A FRAMEWORK FOR THE USE OF WEARABLES TO ENABLE STUDY OF STRESS.

by

Kanchinadam Teja Simha

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in
Computer Science

Charlotte

2016

Approved by:

____
Dr. Yu Wang

____
Dr. Jamie Payton

____
Dr. Sara Levens

ACKNOWLEDGEMENTS

Firstly, I would like to thank my advisor Dr. Jamie Payton. I have worked with her throughout my masters and she was always available whenever I had a trouble understanding something on research or writing. She is a great mentor and continuously steered me in the right direction whenever she thought it’s necessary.

I would also like to thank Prof. Yu Wang and Dr. Sara Levens for their valuable support, guidance and systematic approach which helped me to complete this thesis. A special thanks to my lab mates Zaire Ali and Priyanka Mehta for their support, guidance and contribution to this study.

I would like to extend my thanks to all the participants who contributed to this study and a big thanks to UNC Charlotte for giving me an opportunity to present this work.

Lastly, I sincerely want to express my profound gratitude to my parents and my sister for their support and encouragement which helped me to complete this thesis. This accomplishment would not have been possible without them.

ABSTRACT

KANCHINADAM TEJA SIMHA. A framework for the use of wearables to enable study of stress. (Under the direction of DR. YU WANG)

Stress is a common condition that has major health impacts. Even though short-term stress helps people to react to danger immediately, chronic stress has larger health impacts such as early aging, post-traumatic stress disorder, and depression. In this thesis, I propose a continuous stress monitoring approach that can potentially lead to an improved understanding of stress. The approach applies machine learning algorithms to data collected from embedded sensors on a commercially-available wearable device in order to automatically recognize physiological symptoms of stress. Existing approaches for stress detection have typically required specialized sensor equipment or extensive manual labeling of data collected by a researcher in a laboratory setting. A distinguishing feature of our approach is the application of semi-supervised learning algorithms to a data set collected from study participants as they pursue their everyday activities in natural settings. Such an approach provides a foundation for a wearable system that can serve as a platform for use by researchers who wish to collect data about stress from a large population of study participants over extended periods of time, as well as a health and well-being application that alerts users to periods when they are experiencing physiological symptoms associated with stress. Preliminary results for a small study with 8 participants show that my approach can classify physiological symptoms of stress with an average accuracy of 86.3% in per-subject analysis and 75.9% in between-subjects analysis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I: INTRODUCTION

Stress is a common condition that has major health impacts. In a scientific study of workplaces, over 77% of participants stated that they regularly experience physical symptoms caused by stress [1]. Numerous emotional and physical disorders have been linked to stress including depression, anxiety, heart attacks, stroke, hypertension, immune system disturbances that increase susceptibility to infections [1]. Even those who are physically fit are negatively impacted by stress in their personal and professional lives [2].

An improved understanding of stress can potentially improve the health and well-being of millions of people. While an extensive amount of research has examined the health impacts of stress [46—50], there are many questions posed that remain unexplored. Researchers who want to explore stress, however, are limited to time-intensive and expensive options that limit the duration of observation of study subjects. Many existing stress monitoring approaches require study in a laboratory or clinical setting, in which participants are required to wear clinical equipment while being observed by a medical professional. Other approaches to stress monitoring rely on participants to self-report their stress through daily stress diaries and batteries of questionnaires. It is quite challenging for participants to answer questionnaires in real-time, which can lead to forgotten or misreported activities and stress-related experiences. In addition, the reporting required can be extensive and burdensome for the participants.

An emerging approach that holds promise is the use of wearable devices for continuous and less obtrusive stress monitoring. As commercially-available smartwatches, like that shown in Figure 1.1, become more widely adopted, they offer opportunities to study stress across a larger population of study participants without the need for specialized equipment. Modern smartwatches are equipped with sensors that are capable of sampling some of the physiological characteristics associated with stress. Specifically, smartwatches commonly include a heart rate sensor, which can capture the kinds of irregular heartrate associated with stress, and a galvanic skin response sensor, which captures the electrical component variation in the skin and has been shown to be a useful indicator for measuring stress [8, 9, 12, 38]. The raw data extracted from these sensors can then be analyzed to identify signatures in the signals of stress experienced by the smartwatch wearer.

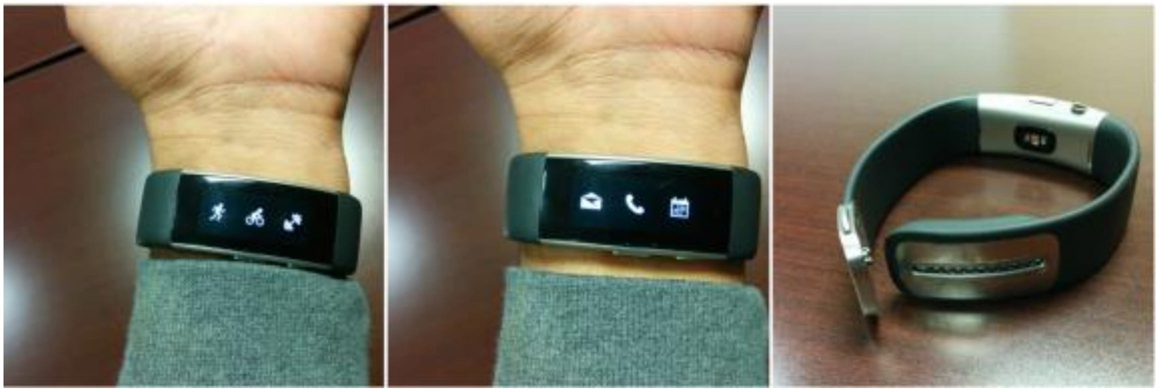

Figure 1.1: Microsoft Smart Band-2

In this thesis, I explore the potential for a wearable system that applies machine learning algorithms to the raw data collected from embedded sensors on a commercially-available smartwatch device to automatically detect physiological symptoms of stress. A machine learning algorithm uses sample input data collected about a phenomenon to learn a model that can be used to make predictions. While existing approaches have examined the application of supervised machine learning to wearable sensor data for stress detection, these approaches have required extensive observation and manual labeling of ground-truth data in a laboratory setting [8, 9, 10, 11, 12, 13, 38]. A distinguishing feature of my approach is the application of *semi-supervised machine learning algorithms* to a data set collected from study participants *in natural settings*. Such an approach provides the ability to monitor study subjects as they pursue their everyday activities over an extended duration of time, without requiring extensive self-reporting, observation, or clinical equipment. Results of a pilot study with 8 users demonstrate the feasibility of this approach, which can classify periods of stress at 75.9% accuracy when learning a model that is generalized across all users and, on average, achieves 86.3% accuracy when the model is tailored to a particular user.

CHAPTER II: MOTIVATION

Machine learning is a well-suited and commonly applied approach to detect activities or conditions from sensor data [3—7]. Given a set of inputs, a machine learning algorithm will use those inputs to “learn” to produce the correct output when encountering a new input. Essentially, the machine learning algorithm is finding a model that fits the known data, and is then using the model to make predictions over new data. The algorithmic process of learning from inputs is called *training* and the process of predicting the associated outputs of new or unseen inputs is called *testing*. If the outputs associated with the learning task are continuous, the machine learning algorithm performs *regression*, which is the prediction of a real value. Predicting a value along a continuous spectrum that reflects the degree of stress that a person is experiencing is an example of a regression problem. If the learning task requires discrete outputs, the process of predicting the output is called *classification.* Predicting if an experience can either be labeled as belonging to the class “stress” or “not stress” is an example of classification. In this thesis, we perform classification to detect stress, where the inputs are physiological responses of stress measured by the Galvanic Skin Response sensor and Heart Rate sensors.

Supervised learning is a particular kind of machine learning approach in which, given a set of inputs and their corresponding ground truth output labels, the algorithm will learn to produce output labels for new or unseen inputs. Figure 2.1 shows the architecture of supervised learning, where *stress* and *not-stress* are the classes of labels given as part of the training input set to learn the classification model, and are the output labels produced as a result of applying the classification model to new sensor data inputs.

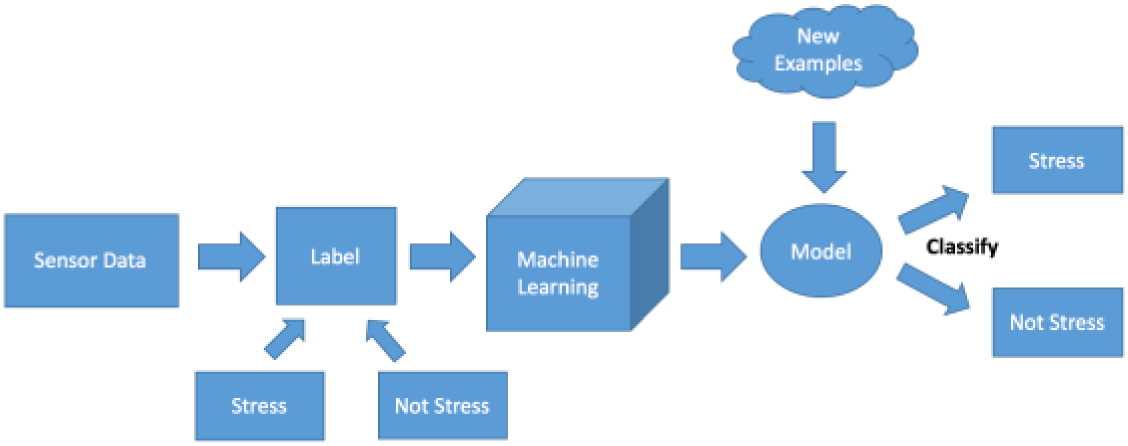

Figure 2.1: Architecture of a supervised learning algorithm.

The limitation of supervised learning is that it is tedious, expensive, and sometimes impossible to acquire a large collection of labeled inputs that can be used as ground truth to derive a model. With respect to the problem of getting input data for supervised stress detection, collecting real-time physiological responses from sensor data is easy but getting their corresponding outputs or labels is often difficult. To get ground truth for the collected sensor data, labeling must be performed at each minute under real-time conditions, which is impractical.

At the other end of the spectrum, an *unsupervised learning* approach requires no labeled inputs. In unsupervised learning, given a set of unlabeled inputs, the algorithm will learn to produce outputs for new inputs. Figure 2.2 illustrates an unsupervised classification approach for stress detection.

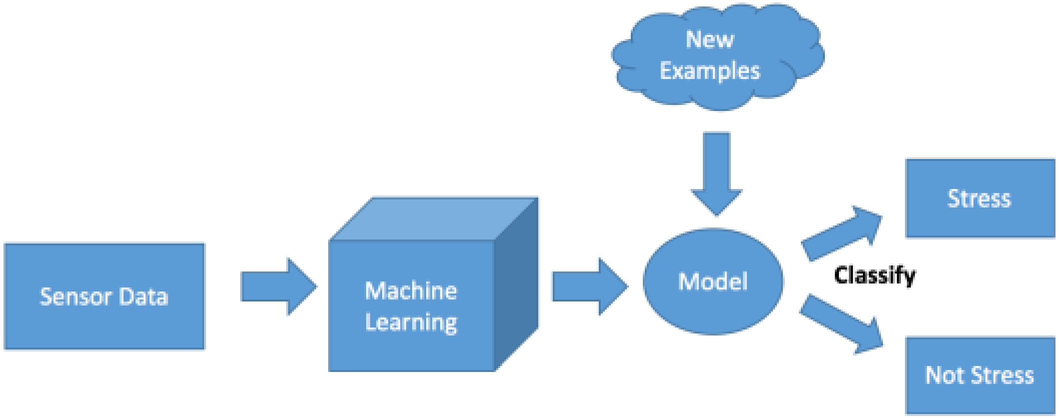

Figure 2.2: Architecture of Unsupervised Learning algorithm.

