July 2018

Baseline Assisted Classification of Heart Rate Variability

Elham Harirpoush
The University of Western Ontario

Supervisor
Lizotte Daniel
The University of Western Ontario

Graduate Program in Computer Science

A thesis submitted in partial fulfillment of the requirements for the degree in Master of Science

© Elham Harirpoush 2018

Follow this and additional works at: https://ir.lib.uwo.ca/etd

Part of the Artificial Intelligence and Robotics Commons

Recommended Citation
https://ir.lib.uwo.ca/etd/5443

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact tadam@uwo.ca.
Abstract

Recently, among various analysis methods of physiological signals, automatic analysis of Electrocardiogram (ECG) signals, especially heart rate variability (HRV) has received significant attention in the field of machine learning. Heart rate variability is an important indicator of health prediction and it is applicable to various fields of scientific research. Heart rate variability is based on measuring the differences in time between consecutive heartbeats (also known as RR interval), and the most common measuring techniques are divided into the time domain and frequency domain. In this research study, a classifier based on analysis of HRV signal is developed to classify different activities including sleep, exam, and exercise. The performance of the classifier is improved using a novel feature construction approach named as baseline assisted classifier.

ECG data are collected from 39 subjects and RR intervals are derived from ECG data using Firstbeat analysis software to compute HRV metrics. These metrics are utilized as features in a logistic regression, SVM, decision tree, random forest classifiers. Performance of all classifiers is assessed by leave one person out cross-validation technique. Features are derived by statistical time domain method from HRV segmentation during 5-minutes recording. Using a combination of 5-min segmentation feature vector and 5-min segmentation feature vector of sleep record results in a median area under the receiver operating curve (AUC) of 88% for sleep and 74% for the exam on leave one person out cross-validation test set data by SVM classifier. These results demonstrate that adding a baseline feature vector of sleep data improves the classification accuracy and classification AUC accuracy of almost all classifiers from HRV measures, and tracking of activity can be achieved by measuring the HRV signal.

Keywords: ECG signal, heart rate variability (HRV), signal processing, feature extraction, classification, machine learning.
Acknowledgments

I would like to express my gratitude to my supervisor, Dr. Dan Lizotte, for his continuous support, patience, and encouragement throughout my research study. His positive attitude and valuable advice has provided me with the motivation to make my experience very pleasant at Western University. I feel very fortunate to have him as a supervisor.

I would also like to thank Dr. Kevin Shoemaker and Rachel Knetsch in preparing the dataset, as well as their help to answer my questions regarding the dataset.

My deepest gratitude goes to my dear father and my lovely sister, Mohammad Javad Harirpoush and Andisheh Harirpoush for their never-ending support and encouragement. My special thanks also goes to my love Keyvan Mirbaha and my wonderful daughter Niusha, who has given me an endless amount of inspiration. This journey I have embarked on would be not be possible without all the love, positive energy and support from them.

Last but not the least, I give my heartfelt thanks to my friends: Annette Megerdichian and Mahnaz Rabbani who always believed in me and were willing to help during the past two years.
Contents

Abstract ii
Acknowledgments iii
List of Figures vi
List of Tables ix

1 Introduction 1
  1.1 Purpose ........................................... 1
  1.2 Contribution .................................... 2
  1.3 Outline .......................................... 2

2 Background 3
  2.1 Heart Rate ........................................ 3
  2.2 Heart Rate Variability .............................. 6
    2.2.1 Heart Rate Variability Analysis .............. 6
    2.2.2 Statistical HRV Features ...................... 7
  2.3 Machine Learning Techniques ..................... 8
    2.3.1 Supervised Learning Algorithms ............. 9
    2.3.2 Unsupervised Learning Algorithms ........... 21
    2.3.3 Evaluating Learning Algorithms ............ 23

3 Methods 27
  3.1 Heart Rate Variability (HRV) Recording ........... 27
    3.1.1 Segmentation of Heart Rate Variability ....... 27
    3.1.2 Binary Classification of Activities .......... 28
  3.2 Baseline Assisted Classification .................. 32

