories. Binary files generated by manycore compiler are used to program each core individually. For execution time and energy consumption analysis of the algorithm, binaries obtained from manycore compiler are mapped on to hardware design of the manycore platform (in verilog) and simulated using Cadence NC-Verilog. The activity factor is then derived and is used by the Cadence Encounter tool for accurate power estimation of application. The manycore simulator reports statistics such as the number of cycles required for Arithmetic Logic Unit (ALU), branch, and communication instructions which are used for the throughput and energy analysis of the PENC manycore architecture.

4.2.2 Mapping of Stress Detection using the KNN and SVM Classifiers on PENC Manycore

4.2.2.1 Feature Extraction Mapping on Manycore In both KNN and SVM accelerator, a single core from PENC manycore is used to process the window of samples into test vectors of 4 features. Due to a limitation in the current instruction set, it is not possible to send the test vector to all other cores which are responsible for classification. We measured that the broadcasting to 47 cores (16 clusters) for SVM and 72 cores (24 Clusters) for both KNN takes 513 and 770 cycles respectively. This measurement is slightly optimistic because there is not any other traffic.

4.2.2.2 KNN Classifier on PENC Manycore The task graph for mapping of the KNN version of stress detection application onto the PENC manycore can be seen in Figure 4.6.b. The mapping highlights the parallelism that exists in each cluster (3-cores) and inter clusters. One cluster was assigned to perform the feature extraction task from the HR and accelerometers signals. The extracted features were broadcasted to 24 clusters ($24 \times 3 = 72$ Processing cores). Every core inside one cluster finds the three minimum distances between the test sample and a part of the training data (the amount of training data and its corresponding labels which were stored in the shared memory with size of 1024 words). One core in each cluster finds three minimum distances for the given cluster and sends them with their corresponding stress labels to the next cluster. All clusters can perform the minimum distance exploration in parallel while they are serially dependent on receiving data from one another. It means that one core in each cluster is responsible for receiving the three minimum distances from the previous cluster. Most of required calculations in KNN classifier are executed in parallel using the PENC manycore.

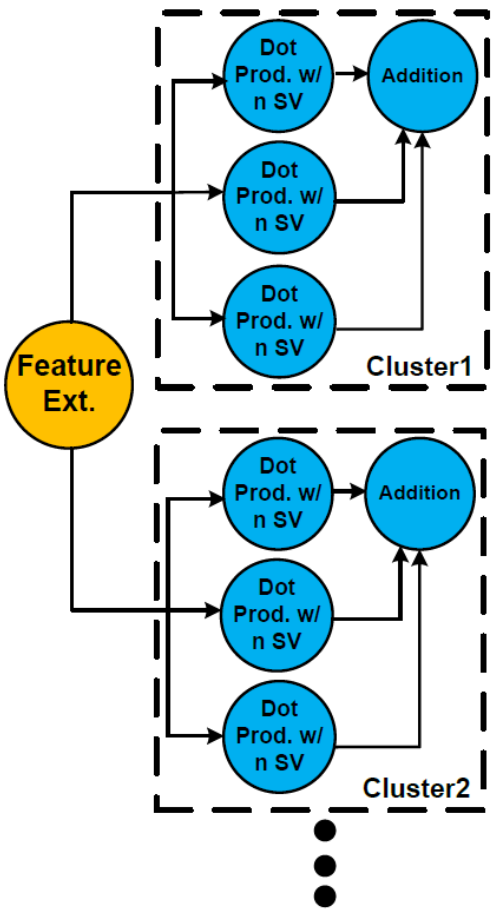
The general KNN model for this given dataset has 16K training data. It needs 24 clusters on PENC manycore to map the entire classifier plus the feature extraction unit.

4.2.2.3 SVM Classifier on PENC Manycore The same as the KNN classifier, the test vector (4 feature set) is generated by feature extraction core which is the input to an SVM classifier. The classifier is trained offline and the support vectors and their corresponding coefficients are stored in the shared memory of each cluster. Figure 4.6.a presents the task graph of SVM mapping on PENC manycore. Each cluster finds the partial dot product results. The final cluster received all these partial results and sums up them to generate the final dot product result. At the end stage this result added with bias value to determine the test sample's label. The general SVM model for the given data set with 6000 support vectors and their coefficients were mapped on PENC manycore with 16 clusters.