While unsupervised learning alleviates the need for training sets with large amounts of labeled data, this approach does not perform as well as supervised learning for classification [39]. Between these extremes is an approach called *semi-supervised learning,* which makes use of both labeled and unlabeled data in training. Figure 2.3 illustrates the architecture of a semi-supervised classification algorithm for stress detection. Semi supervised learning has many approaches and they work on certain assumptions which again depend on the type of semi supervised learning approach. Example of semi supervised learning approaches would be *Label Propagation* in which the unlabeled data is classified first using labeled data as training set. Based on the classification, labels are assigned to the unlabeled data. The classified unlabeled data is added to the set of labeled data and will be used for training. Another approach of semi supervised learning would be *K-means clustering* in which the data is clustered or grouped based on the underlying structure of the data. The data will be labeled based on which cluster it belongs to and will be used for training. This approach is *unsupervised* in nature and labels are not used in this approach. The importance of using unlabeled input data for training has been demonstrated; semi-supervised learning results in lower misclassification rates when compared with supervised learning using only labeled data [15—17].

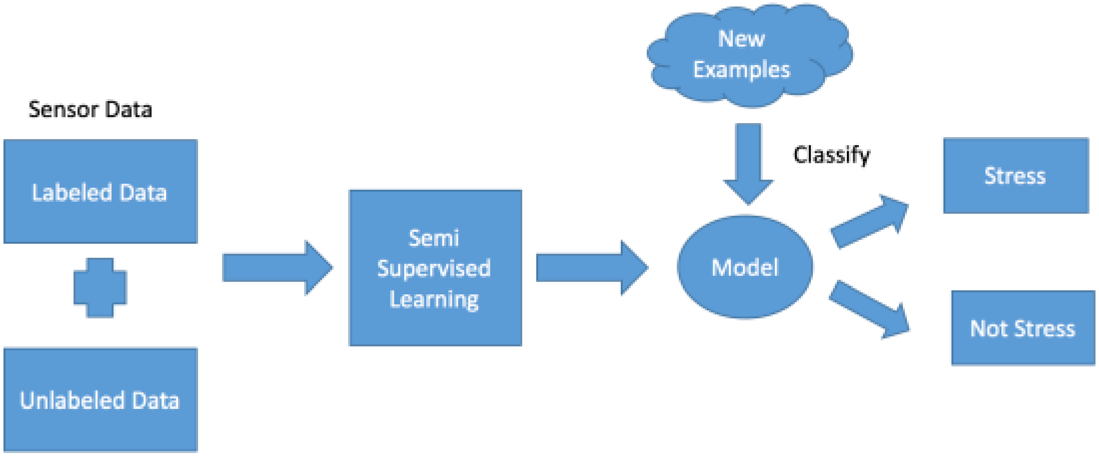

Figure 2.3: Semi supervised learning pipeline

In this thesis, I apply semi-supervised learning to the problem of stress detection over sensor data collected from users in natural settings with a small number of labeled examples.

CHAPTER III: METHODS

From the physiological point of view, stress is a physiological response triggered from the *Autonomic Nervous System (ANS)*. The ANS has two components: The *Sympathetic Nervous System (SNS)* and *Parasympathetic Nervous System (PNS)*. The SNS is responsible for the physiological responses of stress such as increase in heart rate, respiration activity (heavy breathing), and excessive secretion of sweat from sweat glands while the PNS is responsible for bringing these physiological responses to normal state. Most smartwatches include sensors that can capture the physiological responses triggered by the SNS and the PNS, including a heart rate sensor and galvanic skin response.

First, a heart rate sensor, in theory, can be used to capture Heart Rate Variability (HRV). HRV is a physiological response which can be defined as the interval between two successive heartbeats. Under acute stress, HRV is increased for limited duration of time. The Heart Rate (HR) is also a physiological response and is defined as the number of contractions of heart per minute. Several parameters of HR and HRV can be used to measure the physiological response of stress.

Second, *Galvanic Skin Response (GSR)* or *Skin Conductance (SC)* is a physiological response which is a measure of electrical characteristics of the skin. Typically, stress can be measured by continuously monitoring the variation in the electro dermal activity of the skin. It has been demonstrated that even in adverse physical conditions, GSR response is not biased with respect to body activity. SC has two components namely Skin Conductance Response (SCR), which is a fast-changing component, and the skin conductance level. (SCL), which is a slow changing component. The maximum value of the SCR and the latency of SCL are important attributes that can be used to measure stress.

To record physiological responses of stress, we conduct a study in which participants wear smartwatches that collect data related to heart rate variability and galvanic skin response. Specifically, participants in this study were asked to wear a Microsoft Band and carry an Android mobile phone for 3 consecutive days as they go about their normal daily routines. The mobile application deployed on an Android phone communicates with Microsoft Band smartwatch, which has embedded GSR and HR sensors. The Microsoft Band collects the raw sensor data and sends it to the Android application, which stores the data on the phone. To provide the small number of ground truth labels needed for a semi-supervised learning approach, the Android application also provides support for issuing surveys on stress and recording survey responses. Participants were asked to complete two types of surveys: 1) periodic surveys, in which users were asked to answer the questions in the survey in the morning, afternoon and evening of each day of the study and 2) on-demand surveys in which users were asked to participate in the survey right after a particular activity (in this case, smoking or eating) is performed. The surveys, included in Appendix A, are validated instruments related to capturing mood and stress.

Figure 3.1 shows the pipeline for applying machine learning to the data collected in this study for stress detection. In the remainder of this chapter, I describe each stage of the machine learning pipeline, detailing the approaches taken in each stage. The following chapter will present results of applying this pipeline.

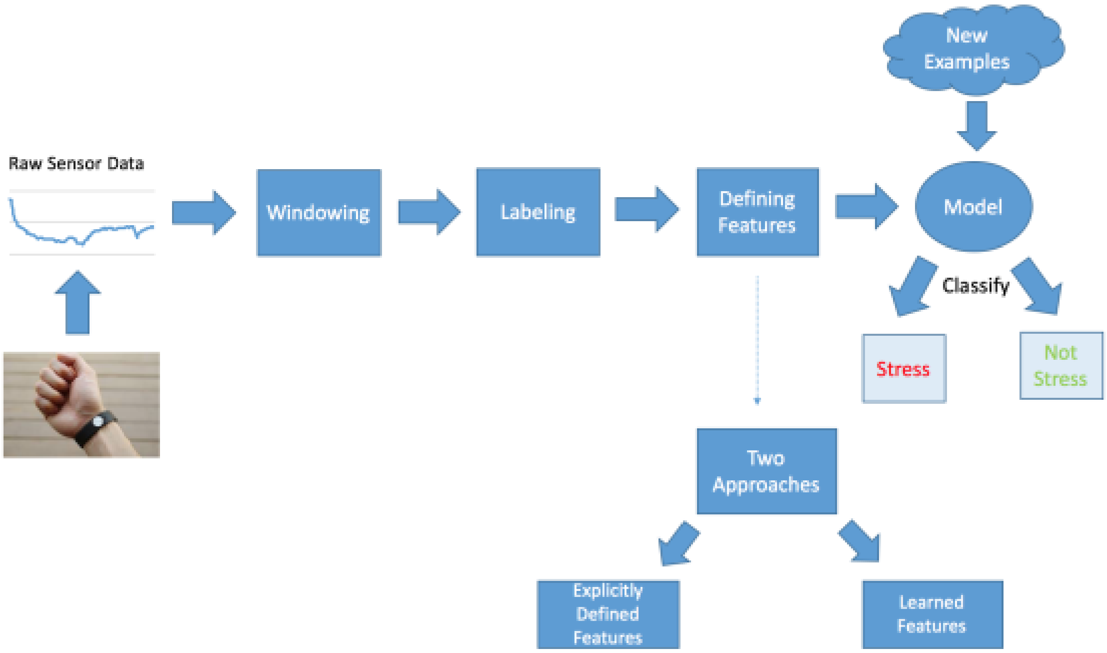

Figure 3.1: Continuous Stress Monitoring system.

3.1 Preprocessing Step: Windowing Raw Data

To detect stress based on the physiological responses, the sensor data needs to be passed to a semi-supervised learning as input (training). Stress is a physiological response and looking at single data point will not effectively capture the stress variation in the signal. Looking at a window of sampled data points will give us better information about the condition over time, as demonstrated in the use of windowed accelerometer data for activity recognition [13]. Using overlapping windows is a desirable approach that gives better performance in activity recognition because it captures information at different phases of the signal; having several samples at different phases will improve the accuracy of the machine learning classification. Fig 3.1.1 and 3.1.2 shows the difference between a regular window and an overlapping window approach. In this study, a 60 second time window with 50% overlapping is implemented. The importance of using 50% overlapping window has been stated in [13] and is considered as the optimal overlapping percentage.

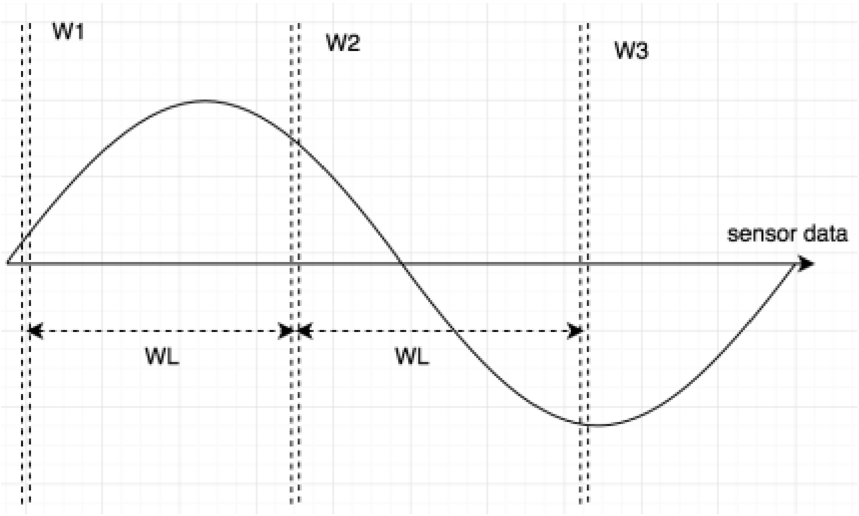

Figure 3.1.1: Representation of regular time window.

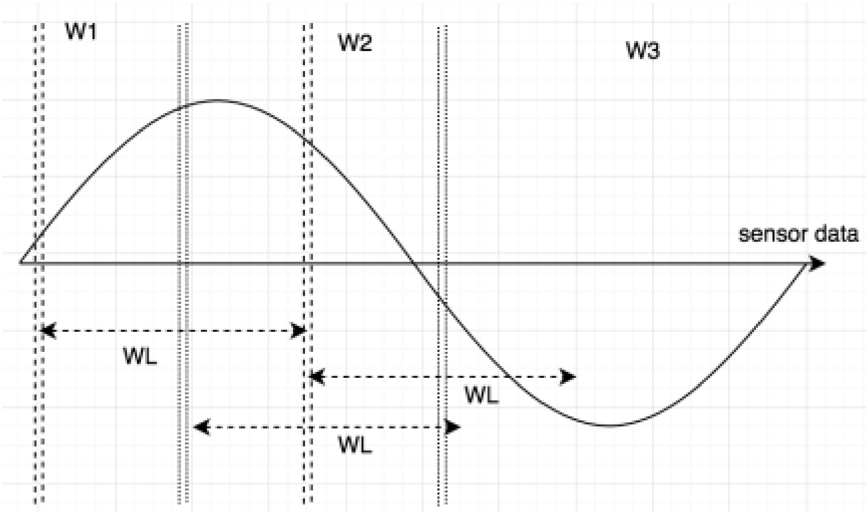

Figure 3.1.2: Representation of an overlapping time window.