4 Results 34
  4.1 Baseline metrics and P-Value ..................... 34
    4.1.1 Null accuracy of HRV ........................ 34
    4.1.2 Null accuracy of BAC ........................ 34
    4.1.3 Features .................................... 35
    4.1.4 HRV P-Value ................................ 35
    4.1.5 BAC P-Value ................................ 36
  4.2 Unsupervised Dimensionality Reduction and Visualization ........... 37
List of Figures

Figure 2.1: SA node and Sympathetic and parasympathetic nervous system 4
Figure 2.2: The ECG waveform ........................................ 4
Figure 2.3: Sigmoid Function ........................................... 12
Figure 2.4: Convex and non-convex function .......................... 13
Figure 2.5: cost-function ............................................... 14
Figure 2.6: Overfitting .................................................. 14
Figure 2.7: training error versus testing error ......................... 15
Figure 2.8: overfitting in logistic regression .......................... 16
Figure 2.9: Maximum margin classification ......................... 17
Figure 2.10: Decision Tree ............................................. 20
Figure 2.11: kfold ....................................................... 24
Figure 2.12: Receiver Operating Characteristic ..................... 26
Figure 3.1: HRV Segmentation .......................................... 28
Figure 4.1: visualizing high dimensional HRV sleep data by t-SNE .. 39
Figure 4.2: visualizing high dimensional HRV exam data by t-SNE .. 39
Figure 4.3: visualizing high dimensional HRV exercise data by t-SNE .. 40
Figure 4.4: visualizing high dimensional BAC sleep data by t-SNE .. 40
Figure 4.5: visualizing high dimensional BAC exam data by t-SNE .. 41
Figure 4.6: visualizing high dimensional BAC exercise data by t-SNE .. 41
Figure 4.7: visualizing high dimensional HRV sleep data in two dimension by pca ............................................... 42
Figure 4.8: visualizing high dimensional HRV sleep data in three dimension by pca ............................................... 43
Figure 4.9: visualizing high dimensional HRV exam data in two dimension by pca ............................................... 43
Figure 4.10: visualizing high dimensional HRV exam data in three dimension by pca ............................................... 44
Figure 4.11: visualizing high dimensional HRV exercise data in two dimension by pca ............................................... 44
Figure 4.12: visualizing high dimensional HRV exercise data in three dimension by pca ............................................... 45
Figure 4.13: visualizing high dimensional BAC sleep data in two dimension by pca ............................................... 45
Figure 4.14: visualizing high dimensional BAC sleep data in three dimension by pca ............................................... 46
Figure 4.15: visualizing high dimensional BAC exam data in two dimension by pca ................................................. 46
Figure 4.16: visualizing high dimensional BAC exam data in three dimension by pca ................................................. 47
Figure 4.17: visualizing high dimensional BAC exercise data in two dimension by pca ................................................. 47
Figure 4.18: visualizing high dimensional BAC exercise data in three dimension by pca ................................................. 48
Figure 4.19: Logistic Regression normalized confusion matrix for sleep ................................................................. 50
Figure 4.20: Logistic Regression ROC curve for sleep ................................................................. 50
Figure 4.21: Logistic Regression normalized confusion matrix for sleep ................................................................. 51
Figure 4.22: Logistic Regression ROC curve for sleep ................................................................. 51
Figure 4.23: Logistic Regression normalized confusion matrix for exam ................................................................. 52
Figure 4.24: Logistic Regression ROC curve for exam ................................................................. 52
Figure 4.25: Logistic Regression normalized confusion matrix for exam ................................................................. 53
Figure 4.26: Logistic Regression ROC curve for exam ................................................................. 53
Figure 4.27: Logistic Regression normalized confusion matrix for exercise ......................................................... 54
Figure 4.28: Logistic Regression ROC curve for exercise ................................................................. 54
Figure 4.29: Logistic Regression normalized confusion matrix for exercise ......................................................... 55
Figure 4.30: Logistic Regression ROC curve for exercise ................................................................. 56
Figure 4.31: SVM normalized confusion matrix for sleep ................................................................. 57
Figure 4.32: SVM ROC curve for sleep ................................................................. 57
Figure 4.33: SVM normalized confusion matrix for sleep ................................................................. 58
Figure 4.34: SVM ROC curve for sleep ................................................................. 58
Figure 4.35: SVM normalized confusion matrix for exam ................................................................. 59
Figure 4.36: SVM ROC curve for exam ................................................................. 60
Figure 4.37: SVM normalized confusion matrix for exam ................................................................. 61
Figure 4.38: SVM ROC curve for exam ................................................................. 61
Figure 4.39: SVM normalized confusion matrix for exercise ................................................................. 62