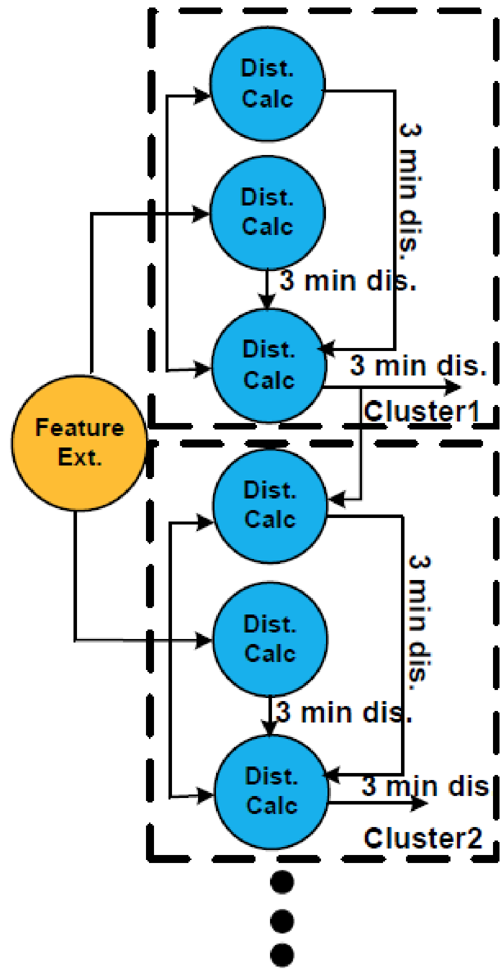
4.2.3 Off-the-shelf Platforms and Experimental Setup

To better gauge the performance of PENC many-core processor for stress monitoring applications, we compared against several commercial off-the-shelf general purpose and programmable processing platforms. In order to do this, we targeted a number of platforms that contain low-power ARM-based CPUs and embedded GPUs.

We measured in real-time both the execution time and power consumption required to classify sample data across a variety of processor combinations. This is achieved by actively recording these metrics for a large number of samples and then averaging to derive the per classification performance. For power results, we measure the power consumption of both the processor and any external memory required. While the Jetson TK1 includes built-in monitoring capabilities, we utilized an external TI INA219 voltage and power IC connected to each system's main power rails to ensure measurement consistency which is shown in Fig 4.7. For each platform, great care was taken to disconnect and power off all other peripherals including HDMI, debug circuitry, and Wi-Fi/Bluetooth. The following discusses details of the targeted platforms including the board capabilities, processors included, and application mappings.



((a)) SVM task graph.



((b)) KNN task graph.

FIG. 4.6: Task graphs of SVM and KNN classifiers for stress detection case study. The graphs highlight the task-level parallelism.

4.2.3.1 Implementation of KNN and SVM Classifiers on NVIDIA Jetson TK1 CPU and GPU NVIDIA's Jetson TK1 is a System-on-Chip (SoC) combining the Kepler graphics processing unit (GPU) and a 4-plus-1 ARM processor arrangement. The 4-plus-1 processor configuration consists of five Cortex ARM-A15 processors, four high performance and one low power processor. Each ARM A15 CPU has a 32KB L1 data and instruction cache supporting 128-bit NEON™ general-purpose single instruction and SIMD instructions. All processors configuration have shared access to a 2MB L2 cache.

Kepler GPU consists of a single Streaming Multiprocessor (SMX) which has a CUDA™ compute capability of 3.2. The Jetson TK1 has 2GB of DDR3 memory that is shared between the CPU and GPU and is rated up to 933MHz. Torch, a scientific computing framework was used to efficiently implement both of these applications on the CPUs and embedded GPU.

We used C and PyCuda to perform stress detection application on Jetson TK1 using its Cortex CPU and Kepler GPU respectively. A complete serial C code was executed on CPU while the data-level parallelization was implemented on the GPU by utilizing multiple threads at the same time. By exploiting the GPU, we were able to achieve several orders of magnitude improvement over the ARM CPU counterpart.