3.2 Preprocessing Step: Feature Definition

Input data to machine learning algorithms is defined in terms of features, or properties of the data that are likely important to the characterization of the associated phenomenon the data is collected about. With respect to our data, a feature can be defined as a distinctive attribute of a signal or wave. Similarly, a feature vector can be defined as the set of such distinctive attributes in a signal or a wave. Since the data in each window corresponds to raw sensor data and since the data is time-variant, each window can be considered as a signal. Feature vectors from each window are defined to train a machine learning algorithm. In this study, features from each window are defined in two ways:

- Using the domain knowledge of the physiological responses from GSR and HR, we have explicitly defined features, which are described in Section 4.2.1.

- By understanding the structure of raw sensor data, features are learned using a machine learning algorithm in an unsupervised way. This approach and the resulting features are described in Section 4.2.2.

3.2.1 Explicitly Defined features

Using the domain knowledge of the physiological responses, HR and GSR features are extracted from each window. HR features are calculated over time-domain and spectral-domain analysis. First, Heart Rate Variability (HRV) is calculated from the HR by the formula ($HRV = \frac{60}{HR} * 1000$). Time-domain features such as mean-HR, mean-HRV, standard deviation of HRV and HR are calculated for each window. RMSSD which is the root mean square of the successive difference in heart-rate intervals and pnn50 which is the percentage of R peaks whose value in greater than 50ms are calculated over HRV. PSD estimate using Welch's formula are calculated for HRV via LF and HF. Ratio of LF and HF is also calculated to estimate the signal to noise ratio. The importance of these features has been explained in [8, 9, 14].

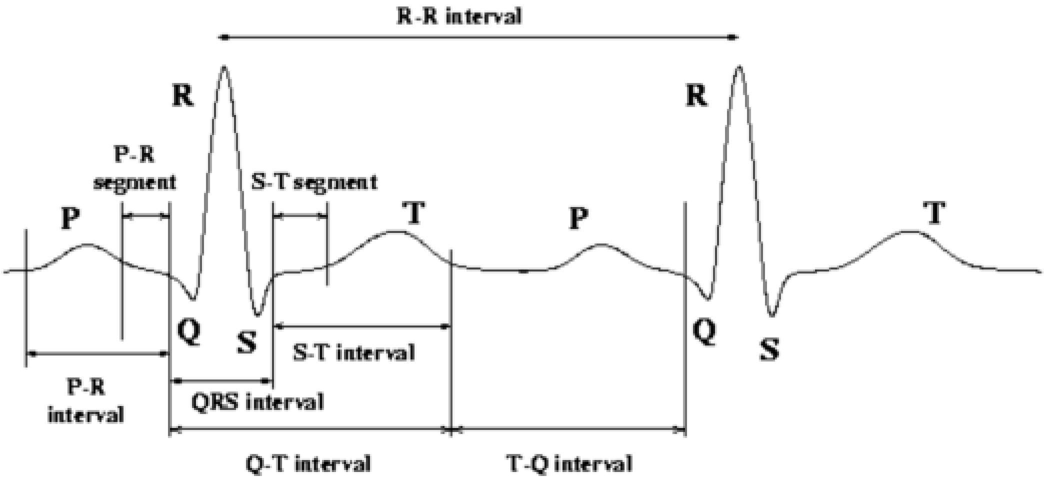

Figure 3.2.1.1 Rutgers University (2003). *Definition of waves, segments and intervals in the normal ECG waveform* [Image]. Retrieved from rutgers.edu/classes/bme/bme305/ModelFitting/ModelFitting_HW.html

The other important heart-rate features would be the QRS interval, P-wave length, Q-wave length, etc. as shown in the Figure 3.2.1.1. These features have not been calculated in this thesis because the sampling rate of MICROSOFT band is 1Hz which means it only gives us the average *R-R interval* value for each second. With wearables such as SHIMMER which have higher sampling rates, we can measure these features and the importance of these features have been demonstrated and are used in cardiology, medical field, etc.

GSR features are calculated by estimating the *startle responses* in a given window. Stress is often associated with instant reflexes and when these reflexes occur there is a sudden change in the physiological symptoms in the body. Sudden change or stimuli in the body is defined as startle responses. The importance of startle responses and the method to extract these responses from skin conductance (GSR) has been demonstrated by Healey and Picard [9]. From Figure 3.2.1.2, startle responses of a signal can be seen as the onset to peak value in a signal. Features such as number of startle responses (number of such onset-peak values), sum of the duration of startle responses (SD), sum of the magnitudes of startle responses are calculated (SM) are extracted.

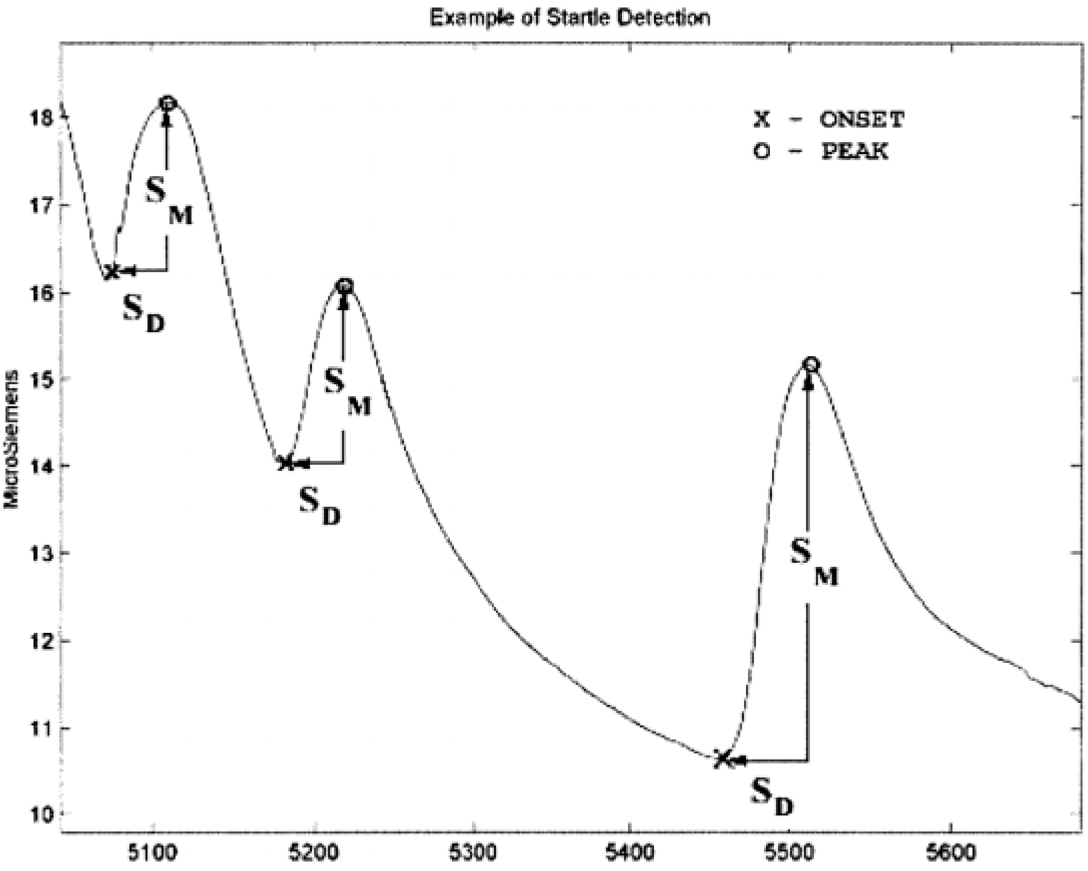

Figure 3.2.1.2 Massachusetts Institute of Technology (2000). *An example of the startle responses occurring in a one and the results of the algorithm showing onset "X" and peak "0" detection. The features SM and SD are calculated as shown*. [Image] Retrieved from J. A. Healey, “Wearable and Automotive Systems for Affect Recognition from Physiology,” Ph.D. dissertation, 2000.

The startle response has been calculated from the same method described in [9] but instead of using an elliptical filter, but [14] have demonstrated the importance of Gaussian filter and after manually observing the startle responses of each user, we have used a sigma value often called as the smoothing function which attenuates the noise from the signal as

0.7 to filter out noisy data. Time-domain features like mean-GSR and standard deviation of GSR are also calculated [8, 9, 12, 14].

All the HR and GSR features are normalized such that they are scaled in the range of [0,1). Normalization is a standard step that ensures that features that are measured in different units are comparable [8].

Table 3.2.1.1: Explicitly Defined Features from Heart rate

Feature	**Feature Set**
Mean HR	Time Domain
Mean RR	Time Domain
RMSSD	Time Domain
pnn50	Time Domain
STD HR	Time Domain
STD RR	Time Domain
LF	Spectral Domain
HF	Spectral Domain
LF/HF	Spectral Domain

Table 3.2.1.2: Explicitly Defined Features from Galvanic Skin Response

Feature	**Feature Set**
Number of startle events	Spectral Domain
Sum of duration of startle events	Spectral Domain
Sum of magnitudes of startle events	Spectral Domain
STD GSR	Time Domain
Mean GSR	Time Domain

We have extracted a total of 14 features ($F_1 – F_{14}$) from HR and GSR sensor data where F_1-F_9 are shown in Table 3.1 and F_9- F_{14} are shown in Table 3.2.

3.2.2 Feature Learning via Contrast Divergence

In a second experiment, we have used Restricted Boltzmann Machines (RBM) [51] to automatically learn features from the raw window of sensor data points. Restricted Boltzmann Machines (RBM) belongs to the class of artificial neural network than can learn the probability distribution over its set of inputs [17]. Contrast divergence algorithm belongs to the class of RBM with a gradient-based approximation of log-likelihood on a short morkov-chain [17]. RBM's learn the probability distribution of points over its set of given inputs and in this study, we have used the Contrast-Divergence (CD) algorithm to automatically learn the features over the window of raw GSR and HR data points. A total of 100 features from each window have been learned using this algorithm. As before, features are normalized to fall in the range of [0,1).

3.3 Preprocessing Step: Defining Labels Over Data

The features computed over the (overlapping) windows of sensor data will serve as input to the machine learning algorithm. To provide labels for our training data set for our semi-supervised learning algorithm, we label each window using survey responses as ground truth. Specifically, we manually label each GSR and heart rate window with the corresponding survey responses as either “stressed” or “not stressed”. The labelling procedure for ground truth is shown in Figure 3.3.1, where $L_1, L_2..., L_N$ are the periodic surveys issued to the users. The surveys are optional to each user and he/she can choose to answer the survey it or not. If the survey responses are answered, then all the corresponding sensor data within a timeframe t of L are labeled with the user’s response of “stress” or “not stressed”. This approach assumes that the subject’s response of the stress condition is valid within time frame *t*. Experiments with various values of t = 60 min, 50 min, 40 min, 30 min, 20 min, and 10 min have been conducted by manually observing the sensor data. Given then results of our experiments, we assume that the responses with the time frame *t=20 min* are valid and use this value for our labelling scheme. All the windows which fall within this time frame are labeled according to the survey response answered by the user and all the windows which does not fall into this timeframe are not labeled and are called unlabeled samples.