Figure 4.40: SVM ROC curve for exercise ................................................................. 62
Figure 4.41: SVM normalized confusion matrix for exercise ................................................................. 63
Figure 4.42: SVM ROC curve for exercise ................................................................. 64
Figure 4.43: Decision Tree normalized confusion matrix for sleep ................................................................. 65
Figure 4.44: Decision Tree ROC curve for sleep ................................................................. 65
Figure 4.45: Decision Tree normalized confusion matrix for sleep ................................................................. 66
Figure 4.46: Decision Tree ROC curve for sleep ................................................................. 67
Figure 4.47: Decision Tree normalized confusion matrix for exam ................................................................. 68
Figure 4.48: Decision Tree ROC curve for exam ................................................................. 68
Figure 4.49: Decision Tree normalized confusion matrix for exam ................................................................. 69
Figure 4.50: Decision Tree ROC curve for exam ................................................................. 69
Figure 4.51: Decision Tree normalized confusion matrix for exercise ................................................................. 70
Figure 4.52: Decision Tree ROC curve for exercise ................................................................. 71
Figure 4.53: Decision Tree normalized confusion matrix for exercise ................................................................. 72
Figure 4.54: Decision Tree ROC curve for exercise ................................................................. 72
Figure 4.55: Random Forest normalized confusion matrix for sleep ................................................................. 73
Figure 4.56: Random Forest ROC curve for sleep ............................ 74
Figure 4.57: Random Forest normalized confusion matrix for sleep ...... 75
Figure 4.58: Random Forest ROC curve for sleep ............................ 75
Figure 4.59: The feature importance of the forest .............................. 76
Figure 4.60: The feature importance of the forest by label .................. 76
Figure 4.61: Random Forest normalized confusion matrix for exam ...... 77
Figure 4.62: Random Forest ROC curve for exam .............................. 77
Figure 4.63: Random Forest normalized confusion matrix for exam ...... 78
Figure 4.64: Random Forest ROC curve for exam .............................. 79
Figure 4.65: Random Forest normalized confusion matrix for exercise .. 80
Figure 4.66: Random Forest ROC curve for exercise .......................... 80
Figure 4.67: Random Forest normalized confusion matrix for exercise .. 81
Figure 4.68: Random Forest ROC curve for exercise .......................... 81
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Time domain measures of HRV</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>Confusion Matrix</td>
<td>25</td>
</tr>
<tr>
<td>3.1</td>
<td>Partial Data Set</td>
<td>28</td>
</tr>
<tr>
<td>3.2</td>
<td>Feature Augmentation</td>
<td>32</td>
</tr>
<tr>
<td>4.1</td>
<td>Features and corresponding label</td>
<td>35</td>
</tr>
<tr>
<td>4.2</td>
<td>HRV Sleep P-value</td>
<td>35</td>
</tr>
<tr>
<td>4.3</td>
<td>HRV Exam P-value</td>
<td>36</td>
</tr>
<tr>
<td>4.4</td>
<td>HRV Exercise P-value</td>
<td>36</td>
</tr>
<tr>
<td>4.5</td>
<td>BAC Sleep P-value</td>
<td>37</td>
</tr>
<tr>
<td>4.6</td>
<td>BAC Exam P-value</td>
<td>37</td>
</tr>
<tr>
<td>4.7</td>
<td>BAC Exercise P-value</td>
<td>38</td>
</tr>
<tr>
<td>4.8</td>
<td>Sleep Precision and Recall with Logistic Regression</td>
<td>49</td>
</tr>
<tr>
<td>4.9</td>
<td>Sleep Precision and Recall with Logistic Regression</td>
<td>50</td>
</tr>
<tr>
<td>4.10</td>
<td>Exam Precision and Recall with Logistic Regression</td>
<td>51</td>
</tr>
<tr>
<td>4.11</td>
<td>Exam Precision and Recall with Logistic Regression</td>
<td>52</td>
</tr>
<tr>
<td>4.12</td>
<td>Exercises Precision and Recall with Logistic Regression</td>
<td>54</td>
</tr>
<tr>
<td>4.13</td>
<td>Exercises Precision and Recall with Logistic Regression</td>
<td>55</td>
</tr>
<tr>
<td>4.14</td>
<td>Sleep Precision and Recall with SVM</td>
<td>56</td>
</tr>
<tr>
<td>4.15</td>
<td>Sleep Precision and Recall with SVM</td>
<td>58</td>
</tr>
<tr>
<td>4.16</td>
<td>Exam Precision and Recall with SVM</td>
<td>59</td>
</tr>
<tr>
<td>4.17</td>
<td>Exam Precision and Recall with SVM</td>
<td>60</td>
</tr>
<tr>
<td>4.18</td>
<td>Exam Precision and Recall with SVM</td>
<td>61</td>
</tr>
<tr>
<td>4.19</td>
<td>Exercise Precision and Recall with SVM</td>
<td>63</td>
</tr>
<tr>
<td>4.20</td>
<td>Sleep Precision and Recall with Decision Tree</td>
<td>64</td>
</tr>
<tr>
<td>4.21</td>
<td>Sleep Precision and Recall with Decision Tree</td>
<td>66</td>
</tr>
<tr>
<td>4.22</td>
<td>Exam Precision and Recall with Decision Tree</td>
<td>67</td>
</tr>
<tr>
<td>4.23</td>
<td>Sleep Precision and Recall with Decision Tree</td>
<td>68</td>
</tr>
<tr>
<td>4.24</td>
<td>Sleep Precision and Recall with Decision Tree</td>
<td>70</td>
</tr>
<tr>
<td>4.25</td>
<td>Sleep Precision and Recall with Decision Tree</td>
<td>71</td>
</tr>
<tr>
<td>4.26</td>
<td>Sleep Precision and Recall with Random Forest</td>
<td>73</td>
</tr>
<tr>
<td>4.27</td>
<td>Sleep Precision and Recall with Random Forest</td>
<td>74</td>
</tr>
<tr>
<td>4.28</td>
<td>Exam Precision and Recall with Random Forest</td>
<td>77</td>
</tr>
<tr>
<td>4.29</td>
<td>Exam Precision and Recall with Random Forest</td>
<td>78</td>
</tr>
<tr>
<td>4.30</td>
<td>Exercise Precision and Recall with Random Forest</td>
<td>79</td>
</tr>
<tr>
<td>4.31</td>
<td>Exercise Precision and Recall with Random Forest</td>
<td>81</td>
</tr>
</tbody>
</table>
Table 5.1 Comparison of the classification accuracy (%) and AUC with leave one person out on HRV and BAC datasets
Chapter 1