The blocks are completely utilized by mapping data to threads. We mapped the data to multiple blocks and used global memory to establish communication between threads in different blocks.

The SVM classifier was mapped on both CPU and GPU with 6000 support vectors.

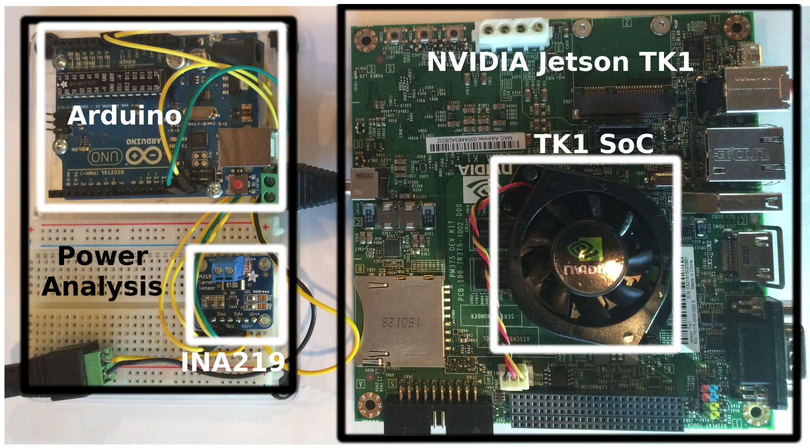


FIG. 4.7: Experimental setup to obtain power and execution time measurements for NVIDIA Jetson TK1 using T1 INA219 and Arduino. For the experiments both ARM A15 CPU core and embedded GPU are targeted.

The entire stress monitoring ill was mapped on 6 blocks The KNN classifier with 16K training data was implemented using the CPU and GPU as well.

4.3 Implementation Results and Platform Comparison

For stress detection application, the feature extraction plus classifier are implemented on different hardware-based and software-based platforms including ASIC, Artix-7 100T FPGA on Nexys board, PENC manycore with Intel Edison acting as host and NVIDIA TK1 ARM CPU and Kepler GPU.

The ASIC implementation was proposed for personalized stress detection which uses each participant's data. The other platforms were used to examine the generalized stress detection model which utilizes the data from all participants.

The post layout design for SVM and KNN indicates that the SVM processor outperforms the KNN processor in aspect of power consumption, execution time and the accuracy for personalized stress detection. At the same frequency of 250 MHz the proposed SVM processor consumes 23x less energy and requires 12x less execution time to make one decision compared to the KNN processing.

For a fair comparison among different platforms, the power of Nexys board is added to FPGA results and the power of the Atom processor was added to PENC. Tables 4.2 and 4.3 demonstrate the comparison results for all platforms for both KNN and SVM versions (generalized model) of stress detection respectively. Due to the presence of task and data level parallelism in PENC architecture, The throughput of PENC accelerator is significantly higher than other platforms (56,691 dec/sec for KNN implementation and 111,111 dec/sec for SVM implementation). The experimental results indicate that for larger design such as KNN with 16K training data, PENC accelerator improves the energy efficiency by 4.7x and 268x over the FPGA and GPU, respectively. For SVM Implementation with 6000 support vectors as a smaller design, the energy consumption of PENC and FPGA are in the same rang.

To better understand the benefit of PENC manycore 4.8 provides comparisons of PENC manycore to processor combination in terms of energy-delay-product (EDP) for stress detection case study with KNN and SVM classifiers. The PENC has significantly lower EDP than all other processors for both ML kernels. Minimizing EDP is very important for biomedical applications as it is critical to both promptly making decisions and doing so with minimal energy. The custom FPGA solution achieves the second best EDP but has the main disadvantage of long development time to design hardware-defined solu-

Processor	Clock (MHz)	Power (mW)	Chip Area (mm^2)	Throughput (dec/sec)	Energy (mJ)	Energy Efficiency (dec/sec/watt)	Energy Efficiency Improvement (over baseline)
TK1 CPU (baseline)	1,092	3,746	529	3.17	1,180	0.84	1x
TK1 GPU	852	5,832.225	529	244.44	23.86	41.91	49x
Artix-7 100T FPGA	166	833	361	2,000	0.41	2,400	2835x
PENC Manycore + Atom	1,000	5,050	175	56,961	0.088	11,279	13,318x

Table 4.2: Breakdown of hardware results from running stress detection applications on a variety of processing platforms (Feature Extraction + KNN classifier with 16k training samples). Results include throughput, energy and energy efficiency. CPU has a fully serial implementation on a single A15 core and is used as baseline. For FPGA the power of Nexys board is added. The power of Atom processor is added to PENC.