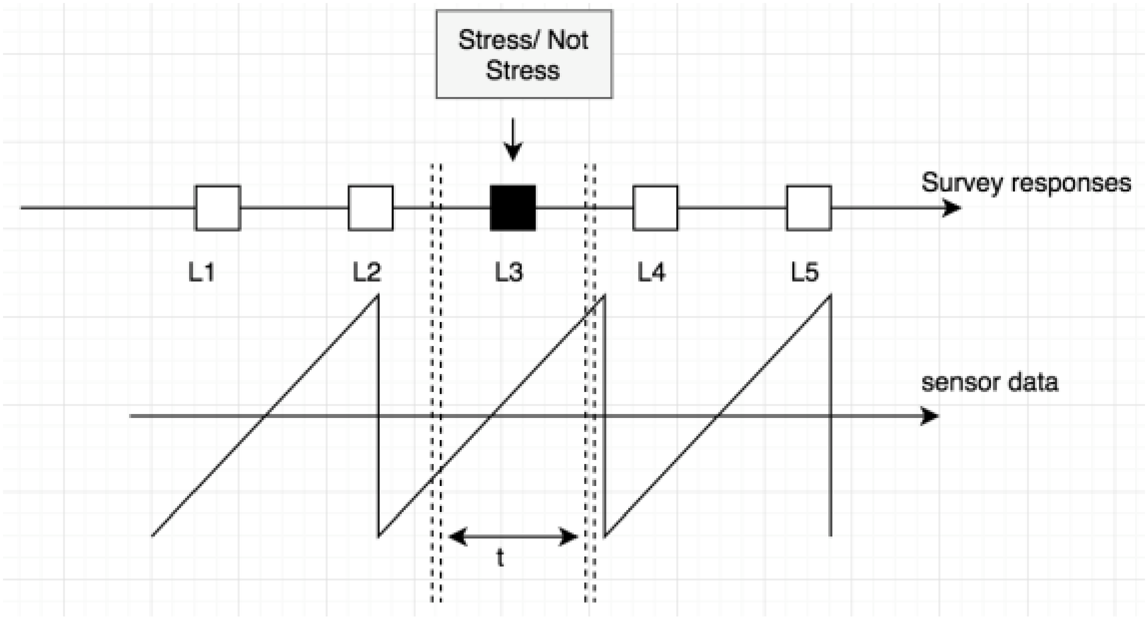

Figure 3.3.1: Labeling sensor data based on survey responses

3.4 Semi-Supervised Learning with Boosting

Out of 222 survey responses, more than 133 survey responses are for *not stress,* 44 responses are for *slightly stressed,* 20 responses are for *somewhat stressed,* 13 survey responses are for *moderately stressed* and one survey response is for *very stressed.* Nearly 20-30 survey responses have no associated sensor data. Since we do not have equal distribution of classes and since *not stress* class is higher than 50% of all the other classes combined, we group reports for “moderately stressed” and “very stressed” into a single category with a label of “stressed”. All other reports are labeled “not stressed”. We can then treat this as a binary classification problem that learns a classification model that can differentiate between two classes: stressed and not stressed.

Selection of the machine learning technique used for classification is an important consideration. My selection of algorithm addresses the challenge of label noise. Since the labeling scheme we have set involves a fixed t=20min threshold, there is a good possibility that there is noise in the labels. Additionally, since the labels are from the survey responses, the responses are subjective and may not correspond as we would expect with physiological symptoms of stress. This kind of label noise will affect the classification accuracy as most classifiers are not robust to label noise. There are many approaches to handle label noise and [26] explains the comprehensive survey of different techniques that reduce label noise. In my approach, to reduce the label noise, I select boosting as the machine learning technique for stress detection. Boosting is a machine learning technique which comprises of an ensemble of weak learners primarily tasked to reduce overfitting, which is a machine learning problem in which the outputs produced are not correct even though the algorithm performs well during learning. Several algorithms of Boosting have been shown to be more robust to label noise such as Regularized AdaBoost, Logit Boost, Brown Boost [26-29].

XGBoost is used for supervised learning problems which belongs to the class of Boosting and it considers boosting as an optimization problem over a cost function with a regularization factor [33]. In this study, I use XGBoost as our base classifier [35] to our semi-supervised learning approaches because of the presence of label noise in the dataset. We explore the use of several semi-supervised learning approaches to our labeled and unlabeled data. In each of our semi-supervised learning approaches, XGBoost as the base classifier. In our experiments, we compare two semi-supervised learning approaches: co-training and self-learning.

3.4.1 Semi-Supervised Learning Approach 1: Co-Training

Co-training belongs to a class of semi-supervised learning which is extensively used in text-mining. It assumes that each example is described using two different feature sets that provide different, complementary information about the instance. Ideally, the two views are conditionally independent and each view is sufficient. Co-training first learns a separate classifier for each view using any labeled examples. The most confident predictions of each classifier on the unlabeled data are then used to iteratively construct additional labeled training data [21]. Figure 3.4.1.1 explains the architecture of co-training where labeled and unlabeled data is used as training data by the co-training algorithm and the classifiers C1 and C2 are the supervised machine learning algorithms.

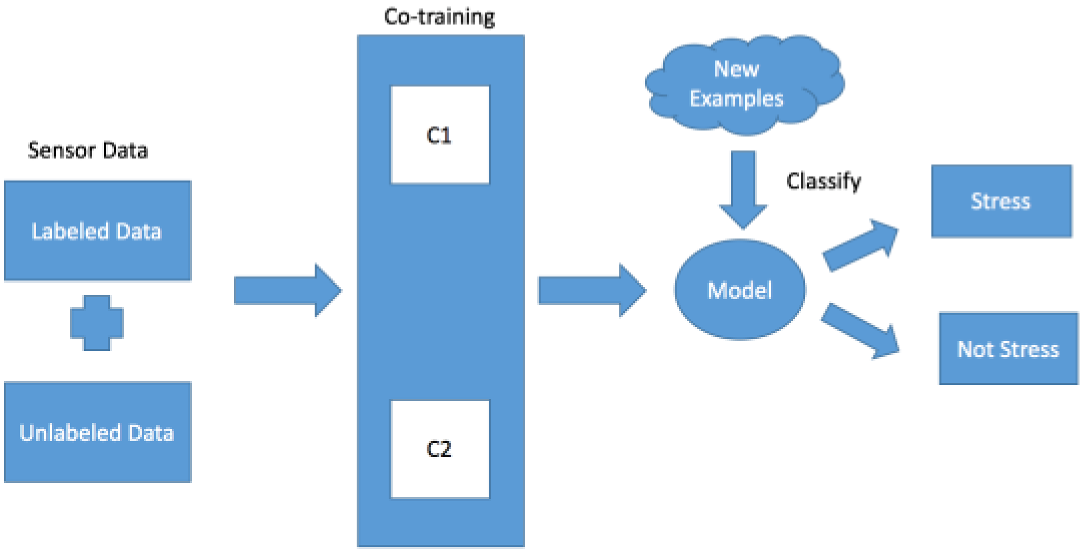

Figure 3.4.1.1: Co-training pipeline

We have used XGBoost as the classifiers (*C1* and *C2*) in our co-training algorithm because Boosting helps to overcome label noise and reduce the overall bias on the classification.

3.4.2 Semi-Supervised Learning Approach 2: Self-Training

Self-training is one of the earliest techniques using both labeled and unlabeled data to improve learning [24, 25]. Given a set of labeled data L and unlabeled data U, self-training proceeds as follows: train a classifier h using L, and classify U with h; select a subset U ⊂ U for which h has the highest confidence scores; add U to L and remove U from U. This process is repeated until the algorithm converges. Figure 3.4.2.1 describes the architecture of self-training where *Clf* is a supervised machine learning algorithm.

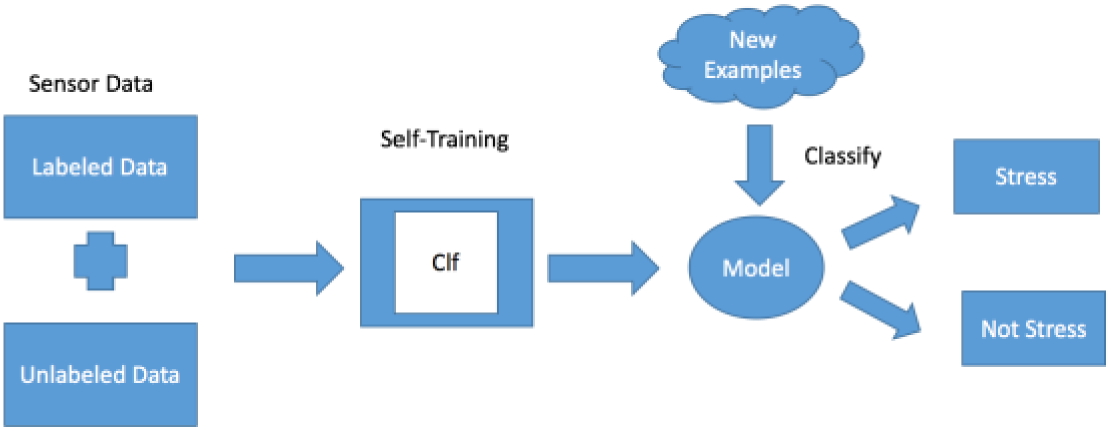

Figure 3.4.2.1: Self-training pipeline

CHAPTER IV: RESULTS

We report results for subject-wise and between-subjects analysis. In a subject-wise experiment, the machine learning algorithm considers only the inputs for a single user to learn a classifier that is tailored to that user. In our subject-wise experiments, we omit two participants as they do not have the required sensor data and labels needed for training a classifier. In a between-subjects experiment, the machine learning algorithm is given the labeled and unlabeled data from all study participants, which results in a generalized classification.

We have applied the following methods to conduct *subject-wise* and *between-subjects* experiments to understand the physiological symptoms of stress for each user:

Methods:

- Experiment with only labeled samples using explicitly defined features (F_1 – F_{14}).
- Experiment with only labeled samples using learned features (L_1 – L_{100}).
- Experiment with labeled and unlabeled samples using explicitly defined features (F_1 – F_{14}) using Co-Training algorithm.
- Experiment with labeled and unlabeled samples using explicitly defined features (F_1- F_{14}) using Self-Training algorithm.

- Experiment with labeled and unlabeled samples using learned features ($L_1 – L_{100}$) using Self- Training algorithm.

4.1 Performance Evaluation Metrics

The accuracy of the classification model can be defined as the percentage of outputs which are correctly predicted by the machine learning algorithm given a set of inputs. Another tool used to evaluate the performance of a classification task is the *confusion matrix* shown in Figure 4.1.1, which shows the ground truth associated with a labeled input and the resulting predicted value.