Introduction

1.1 Purpose

Recently among the various physiological signals, processing of Electrocardiogram (ECG) signal, especially heart rate variability (HRV) has received a significant attention, since it has been used for health prediction, and it is applicable to the variety of science from sport to physiology. Heart rate variability (HRV) is based on evaluating the differences in time between consecutive heartbeats (also known as R-R interval).

In an identical situation, HRV values of a healthy person are usually larger than someone with pathological cases. Evidence says that a stressful situation due to exercise, psychological events, or other internal or external stressful factors, results in a reduced HRV (smaller changes in the heartbeats). Meanwhile, a higher HRV (larger changes in the heartbeats) indicates that the body has a greater capacity to withstand stress, or recovers better from a past stressful situation [24]. This biomedical signal is an important health assessment parameter, for example, it has been used for detection and prediction of human stress [38], stroke, hypertension, sleep disorder and many more. In traditional medical methods, HRV signals were analyzed by specialized physicians who monitor and inspect the signals. Due to development of computer technology, the signals are now analyzed automatically by taking advantage of machine learning techniques.

The popular techniques to analyze the heart rate variability fall into three categories as:

- time domain
- spectral or frequency domain based on fast Fourier transform (FFT) [18]
- nonlinear methods consisting of Markov modeling [56], entropy-based metrics [21], probabilistic modeling [7].

In this study, HRV analysis of three main activities including sleep, exam, and exercise has been performed for 39 individuals. Seven commonly-used statistical time domain parameters [15][58][11] which are calculated from HRV segmentation during 5-minutes recording, comprising of RMSSD, SDNN, SDANN, SDANNi, SDSD,
PNN50, and AutoCorrelation are considered and described in Section 2.2.2. Machine learning techniques are applied for classifying the statistical parameters above, to predict the individual’s physical states including sleep, exam, and exercise based on an important physiological factor named HRV.