Processor	Clock (MHz)	Power (mW)	Chip Area (mm^2)	Throughput (dec/sec)	Energy (mJ)	Energy Efficiency (dec/sec/watt)	Energy Efficiency Improvement (over baseline)
TK1 CPU (baseline)	1,092	3,746	529	14.28	262.22	3.81	1x
TK1 GPU	852	5,723	529	350.5	16.33	61.23	16.05x
Artix-7 200T FPGA	166	708	361	27,394	0.025	38,692	10,145x
PENC Manycore + Atom	1,000	3,341	175	111,111	0.028	33,251	8,719x

Table 4.3: Breakdown of hardware results from running stress detection applications on a variety of processing platforms (Feature Extraction + SVM classifier with 6000 support vectors). Results include throughput, energy and energy efficiency. CPU has a fully serial implementation on a single A15 core and is used as baseline. For FPGA the power of Nexys board is added. The power of Atom processor is added to PENC.

tion. Furthermore, the PENC manycore requires 133x and 3.5x lower EDP compared to FPGA solution for stress detection with KNN and SVM kernels respectively.

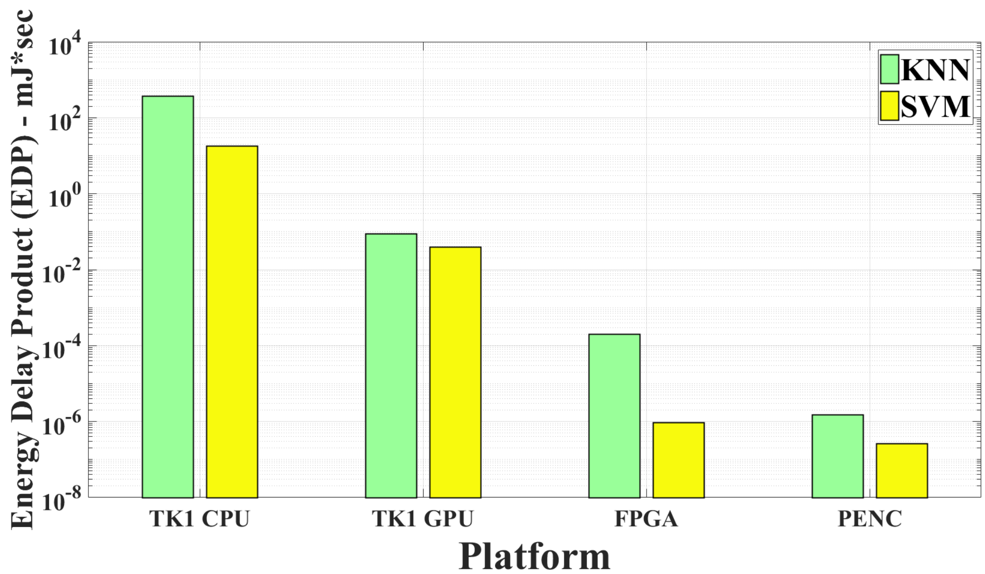


FIG. 4.8: Comparison of energy-delay-product (EDP) for stress detection case study with KNN and SVM classifiers when implemented on several processor combinations including Jetson TK1 CPU and GPU, Artix-7 FPGA and PENC manycore.

Chapter 5

CONCLUSION

5.1 Result Summary

Health monitoring applications share strong commonalities, including requiring sampling from several physiological signals at various rates, preprocessing as well as removing noise, feature extraction and machine learning kernels. In this research, we demonstrated an accurate stress-monitoring system by utilizing multiple physiological signals. Our analysis indicated that using heart rate and accelerometer signals for determining the level of stress generated the most accurate classification with both KNN and SVM classifiers.The average accuracy of the personalized stress monitoring system with KNN and SVM classifiers are 95.8% and 96.7% respectively.