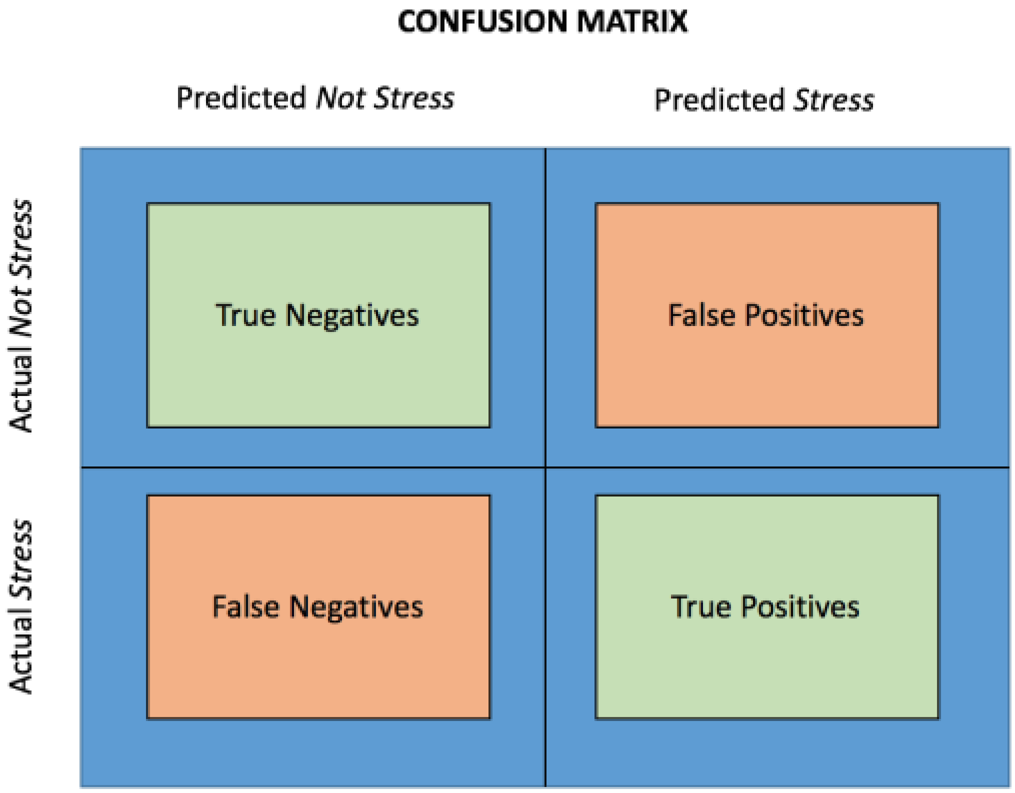

Figure 4.1.1 Confusion matrix.

The confusion matrix provides a visual representation of metrics that are commonly used to evaluate a classification model:

- *True Positives (TP)* indicates the cases where the physiological symptoms states *stress* and the algorithm produces the outputs as *stress.*

- *True Negatives (TN)* indicates the cases where the physiological symptoms states *not stress* and the algorithm produces the outputs as *not stress*.

- *False Positives (FP)* indicates the cases where the physiological symptoms states *not stress* and the algorithm produces the outputs as *stress*. False Positives are often termed as Type-I error.

- *False Negatives(FN)* indicates the cases where the physiological symptoms states *stress* and the algorithm produces the outputs as *not stress*. False Negatives are often termed as Type-II error.

The following performance measures can be calculated based on the confusion matrix:

- *Misclassification rate* which is the percentage of new inputs which are wrongly classified by the classifier can be calculated as *(FP + FN)/Total-number-of-inputs*.

- *Accuracy* can be calculated as *(TP + TN)/Total-number-of-inputs.*

- *Precision,* which is the percentage of examples which are classified as *stress* are actually *stress*.

- *Recall*, which is the percentage of examples which are classified as *not stress* are actually *not stress.*

- *F1-measure* which is the harmonic mean of *precision* and *recall.* F1-score is also a measure of classification accuracy.

- Precision, Recall & F1-measures are calculated for individual classes. To get the combined metric value for the experiment, *average* is calculated. For classification, *Average* is estimated by finding the area under the curve using ROC analysis and the importance of this is described in [52].

To evaluate the performance of our models, we have used Stratified K-fold validation where the data is divided into *K-folds* and each time one fold is used for validation and the remaining folds are used for training the algorithm, repeating the process for K-times. Stratification ensures that each fold has equal distribution of classes. Stratification is generally a better scheme, both in terms of bias and variance, when compared to a traditional cross-validation approach [35].

4.2 Performance Evaluation Results

4.2.1 Method 1: Baseline with Defined Features

To serve as a baseline, we apply supervised machine learning to explicitly defined features. With explicitly defined features (F_1 – F_{14}), we have trained a XGBoost classifier with maximum depth of base learners as 3, learning rate of 0.1 and number of boosted trees as 100. The accuracy was examined using stratified 10-fold cross validation as the model was trained with 75% training and tested with 25% test set for 10 times.

The confusion matrices and the corresponding classification metrics for *subject-wise experiments* are shown below:

Table 4.2.1.1: Confusion matrix for User 1 using Method 1

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	611	151
Actual *Stress*	72	103

Table 4.2.1.2: Classification metrics calculated from Table 4.2.1.1

Class	Precision	Recall	F1-measure
Stress	58.9	40.6	48
Not Stress	80.18	89.45	84.5
Average (ROC)	74	76	75

Table 4.2.1.3: Confusion matrix for User 2 using Method 1

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	500	0
Actual *Stress*	0	197

Table 4.2.1.4: Classification metrics calculated from Table 4.2.1.3

Class	Precision	Recall	F1-measure

Stress	100	100	100
Not Stress	100	100	100
Average (ROC)	100	100	100

Table 4.2.1.5: Confusion matrix for User 3 using Method 1

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	239	10
Actual *Stress*	15	55

Table 4.2.1.6: Classification metrics calculated from Table 4.2.1.5

Class	Precision	Recall	F1-measure
Stress	78.57	84.6	81.48
Not Stress	95.98	94.09	95.02
Average (ROC)	92	92	92

Table 4.2.1.7: Confusion matrix for User 4 using Method 1

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	10	3
Actual *Stress*	5	79

Table 4.2.1.8: Classification metrics calculated from Table 4.2.1.7

Class	Precision	Recall	F1-measure
Stress	94.04	96.34	95.18
Not Stress	76.92	66.7	71.42
Average (ROC)	91	92	92

Table 4.2.1.9: Confusion matrix for User 5 using Method 1

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	49	9
Actual *Stress*	9	29

Table 4.2.1.10: Classification metrics calculated from Table 4.2.1.9

Class	Precision	Recall	F1-measure
Stress	76.31	76.31	76.31
Not Stress	84.48	84.48	84.48
Average (ROC)	81	81	81

Table 4.2.1.11: Confusion matrix for User 6 using Method 1

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	495	127
Actual *Stress*	67	37

Table 4.2.1.12: Classification metrics calculated from Table 4.2.1.11

Class	Precision	Recall	F1-measure
Stress	35.57	22.5	27.6
Not Stress	79.5	88.07	83.61
Average (ROC)	70	71	71

The confusion matrices and the corresponding classification metrics *for between-subjects experiment* is shown below:

Table 4.2.1.13: Confusion matrix for the combined dataset using Method 1

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	1937	524
Actual *Stress*	298	276

Table 4.2.1.14: Classification metrics calculated from Table 4.2.1.13

Class	Precision	Recall	F1-measure
Stress	48.08	34.5	40.17
Not Stress	78.70	86.6	82.4
Average (ROC)	71	73	71

4.2.2. Method 2: Baseline with Learned Features

As a baseline, we implemented supervised learning using learned features (L_1 – L_{100}). Specifically, we have trained a XGBoost classifier with maximum depth of base learners as 3, learning rate of 0.1 and number of boosted trees as 100. The accuracy was examined using stratified 10-fold cross validation as the model was trained with 75% training and tested with 25% test set for 10 times.

The confusion matrices and the corresponding classification metrics for *subject-wise experiments* are shown below:

Table 4.2.2.1: Confusion matrix for User 1 using Method 2

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	583	156
Actual *Stress*	100	88

Table 4.2.2.2: Classification metrics calculated from Table 4.2.2.1

Class	Precision	Recall	F1-measure
Stress	46.80	36.06	40.74
Not Stress	78.89	85.35	81.99
Average (ROC)	69	72	70

Table 4.2.2.3: Confusion matrix for User 2 using Method 2

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	500	0
Actual *Stress*	0	197

Table 4.2.2.4: Classification metrics calculated from Table 4.2.2.3

Class	Precision	Recall	F1-measure
Stress	100	100	100
Not Stress	100	100	100
Average (ROC)	100	100	100

Table 4.2.2.5: Confusion matrix for User 3 using Method 2

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	220	37
Actual *Stress*	24	38

Table 4.2.2.6: Classification metrics calculated from Table 4.2.2.5

Class	Precision	Recall	F1-measure
Stress	61.29	50.7	55.47
Not Stress	85.6	90.16	87.9
Average (ROC)	82	81	81

Table 4.2.2.7: Confusion matrix for User 4 using Method 2

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	8	1
Actual *Stress*	7	81

Table 4.2.2.8: Classification metrics calculated from Table 4.2.2.7

Class	Precision	Recall	F1-measure
Stress	92.04	98.7	95.2
Not Stress	88.9	53.3	66.7
Average (ROC)	93	92	91

Table 4.2.2.9: Confusion matrix for User 5 using Method 2

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	41	20
Actual *Stress*	17	18

Table 4.2.2.10: Classification metrics calculated from Table 4.2.2.9

Class	Precision	Recall	F1-measure
Stress	51.42	47.36	49.31
Not Stress	67.21	70.7	69
Average (ROC)	61	61	61

Table 4.2.2.11: Confusion matrix for User 6 using Method 2

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	478	117
Actual *Stress*	84	47

Table 4.2.2.12: Classification metrics calculated from Table 4.2.2.11

Class	Precision	Recall	F1-measure
Stress	35.87	26.65	31.8
Not Stress	80.3	85.05	82.62
Average (ROC)	70	72	71

The confusion matrices and the corresponding classification metrics *for between-subjects experiment* is shown below:

Table 4.2.2.13: Confusion matrix for the combined dataset using Method 2

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	1843	550
Actual *Stress*	392	250

Table 4.2.2.14: Classification metrics calculated from Table 4.2.2.13

Class	Precision	Recall	F1-measure
Stress	38.9	31.25	34.67
Not Stress	77	82.46	79.64
Average (ROC)	67	69	68

4.2.3 Method 3: Semi-Supervised (Co-training) with Defined Features

With explicitly defined features ($F_1 – F_{14}$), we have trained a co-training algorithm using two XGBoost classifiers in which F_1- F_7 are passed to first classifier and $F_8 – F_{14}$ are passed to the second classifier. The accuracy was examined using stratified 10-fold cross validation as the model was trained with 75% training and tested with 25% test set for 10 times.

The confusion matrices and the corresponding classification metrics for *subject-wise experiments* are shown below:

Table 4.2.3.1: Confusion matrix for User 1 using Method 3

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	633	198
Actual *Stress*	50	56

Table 4.2.3.2: Classification metrics calculated from Table 4.2.3.1

Class	Precision	Recall	F1-measure
Stress	52.82	22.04	31.11
Not Stress	76.17	92.67	83.61
Average (ROC)	70	74	69

Table 4.2.3.3: Confusion matrix for User 2 using Method 3

	Predicted *Not Stress*	Predicted *Stress*

Actual *Not Stress*	500	4
Actual *Stress*	0	193

Table 4.2.3.4: Classification metrics calculated from Table 4.2.3.3

Class	Precision	Recall	F1-measure
Stress	100	97.96	98.97
Not Stress	99.20	100	99.60
Average (ROC)	100	99	98

Table 4.2.3.5: Confusion matrix for User 3 using Method 3

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	249	22
Actual *Stress*	5	43

Table 4.2.3.6: Classification metrics calculated from Table 4.2.3.5

Class	Precision	Recall	F1-measure
Stress	89.58	66.15	76.10
Not Stress	91.9	98.03	94.85
Average (ROC)	91	92	91

Table 4.2.3.7: Confusion matrix for User 4 using Method 3

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	11	3
Actual *Stress*	4	79

Table 4.2.3.8: Classification metrics calculated from Table 4.2.3.7

Class	Precision	Recall	F1-measure
Stress	95.18	96.34	95.75
Not Stress	78.57	73.3	75.86
Average (ROC)	93	93	93

Table 4.2.3.9: Confusion matrix for User 5 using Method 3

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	48	13
Actual *Stress*	10	25

Table 4.2.3.10: Classification metrics calculated from Table 4.2.3.9

Class	Precision	Recall	F1-measure
Stress	71.42	65.78	68.49
Not Stress	78.68	82.7	80.67
Average (ROC)	78	78	78

Table 4.2.3.11: Confusion matrix for User 6 using Method 3

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	541	144
Actual *Stress*	21	20

Table 4.2.3.12: Classification metrics calculated from Table 4.2.3.11

Class	Precision	Recall	F1-measure
Stress	48.78	12.19	19.51
Not Stress	78.97	92.26	86.76
Average (ROC)	69	72	70

The confusion matrices and the corresponding classification metrics *for between-subjects experiment* is shown below:

Table 4.2.3.13: Confusion matrix for the combined dataset using Method 3

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	2065	564
Actual *Stress*	170	236

Table 4.2.3.14: Classification metrics calculated from Table 4.2.3.13

Class	Precision	Recall	F1-measure
Stress	58.12	29.5	39.13
Not Stress	78.54	92.39	84.9

Average (ROC)	73	76	73

4.2.4 Method 4: Semi-Supervised (Self-training) with Defined Features

With explicitly defined features (F_1 – F_{14}), we have trained a self-training algorithm using an XGBoost classifier for 200 iterations. The accuracy was examined using stratified 10-fold cross validation as the model was trained with 75% training and tested with 25% test set for 10 times.

The confusion matrices and the corresponding classification metrics for *subject-wise experiments* are shown below:

Table 4.2.4.1: Confusion matrix for User 1 using Method 4

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	610	162
Actual *Stress*	73	92

Table 4.2.4.2: Classification metrics calculated from Table 4.2.4.1

Class	Precision	Recall	F1-measure
Stress	55.75	36.22	43.91
Not Stress	79.01	89.31	83.84
Average (ROC)	73	75	73

Table 4.2.4.3: Confusion matrix for User 2 using Method 4

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	500	4
Actual *Stress*	0	193

Table 4.2.4.4: Classification metrics calculated from Table 4.2.4.3

Class	Precision	Recall	F1-measure
Stress	100	97.96	98.97
Not Stress	99.20	100	99.60
Average (ROC)	100	99	98

Table 4.2.4.5: Confusion matrix for User 3 using Method 4

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	240	13
Actual *Stress*	14	52

Table 4.2.4.6: Classification metrics calculated from Table 4.2.4.5

Class	Precision	Recall	F1-measure
Stress	78.78	80	79.38
Not Stress	94.86	94.48	94.67

Average (ROC)	92	92	92

Table 4.2.4.7: Confusion matrix for User 4 using Method 4

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	8	7
Actual *Stress*	7	75

Table 4.2.4.8: Classification metrics calculated from Table 4.2.4.7

Class	Precision	Recall	F1-measure
Stress	91.46	91.46	91.46
Not Stress	53.3	53.3	53.3
Average (ROC)	86	86	86

Table 4.2.4.9: Confusion matrix for User 5 using Method 4

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	51	14
Actual *Stress*	7	24

Table 4.2.4.10: Classification metrics calculated from Table 4.2.4.9

Class	Precision	Recall	F1-measure
Stress	77.41	63.15	69.56

Not Stress	78.46	87.93	82.9
Average (ROC)	73	75	73

Table 4.2.4.11: Confusion matrix for User 6 using Method 4

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	499	130
Actual *Stress*	63	34

Table 4.2.4.12: Classification metrics calculated from Table 4.2.4.11

Class	Precision	Recall	F1-measure
Stress	35.05	20.73	26.05
Not Stress	79.33	88.79	83.79
Average (ROC)	69	73	71

The confusion matrices and the corresponding classification metrics *for between-subjects experiment* is shown below:

Table 4.2.4.13: Confusion matrix for the combined dataset using Method 4

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	2002	559
Actual *Stress*	233	241

Table 4.2.4.14: Classification metrics calculated from Table 4.2.4.13

Class	Precision	Recall	F1-measure
Stress	50.84	30.125	37.83
Not Stress	78.17	89.57	83.48
Average (ROC)	71	74	71

4.2.5 Method 5: Semi-Supervised (Self-training) with Learned Features

With learned features (L_1 – L_{100}), we have trained a self-training algorithm using an XGBoost classifier for 200 iterations. The accuracy was examined using stratified 10-fold cross validation as the model was trained with 75% training and tested with 25% test set for 10 times.

The confusion matrices and the corresponding classification metrics for *subject-wise experiments* are shown below:

Table 4.2.5.1: Confusion matrix for User 1 using Method 5

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	583	166
Actual *Stress*	100	88

Table 4.2.5.2: Classification metrics calculated from Table 4.2.5.1

Class	Precision	Recall	F1-measure
Stress	46.8	34.64	39.81
Not Stress	77.83	85.35	81.42

Average (ROC)	68	71	69

Table 4.2.5.3: Confusion matrix for User 2 using Method 5

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	500	0
Actual *Stress*	0	197

Table 4.2.5.4: Classification metrics calculated from Table 4.2.5.3

Class	Precision	Recall	F1-measure
Stress	100	100	100
Not Stress	100	100	100
Average (ROC)	100	100	100

Table 4.2.5.5: Confusion matrix for User 3 using Method 5

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	218	15
Actual *Stress*	36	50

Table 4.2.5.6: Classification metrics calculated from Table 4.2.5.5

Class	Precision	Recall	F1-measure
Stress	58.13	76.92	66.22
Not Stress	93.56	85.82	89.52
Average (ROC)	79	71	74

Table 4.2.5.7: Confusion matrix for User 4 using Method 5

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	8	23
Actual *Stress*	7	59

Table 4.2.5.8: Classification metrics calculated from Table 4.2.5.7

Class	Precision	Recall	F1-measure
Stress	89.39	71.95	79.72
Not Stress	25.80	53.3	34.7
Average (ROC)	80	69	73

Table 4.2.5.9: Confusion matrix for User 5 using Method 5

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	44	22
Actual *Stress*	14	16

Table 4.2.5.10: Classification metrics calculated from Table 4.2.5.9

Class	Precision	Recall	F1-measure
Stress	53.3	42.10	47.05
Not Stress	66.7	75.86	70.96
Average (ROC)	61	62	62

Table 4.2.5.11: Confusion matrix for User 6 using Method 5

	Predicted *Not Stress*	Predicted *Stress*
Actual *Not Stress*	480	129
Actual *Stress*	82	35

Table 4.2.5.12: Classification metrics calculated from Table 4.2.5.11

Class	Precision	Recall	F1-measure
Stress	29.91	21.34	24.9
Not Stress	78.81	85.40	81.98
Average (ROC)	68	71	69

The confusion matrices and the corresponding classification metrics *for between-subjects experiment* is shown below:

Table 4.2.5.13: Confusion matrix for the combined dataset using Method 5

	Predicted *Not Stress*	Predicted *Stress*

Actual *Not Stress*	1915	495
Actual *Stress*	320	305

Table 4.2.5.14: Classification metrics calculated from Table 4.2.5.13

Class	Precision	Recall	F1-measure
Stress	48.4	38.125	42.80
Not Stress	79.46	85.68	82.45
Average (ROC)	71	73	72

CHAPTER V: RELATED WORK

Physiological sensors have been widely used to detect stress response. In particular, the importance of sensors that capture Heart Rate Variability (HRV) and Skin Conductance (SC) has been demonstrated throughout the literature [8, 9 ,10, 11, 12, 13, 16, 38].

Heart Rate variability (HRV) can be a non-invasive measurement to detect stress. Salahuddin et. Al in [14] have calculated important features of HRV that detect stress. In their experiment, they have used an ECG sensor. To induce stress , each subject was asked to perform a computer based Stroop test, in which users have to identify colors on a fast moving screen. The subjects' response times were recorded as a baseline indicator and HRV values were recorded. The results show that the most promising features of HRV that detect stress were mean HRV, mean HR, the root mean square of the successive difference between RR intervals (RMSSD), pnn50 which is the percentage of peaks which are greater than 50ms, power spectral density using Welch's method over high pass filter (HF) and low pass filter (LF) to calculate strength of the signal and strength of the noise, LF/HF which is the ratio of signal-to-noise, normalized LF (LFnu) and normalized HF (HFnu). In addition to HRV, variation in breathing is considered as an important indicator of stress and Jongyoon et. Al in [40] has classified stress using a custom made respiratory sensor, heart-rate and accelerometer. As a baseline measurement for stress, users were asked to perform a colored word test (CWT), a variation of a Stroop test in which the user has to identify the color of the word shown on the screen. They have extracted features spectral domain features: LF, HF from PSD and a total of 6 features from PDM. Using a supervised learning approach in a laboratory setting with researcher-labeled data, they were able to achieving an accuracy of 83% in a within subjects analysis and 69% a between-subjects analysis.

Skin Conductance (SC) or Electro Dermal Activity (EDA) is another indicator that detects stress; unlike HRV, EDA features are not effected with respect to body's motion [8]. Roberto et. al [41] have measured stress from the EDA response by discriminating cognitive load, which is the amount of mental effort being used in working memory. They have calculated EDA response using a custom-made sensor and as a baseline measurement for stress and to measure the cognitive load, Stroop test was conducted. Stress was classified using supervised machine learning algorithms with an accuracy of 82.8%. The significance of skin conductance (SC) in detecting stress was also described by Breslau et. Al in [12]. Their work has demonstrated that by measuring the amplitude of fast changing component (SCR) of SC and average latency of slow changing component (SCL) of SC, stress which is a physiological response often considered as a stimulus can be detected. Later, Healey and Picard in [9] have derived a measure of the startle responses (shown in Figure 3.2.1.2) from SCL and SCR components of the GSR signals. From the startle responses features such as the number of startle responses, duration of startle responses and magnitude of startle responses were derived. In addition to GSR sensor, they have used ECG (for HRV), EMG which measures the electrical activity in muscles, and extracted their features in time and spectral domains and classified mental stress by labeling the sensor data with the subjective self-reports and video recordings for ground truth. Using a supervised learning approach, they were able to classify stress with an accuracy of 97%. Even though their methods have demonstrated continuous stress monitoring in a real-world setting, the amount of sensors the user has to wear, video recordings and extensive self-reporting makes it difficult to implement their setting in everyday activities.