We also presented the post-layout (ASIC) implementation of the SVM and KNN processors which minimizes power consumption and maintains a low-area footprint for for personalized stress monitoring. This is particularly critical when real-world applications are considered.

This research also explored the choice of embedded processors for energy-efficient processing of physiological signals for the multi-modal stress detection application. Three software-based (CPU, GPU and PENC manycore) and one hardware-based (FPGA) platforms were compared. This comparison was based on throughput, power and energy efficiency. To further push the energy-efficiency, we utilized a custom lightweight, symmetric manycore architecture which enables exploiting task-level and data-level parallelism within ML kernels.

The implementation results revealed that the PENC improves the energy efficiency by 5x and 268x over FPGA and TK1 GPU. For KNN mapping with 16K training data, as a large design, The PENC manycore was found to have the highest throughput (decision/sec) and lowest energy usage among other platforms. The small EDP value for PENC makes it an appropriate candidate for stress detection and similar wearable biomedical applications.

5.2 Future Work

There are several options to extend this research. Firstly, this dataset has several performance recordings including correct decisions (The number of shooting to enemy targets) and incorrect decisions (The number of shooting to friend targets) per each participant in three different conditions. We started working to find an appropriate model for task performance based on the heart rate variability (sympathetic and parasympathetic activities) and the received feedback. Our primary model showed the relation between the correct decision and parasympathetic activity. This approach can be extended to explore different effective parameters on task performance.

Secondly, there are other physiological recordings in this dataset which we did not evaluate. We chose the most important signals according to previous research on stress detection.

Thirdly, we can investigate utilizing the deep neural network approach for both feature extraction and stress classification.

Further, we can apply the features and classifiers of this research to other larger dataset, to find a more precise model to detect the level of stress.

Finally, we can extend this research to predict individuals' emotional states (more than the stress level) using the physiological sensors in combination with face and voice features.

Appendix A

ABBREVIATIONS

b	bias parameter in SVM classifier
K	Number of nearest neighbors in KNN classifier
RR	The interval between successive R peaks in ECG
S	support vector in SVM classifier

ACC	Accelerometer
ARL	Army Research Laboratory
ASIC	Application Specific Integrated Circuit
BVP	Blood Volume Pulse
CAM	Content-Addressable Memory
CMOS	Complementary Metal-Oxide-Semiconductor
DCM	Distributed Cluster-level Shared Memory
DSP	Digital Signal Processing
ECG or EKG	Electrocardiogram
EDA	Electrodermal Activity
EDP	Energy Delay Product
EMG	Electromyogram
FIFO	First In First Out
FPGA	Field Programmable Gate Array
FSM	Finite State Machine

GPU	Graphic Processing Unit
GSR	Galvanic Skin Conductivity
HDL	Hardware Description Language
HR	Heart Rate
HRV	Heart Rate Variability
KNN	K Nearest Neighbor
LDA	Linear Discriminant Analysis
PCA	Principle Component Analysis
ML	Machine Learning
PENC	Power Efficient Nano Cluster
RESP	Respiration
RTL	Register Transfer Level
SCR	Skin Conductance Response
SOC	System on Chip
SpO_2	Oxygen Saturation
STD	Standard Deviation
SVM	Support Vector Machine
TSMC	Taiwan Semiconductor Manufacturing company

REFERENCES

[AHR] Aware home research initiative. `http://www.cc.gatech.edu/fce/ahri/projects/index.html`. Accessed: 2017-03-23.

[Attaran, Brooks, & Mohsenin2016] Attaran, N.; Brooks, J.; and Mohsenin, T. 2016. A low-power multi-physiological monitoring processor for stress detection. In *2016 IEEE SENSORS*, 1–3.

[Bisasky *et al.*2013] Bisasky, J.; Homayoun, H.; Yazdani, F.; and Mohsenin, T. 2013. A 64-core platform for biomedical signal processing. In *Quality Electronic Design (ISQED), 2013 14th International Symposium on*, 368–372. IEEE.

[Bisasky, Chandler, & Mohsenin2012] Bisasky, J.; Chandler, D.; and Mohsenin, T. 2012. A many-core platform implemented for multi-channel seizure detection. In *Circuits and Systems (ISCAS), IEEE International Symposium on*, 564–567.