A stress detection system based on physiological sensors and fuzzy logic using HR, ECG, GSR, electromyography, and respiration rate sensors attached to a participant's finger, wrist and ankle detected stress with 97.4% accuracy [44]. David et. Al have also detected stress using wearable physiological sensors to capture photo plethysmograph (PPG), EDA and ECG [43]. The baseline measure for stress was calculated using a variety of public speaking and cognitive load tasks. Using a supervised machine learning approach, they have achieved an average accuracy of 79% on stress detection. Jacqueline et. Al in [42] has conducted an experiment which detects mental stress with sensors: ECG, GSR, respiration rate and EMG. The baseline measure for stress was calculated based on the user's self-reports and stress was classified using a supervised machine learning approach with an average accuracy of 79%. Continuous stress monitoring using off-the-shelf wearable devices by Sun et. Al in [8] were able to detect mental stress with the help of a SHIMMER wearable device. With just two physiological sensors: GSR and HR along with an accelerometer and Stroop tests for baseline measures, they classified stress using a supervised machine learning approach on three physical activities: sit, stand and walk with an average accuracy of 90.1%. Even though these experiments were promising, results were obtained only for experiments in a laboratory based environment with extensive labeling.

Continuous stress monitoring approaches using off-the-shelf wearable devices attached to a mobile phone are closely related to the experiments conducted in this thesis. Picard et. al [10] have measured stress with a mobile phone attached to a wearable device. Several features were extracted from sensors: GSR, HR and accelerometer and as a baseline measure for stress, extensive surveys were conducted for each participant. They have achieved an accuracy of around 75% using a supervised machine learning approach for stress classification.

Again, supervised machine learning requires substantial manual labeling in a laboratory setting. Every approach mentioned above has used supervised machine learning for stress detection.

Unsupervised learning approaches have also been used for stress detection. Continuous stress monitoring in a real-world driving task has been demonstrated by Rajiv et Al in [45]. With GSR and photo plethysmograph (PPG), which measures changes in light absorption of the skin sensors, they have classified stress using an unsupervised clustering approach and achieved an average accuracy of 71%. Ramos et. Al in [16] have implemented an approach intended to recognize stress as users perform physical tasks such as running, cycling, etc. Their approach used an unsupervised k-means clustering algorithm to label the data; with Naïve Bayes and Logistic Regression as base classifiers, they have achieved an accuracy of around 65%. Since stress is a physiological response, a raw unsupervised clustering approach often underperforms because it is difficult to differentiate a stress response from a normal body's response with no provided baseline measurement.

Others have tried to measure stress using methods other than physiological sensors. Torabi et. Al in [15] have implemented semi-supervised learning on speech data to detect stress in speakers. They have calculated applied semi-supervised learning approach on speech data and extracted features like high pitch, LFPC's, etc. and they've not used any physiological sensors.

The classification results of various approaches that detect have been shown in Table below:

Table 5.1: Comparison of other approaches for stress detection

Citation	Type of approach	Experimental Design	Accuracy on Stress detection
[40]	Supervised Learning	Physiological Sensors + Stroop Test + Laboratory Setting	69%
[41]	Supervised Learning	Physiological Sensors + Cognitive Load + Laboratory Setting	82.8%

[9]	Supervised Learning	Physiological Sensors + Self-reports + Video Recording + Real-World Setting	97%
[44]	Supervised Learning	Physiological Sensors + Label Based on Activity + Laboratory Setting	97.4%
[43]	Supervised Learning	Physiological Sensors + Cognitive Load & Public Speaking + Laboratory Setting	79%
[42]	Supervised Learning	Physiological Sensors + Self-reports + Laboratory Setting	79%
[8]	Supervised Learning	Physiological Sensors from Wearable Devices + Stroop tests + Laboratory Setting	90.1%
[10]	Supervised Learning	Physiological Sensors from Wearable Devices + Self-	75%

		reports + Real World Setting	
[45]	Unsupervised Learning	Physiological Sensors + Laboratory Setting	90.1%
[16]	Unsupervised Learning	Physiological Sensors from Wearable Devices + Real World Setting	65%
[15]	Semi Supervised Learning	Speech data + voiced parts of data as labels	72.3%

CHAPTER VI: DISCUSSION, CONCLUSION AND FUTURE WORK

Techniques in the past involved building a custom made sensor systems which is comprised of ECG, GSR and accelerometer involved participants to carry multiple, bulky sensors [8, 10]. In addition, since the sensors were custom-made, they are difficult to use widely to study participants in real-world scenarios and conditions. Most approaches to date have also required manual labeling of the entire training set [8, 9]. Stress recognition using wearable sensors and mobile phone [9] is closely related to our proposed work. Our proposed work is the first which uses a wrist worn off-the-shelf sensor with limited manual labeling and a semi supervised learning approach which detects perceived or actual stress in real time, reports detected stress to the user, and records the data for use by researchers.

In *subject-wise* analysis, we have omitted two participants from because of lack of labeled samples and conducted experiments for six users. The stress detection rate for *User 2, User 3, User 4, User 5* is high as the experiments resulted in a good number of True Positives (TP). The stress detection rate of *User 1, User 6* is and the number of False Positives (FP) is more in number. We have analyzed the data of these noisy users and observed that the data set contains sensor values as zeros for an extended duration of time. This may be because of low battery of wearable device; the user has not worn the wearable device or the wearable couldn’t capture the data. Since, the data was collected in real time settings when the users were involved in their everyday activities, we cannot point out the exact reason for this. The accuracies for stress classification across different user is shown in Figure 6.1

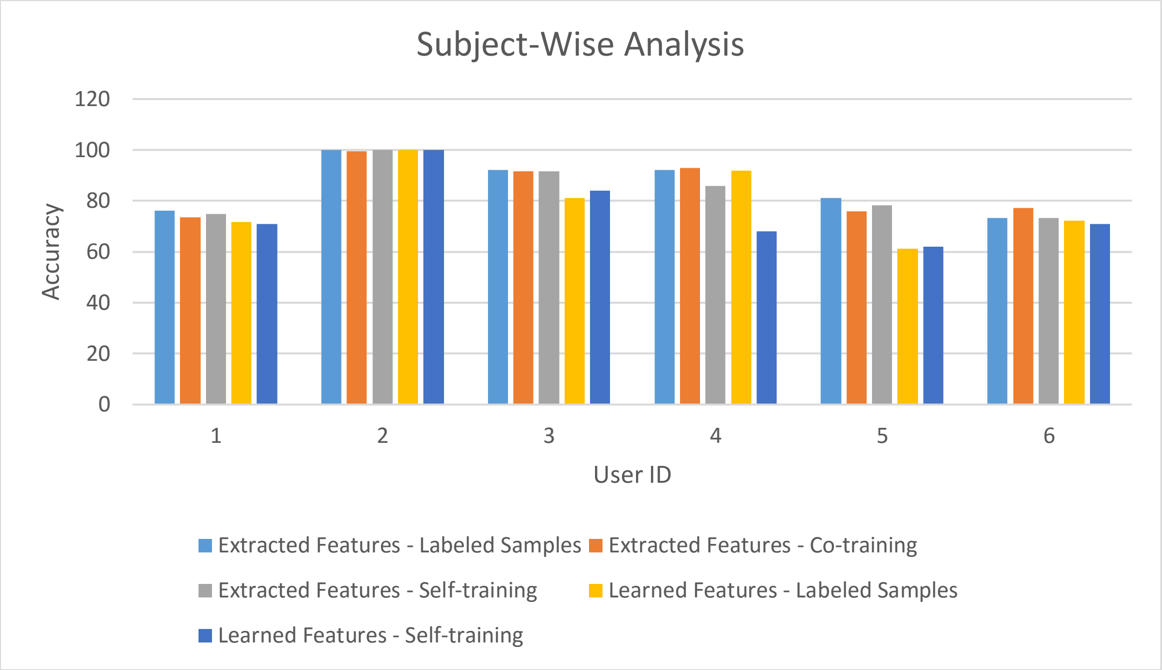

Figure 6.1: Analysis on Subject-Wise experiments

From Figure 6.1, User-2 has shown the best classification accuracy of around 100% in all of the experiments conducted. Self-training using learned features has performed poorly in the experiments performed. Co-training with explicitly defined features has performed better in all cases except for *User 1* and *User 5*. The learned features have performed better in all cases except for User-5.

To find the common physiological symptoms of stress across all users, we have conducted a between-subjects experiment grouping 8 participants together and omitting a single participant because of lack of labeled samples. The results of this experiment are shown in Figure 6.2.

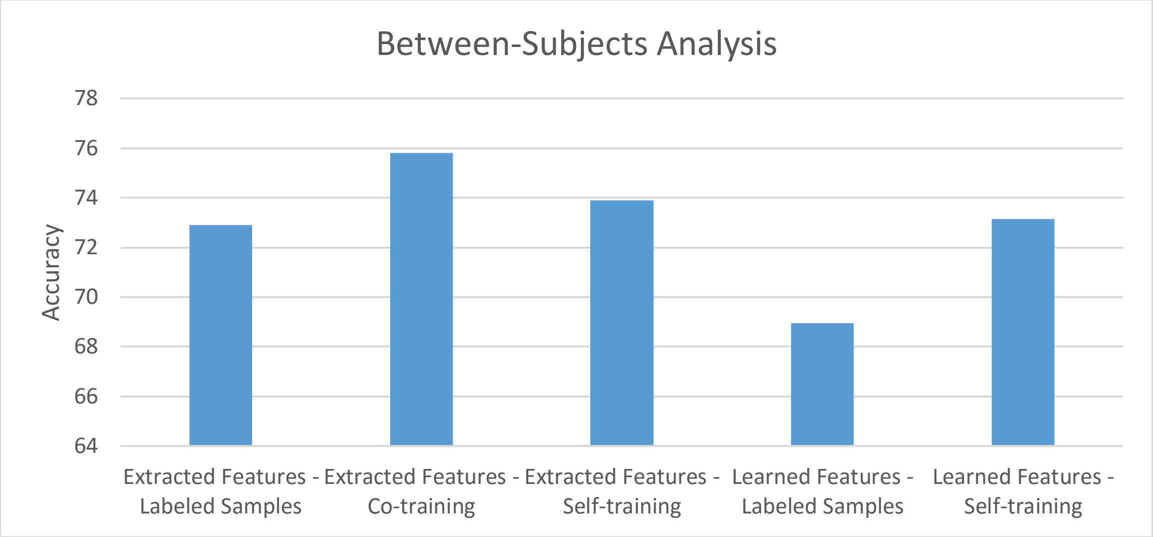

Figure 6.2: Analysis on Between-Subjects experiments

From the Figure 6.2, we can see that co-training with extracted features has performed better across all users with the highest classification accuracy of 75.7%. Since we are taking advantage of unlabeled samples in our training, co-training and self-training models used in between-subject's analysis are performing better than labeled samples using extracted features. The performance of learned features with labeled samples is poor in between-subject's analysis. However, with the unlabeled data available for training, the performance of learned features is better than the performance of learned features with labeled samples alone. Based on the between-subjects experiments and within-subjects experiment, we have observed that the physiological symptoms of stress are different for different users.

As mentioned before, we have observed the data for two *User 1* and *User 5* is noisy and if we exclude the samples of these users in the between-subjects experiment, the stress is detected with an accuracy of 85%. Even though we have achieved results that are competitive with the state of the art, our analysis is based on a small user study consisting only of eight users. Moreover, the ground truth for our semi-supervised learning approach is based on user’s reports of perceived stress, which are subjective and may not correctly align with physiological symptoms of stress. Errors in labeling can be propagated by our semi-supervised learning approach, exacerbating the problem of having noisy labels. To confirm and to address these drawbacks, we plan to run a larger user study in future, where we plan to collect data in a controlled environment by following an IRB-approved protocol for inducing stress and collecting both self-reported survey responses and researcher-labeled sensor data as a baseline measure that can be used as ground truth in our semi-supervised approach.

Initially, on the day of the user study we would ask the user to listen to soft music to make him relax and to remove any traces of possible mental stress that he is going through at that moment. Then we plan to slowly induce stress by making him perform Stroop test, Public Speaking and various Cognitive load tasks [8, 9, 40, 41, 42, 43]. We want to learn about various levels of mental stress by collecting data from this user study. Therefore, we plan to follow the tentative pipeline of calm → stress → more stress → calm → stress → more stress → some more stress → calm for each user. We also plan to include accelerometer data in the future work as [8] pointed out that as how sensor data is effected with respect to motion. We plan to experiment with implementing Contrastive Pessimistic Likelihood Estimation (CPLE) [33] as our semi-supervised learning model.

The semi supervised learning using physiological sensors from a wearable device with minimal self-reporting for ground truth on a real-world setting has achieved promising results on a short user study. As such, this approach can lead to the creation of a wearable platform for use by researchers who wish to collect data about stress from a large population of study participants over extended periods of time as well as a health and well-being application that alerts users to periods when they are experiencing physiological symptoms associated with stress.

REFERENCES

[1] T. G. Vrijkotte, L. J. van Doornen, and E. J. de Geus, “Effects of work stress on ambulatory blood pressure, heart rate, and heart rate variability.,” Hypertension, vol. 35, no. 4, pp. 880–6, Apr. 2000.

[2] R. K. Dishman, Y. Nakamura, M. E. Garcia, R. W. Thompson, A. L. Dunn, and S. N. Blair, “Heart rate variability, trait anxiety, and perceived stress among physically fit men and women,” International Journal of Psychophysiology, vol. 37, no. 2, pp. 121–133, Aug. 2000.

[3] L. Chen, J. Hoey, C. Nugent, D. Cook, and Z. Yu. Sensor-based activity recognition: A survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C, 42(6):790-808, 2012.

[4] N. Krishnan and D. Cook. Activity recognition on streaming sensor data. Pervasive and Mobile Computing, 20:138-154, 2014

[5] D. Cook, N. Krishnan, and P. Rashidi. Activity discovery and activity recognition: A new partnership. IEEE Transactions on Systems, Man, and Cybernetics, Part B, 43(3):820-828, 2013.

[6] C. Chen and D. Cook. Novelty detection in human behavior through analysis of energy utilization. Human Behavior Recognition Technologies, IGI Global, 2011.

[7] E. Kim, S. Helal, and D. Cook. Human activity recognition and pattern discovery. IEEE Pervasive Computing, 9(1):48-53, 2010.

[8] Sun, Feng-Tso, Cynthia Kuo, Heng-Tze Cheng, Senaka Buthpitiya, Patricia Collins, and Martin Griss. "Activity-aware mental stress detection using physiological sensors." In Mobile computing, applications, and services, pp. 211-230. Springer Berlin Heidelberg, 2012.

[9] Healey, J.A., Picard, R.W.: Detecting stress during real-world driving tasks us- ing physiological sensors. IEEE Transactions on Intelligent Transportation Sys- tems 6(2), 156–166 (2005)

[10] Sano, Akihide, and Rosalind W. Picard. "Stress recognition using wearable sensors and mobile phones." In Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on, pp. 671-676. IEEE, 2013.

[11] Ertin, Emre, Nathan Stohs, Santosh Kumar, Andrew Raij, Mustafa al'Absi, and Siddharth Shah. "AutoSense: unobtrusively wearable sensor suite for inferring the onset,

causality, and consequences of stress in the field." In Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems, pp. 274-287. ACM, 2011.

[12] N. Breslau, R. Kessler, and E. L. Peterson. Post-traumatic stress disorder assessment with a structured interview: reliability and concordance with a standardized clinical interview. International Journal of Methods in Psychiatric Research, 7(3):121–127, 1998

[13] Bao, L., and Intille, S. S. 2004. Activity recognition from userannotated acceleration data. In Proceceedings of the 2nd International Conference on Pervasive Computing, 1–17.

[14] Salahuddin, Lizawati, et al. "Ultra-short term analysis of heart rate variability for monitoring mental stress in mobile settings." 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE, 2007.

[15] Torabi, S., F. AlmasGanj, and A. Mohammadian. "Semi-Supervised Classification of Speaker's Psychological Stress." 2008 Cairo International Biomedical Engineering Conference. IEEE, 2008.

[16] Ramos, Julian, Jin-Hyuk Hong, and Anind K. Dey. "Stress Recognition-A Step Outside the Lab." PhyCS. 2014.

[17] M. A. Carreira-Perpi˜nan and G. E. Hinton, “On contrastive divergence learning,” in Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTATS’05), (R. G. Cowell and Z. Ghahramani, eds.), pp. 33–40, Society for Artificial Intelligence and Statistics, 2005.

[18] M.-F. Balcan and A. Blum. A discriminative model for semi-supervised learning. Journal of the ACM, 2009.

[19] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. Journal of Machine Learning Research, 7:2399–2434, November 2006.

[20] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In Proc. 18th International Conf. on Machine Learning, 2001

[21] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In COLT: Proceedings of the Workshop on Computational Learning Theory, 1998.

[22] J. Bakker, M. Pechenizkiy, and N. Sidorova. What’s your current stress level? detection of stress patterns from gsr sensor data. In Proc. of 2nd HaCDAIS Workshop @ ICDM 2011. IEEE Press, 2011

[23] J. Choi, R. Gutierrez-Osuna, Using heart rate monitors to detect mental stress, in: Proceedings of IEEE, 2009, pp. 219–223

[24] D. Yarowsky, “Unsupervised word sense disambiguation rivalling supervised methods,” in Proceedings of the Thirtythird Annual Meeting of the Association for Computational Linguistics, 1995, pp. 189–196.

[25] E. Riloff, J. Wiebe, and T. Wilson, “Learning subjective nouns using extraction pattern bootstrapping,” in Proceedings of the seventh conference on Natural language learning at HLTNAACL 2003 - Volume 4, ser. CONLL ’03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 25–32.

[26] D. Angluin and P. Laird, “Learning from noisy examples,” Machine Learning, vol. 2, pp. 343–370, 1988.

[27] R. A. McDonald, D. J. Hand, and I. A. Eckley, “An empirical comparison of three boosting algorithms on real data sets with artificial class noise,” in Proc. 4th Int. Workshop Multiple Classifier Syst., Guilford, U.K., Jun. 2003, pp. 35–44.

[28] P. Melville, N. Shah, L. Mihalkova, and R. J. Mooney, “Experiments on ensembles with missing and noisy data,” in Proc. 5th Int. Workshop Multi Classifier Syst., Cagliari, Italy, Jun. 2004, pp. 293–302.

[29] J. Friedman, T. Hastie, and R. Tibshirani, “Additive logistic regression: A statistical view of boosting,” Ann. Statist., vol. 28, no. 2, pp. 337–374, 2000

[30] G. Rätsch, B. Schölkopf, A. J. Smola, S. Mika, T. Onoda, and K.-R. Müller, “Robust ensemble learning for data mining,” in Proc. 4th Pacific-Asia Conf. Knowl. Discovery Data Mining, Current Issues New Appl., Kyoto, Japan, Apr. 2000, pp. 341–344.

[31] G. Rätsch, T. Onoda, and K.-R. Müller, “Regularizing AdaBoost,” in Advances in Neural Information Processing Systems, vol. 11. Denver, CO, USA: MIT Press, Nov./Dec. 1998, pp. 564–570.

[32] G. Rätsch, T. Onoda, and K. R. Müller, “An improvement of AdaBoost to avoid overfitting,” in Proc. 5th Int. Conf. Neural Inf. Process., Kitakyushu, Japan, Oct. 1998, pp. 506–509

[33] Friedman, Jerome H. "Stochastic gradient boosting." Computational Statistics & Data Analysis 38.4 (2002): 367-378.

[34] Chen, Tianqi, and Tong He. "xgboost: eXtreme Gradient Boosting." R package version 0.4-2 (2015).

[35] Diamantidis, N. A., D. Karlis, and Emmanouel A. Giakoumakis. "Unsupervised stratification of cross-validation for accuracy estimation." Artificial Intelligence 116.1 (2000): 1-16.

[36] Loog, Marco. "Contrastive Pessimistic Likelihood Estimation for Semi-Supervised Classification." arXiv preprint arXiv:1503.00269 (2015).

[37] coewww.rutgers.edu/classes/bme/bme305/ModelFitting/ModelFitting_HW.html

[38] dtic.mil/dtic/tr/fulltext/u2/a468462.pdf

[39] Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. "Unsupervised learning." The elements of statistical learning. Springer New York, 2009. 485-585.

[40] Choi, Jongyoon, and Ricardo Gutierrez-Osuna. "Using heart rate monitors to detect mental stress." 2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks. IEEE, 2009.

[41] Setz, Cornelia, et al. "Discriminating stress from cognitive load using a wearable EDA device." IEEE Transactions on information technology in biomedicine 14.2 (2010): 410-417.

[42] Singh, Rajiv Ranjan, Sailesh Conjeti, and Rahul Banerjee. "An approach for real-time stress-trend detection using physiological signals in wearable computing systems for automotive drivers." 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC). IEEE, 2011.

[43] Sandulescu, Virginia, et al. "Stress detection using wearable physiological sensors." International Work-Conference on the Interplay Between Natural and Artificial Computation. Springer International Publishing, 2015.

[44] de Santos Sierra, Alberto, et al. "A stress-detection system based on physiological signals and fuzzy logic." IEEE Transactions on Industrial Electronics 58.10 (2011): 4857-4865.

[45] Wijsman, Jacqueline, et al. "Towards mental stress detection using wearable physiological sensors." 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE, 2011.

[46] Murphy, Lawrence R. "Stress management in work settings: a critical review of the health effects." American Journal of Health Promotion 11.2 (1996): 112-135.

[47] Ramirez, A. J., et al. "Mental health of hospital consultants: the effects of stress and satisfaction at work." The Lancet 347.9003 (1996): 724-728.

[48] McEwen, Bruce S. "Protective and damaging effects of stress mediators." New England journal of medicine 338.3 (1998): 171-179.

[49] McEwen, Bruce S. "Central effects of stress hormones in health and disease: Understanding the protective and damaging effects of stress and stress mediators." European journal of pharmacology 583.2 (2008): 174-185.

[50] Khansari, David N., Anthony J. Murgo, and Robert E. Faith. "Effects of stress on the immune system." Immunology today 11 (1990): 170-175.

[51] Salakhutdinov, Ruslan, Andriy Mnih, and Geoffrey Hinton. "Restricted Boltzmann machines for collaborative filtering." Proceedings of the 24th international conference on Machine learning. ACM, 2007.

[52] Pencina, Michael J., Ralph B. D'Agostino, and Ramachandran S. Vasan. "Evaluating the added predictive ability of a new marker: from area under the ROC curve to reclassification and beyond." Statistics in medicine 27.2 (2008): 157-172.

APPENDIX A: STRESS QUESTIONNAIRE

The application has detected that a smoking session is occurring. Are you currently smoking?

● No

● Yes, please continue to questions

● Yes, please skip this question session

○ If yes please continue is selected:

■ How positive do you feel right now? (Scale of 1 to 7)

■ How stressed are you right now?

● Not Stressed

● Slightly Stressed

● Moderately Stressed

● Highly Stressed

● Extremely Stressed