[Deng *et al.*2012] Deng, Y.; Wu, Z.; Chu, C. H.; and Yang, T. 2012. Evaluating feature selection for stress identification. In *2012 IEEE 13th International Conference on Information Reuse Integration (IRI)*, 584–591.

[EEH] EEHPC lab. `http://eehpc.csee.umbc.edu/`. Accessed: 2017-03-26.

[Frank *et al.*2013] Frank, K.; Robertson, P.; Gross, M.; and Wiesner, K. 2013. Sensor-based identification of human stress levels. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, 127–132. IEEE.

[Healey & Picard2005] Healey, J. A., and Picard, R. W. 2005. Detecting stress during real-world driving tasks using physiological sensors. *IEEE Transactions on Intelligent Transportation Systems* 6(2):156–166.

[Holzinger, Bruschi, & Eder2013] Holzinger, A.; Bruschi, M.; and Eder, W. 2013. On interactive data visualization of physiological low-cost-sensor data with focus on mental stress. In *International Conference on Availability, Reliability, and Security*, 469–480. Springer.

[Jeong *et al.*2007] Jeong, I. C.; Lee, D. H.; Park, S. W.; Ko, J. I.; and Yoon, H. R. 2007. Automobile driver's stress index provision system that utilizes electrocardiogram. In *Intelligent Vehicles Symposium, 2007 IEEE*, 652–656. IEEE.

[Kulkarni *et al.*2016a] Kulkarni, A.; Abtahi, T.; Smith, E.; and Mohsenin, T. 2016a. Low energy sketching engines on many-core platform for big data acceleration. In *Proceedings of the 26th Edition on Great Lakes Symposium on VLSI*, GLSVLSI '16, 57–62. New York, NY, USA: ACM.

[Kulkarni *et al.*2016b] Kulkarni, A.; Jafari, A.; Sagedy, C.; and Mohsenin, T. 2016b. Sketching-based high-performance biomedical big data processing accelerator. In *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on*, 1138–1141.

[Kulkarni *et al.*2016c] Kulkarni, A.; Pino, Y.; French, M.; and Mohsenin, T. 2016c. Real-time anomaly detection framework for many-core router through machine-learning techniques. *Journal on Emerging Technologies in Computing (JETC)* 13(1):10:1–10:22.

[LAC] Laboratory for assisted cognition project. `http://www.cs.rochester.edu/u/kautz/ac/`. Accessed: 2017-03-23.

[Lubin & Zuckerman1999] Lubin, B., and Zuckerman, M. 1999. *Manual for the MAACL-R: Multiple Affect Adjective Check List-Revised*. EdITS/Educational and Industrial Testing Service.

[McDuff, Gontarek, & Picard2014] McDuff, D.; Gontarek, S.; and Picard, R. 2014. Remote measurement of cognitive stress via heart rate variability. In *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, 2957–2960. IEEE.

[Ollander2015] Ollander, S. 2015. Wearable sensor data fusion for human stress estimation. *Masters Thesis, Linkopings university Department of Psychology*.

[Page *et al.*2015] Page, A.; Sagedy, C.; Smith, E.; Attaran, N.; Oates, T.; and Mohsenin, T. 2015. A flexible multichannel eeg feature extractor and classifier for seizure detection. *IEEE Transactions on Circuits and Systems II: Express Briefs* 62(2):109–113.

[Page *et al.*2016] Page, A.; Attaran, N.; Shea, C.; Homayoun, H.; and Mohsenin, T. 2016. Low-power manycore accelerator for personalized biomedical applications. In *Proceedings of the 26th Edition on Great Lakes Symposium on VLSI*, GLSVLSI '16, 63–68. New York, NY, USA: ACM.

[Page *et al.*2017] Page, A.; Adwaya, K.; Attaran, N.; Jafari, A.; Homayoun, H.; and Mohsenin, T. 2017. A low-power manycore accelerator for personalized biomedical applications. *IEEE transaction on Very Large Scale Integrates (VLSI) Systems*.

[Patton2013] Patton, D. 2013. How good is real enough? 300 degree of virtual immersion. *Masters Thesis, Towson University Department of Psychology*.

[Riera *et al.*2012] Riera, A.; Soria-Frisch, A.; Albajes-Eizagirre, A.; Cipresso, P.; Grau, C.; Dunne, S.; Ruffini, G.; and aStarlab Barcelona, S. 2012. Electro-physiological data fusion for stress detection. *Studies in health technology and informatics* 181:228–32.

[Rigas *et al.*2008] Rigas, G.; Katsis, C. D.; Bougia, P.; and Fotiadis, D. I. 2008. A reasoning-based framework for car driver x2019;s stress prediction. In *2008 16th Mediterranean Conference on Control and Automation*, 627–632.

[Salai, Vassányi, & Kósa2016] Salai, M.; Vassányi, I.; and Kósa, I. 2016. Stress detection using low cost heart rate sensors. *Journal of Healthcare Engineering* 2016.

[Setz *et al.*2010] Setz, C.; Arnrich, B.; Schumm, J.; La Marca, R.; Tröster, G.; and Ehlert, U. 2010. Discriminating stress from cognitive load using a wearable eda device. *IEEE Transactions on information technology in biomedicine* 14(2):410–417.

[Sharma & Gedeon2012] Sharma, N., and Gedeon, T. 2012. Objective measures, sensors and computational techniques for stress recognition and classification: A survey. *Computer methods and programs in biomedicine* 108(3):1287–1301.

[Shi *et al.*2010] Shi, Y.; Nguyen, M. H.; Blitz, P.; French, B.; Fisk, S.; De la Torre, F.; Smailagic, A.; Siewiorek, D. P.; al'Absi, M.; Ertin, E.; et al. 2010. Personalized stress detection from physiological measurements. In *International symposium on quality of life technology*, 28–29.

[Shiwu *et al.*2011] Shiwu, L.; Linhong, W.; Zhifa, Y.; Bingkui, J.; Feiyan, Q.; and Zhongkai, Y. 2011. An active driver fatigue identification technique using multiple physiological features. In *Mechatronic Science, Electric Engineering and Computer (MEC), 2011 International Conference on*, 733–737. IEEE.

[Sun *et al.*2010] Sun, F.-T.; Kuo, C.; Cheng, H.-T.; Buthpitiya, S.; Collins, P.; and Griss, M. 2010. Activity-aware mental stress detection using physiological sensors. In *International Conference on Mobile Computing, Applications, and Services*, 211–230. Springer.

[Tavana *et al.*2014] Tavana, M. K.; Kulkarni, A.; Rahimi, A.; Mohsenin, T.; and Homayoun, H. 2014. Energy-efficient mapping of biomedical applications on domain-specific accelerator under process variation. In *Low Power Electronics and Design (ISLPED), 2014 IEEE/ACM International Symposium on*, 275–278. IEEE.

[Viseh2014] Viseh, S. 2014. A low power on-board processor for a tangue assistive device. *Masters Thesis, University of Maryland Baltimore County*.

[Wac & Tsiourti2014] Wac, K., and Tsiourti, C. 2014. Ambulatory assessment of affect: Survey of sensor systems for monitoring of autonomic nervous systems activation in emotion. *IEEE Transactions on Affective Computing* 5(3):251–272.

[Wagner, Kim, & Andre2005] Wagner, J.; Kim, J.; and Andre, E. 2005. From physiological signals to emotions: Implementing and comparing selected methods for feature extraction and classification. In *2005 IEEE International Conference on Multimedia and Expo*, 940–943.

[Wijsman *et al.*2011] Wijsman, J.; Grundlehner, B.; Liu, H.; Hermens, H.; and Penders, J. 2011. Towards mental stress detection using wearable physiological sensors. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, 1798–1801. IEEE.

[Zhai & Barreto2006] Zhai, J., and Barreto, A. 2006. Stress recognition using non-invasive technology. In *FLAIRS Conference*, 395–401.

[Zhai *et al.*2005] Zhai, J.; Barreto, A. B.; Chin, C.; and Li, C. 2005. Realization of stress detection using psychophysiological signals for improvement of human-computer interactions. In *SoutheastCon, 2005. Proceedings. IEEE*, 415–420. IEEE.