1.2 Contribution

In recent years, artificial intelligence and machine learning techniques have led to a wide range of HRV analysis results. Supervised learning algorithms were applied in many clinical studies by analyzing heart rate variability. The neural network is a well-known technique of HRV analysis [10].

In this study, the most widely-used machine learning techniques including logistic regression, support vector machine, decision tree, and random forest are discussed and examined to classify each activity based on corresponding heart rate variability signals. In order to achieve a desired level of accuracy, a novel method named as baseline assisted classification is introduced and compared for different methods.

1.3 Outline

In Chapter 2, a summary description is given on heart rate variability principles and related measuring methods. Also, applicable machine learning algorithms are explained. In Chapter 3, details of calculation based on introduced methods in Chapter 2 are discussed, and the new method of baseline assisted classification is introduced. The results and the conclusions of this study are reported in Chapters 4 and 5.
Chapter 2

Background

Basic concepts of heart rate (HR), heart rate variability (HRV), and analysis methods for HRV is provided in this chapter.

2.1 Heart Rate

Heart rate (HR) contains vital information about the level of an individual’s health. It is the number of heartbeat per minute (bpm). Contrary to common belief, a normal heartbeat rate doesn’t repeat regularly and varies from person to person.

Several studies indicate that the normal resting range of heart rate for an adult is between 50 and 90 beats per minutes [3] [30], while the American Heart Association indicates a normal resting range for an adult is between 60 and 100 beats per minute [5].

Many physiological factors influence heartbeats such as physical demands, mental/emotional stress, sitting or moving state, and many more. While running, for example, the number of beats in a minute exceeds the normal resting rate [6], whereas during sleep, based on the level of individual’s fitness it decreases to 50 beats per minutes [70]. Generally, a high HR demonstrates physical activity or exertion, whereas a low HR corresponds with rest.

Each heartbeat is as a result of electrical impulse that is supplied by Sinoatrial (SA) node which is located at the right upper chamber of the heart (Figure 2.1). The SA node is known as the pacemaker of the heart as it sends electrical impulses at a certain rate and regulates the heart rate by sympathetic nervous system (SNS) and parasympathetic nervous system (PNS) which are two main branches of Autonomic Nerve System (ANS). ANS expresses the balance between SNS and PNS [27] [53] [28] [26] and it is the part of the peripheral nervous system located in the spinal cord, brain stem, and hypothalamus (Figure 2.1). The ANS controls all non-voluntarily systems and organ in the body such as functions of the heart, circulatory system, lungs, muscular system and endocrine system [1]. The SNS increases heart rate by sending a signal to the heart during stress or a demand for increased cardiac output and the PNS slows it down at rest. Therefore, ANS regulates the states of a body in stressful or recovery situations by balancing between SNS and PNS [11] [50].
Figure 2.1: The sinoatrial node is the peacemaker of the heart, and sympathetic and parasympathetic nervous system are two branches of ANS

Typically, electrocardiography (ECG or EKG) is a simple cardiology test for measuring the heartbeats and has been used for a long time because of minimum technology requirements. The electrical activity of the heart is monitored by electrodes over time and displayed graphically [4]. These electrodes are attached to the chest (around the heart) and Limbs (arms and legs) to identify the small electrical changes during each heartbeat. A single normal cycle of the ECG which is corresponding to one heartbeat is comprised by three peaks, named P wave, QRS (a wave complex), and T as shown in Figure 2.2. The P wave represents atrial depolarisation. The QRS complex includes a Q wave, R wave and S wave and indicates ventricular depolarization. After the QRS complex, the T wave shows ventricular repolarization [62][33][66].

Figure 2.2: The figure describes two idealized heartbeats. The R–R interval shows the duration of a heartbeat. The major ECG complexes including one heartbeat are shown by P, QRS, and T.
Detection of the accurate and reliable QRS complex is an important task in ECG signal analysis since its characteristic shape is used for automatic determination of the heart rate which is an essential factor for schematic classification of cardiac cycles. Recently, several accurate algorithms have been proposed for detection of the QRS complex, for example, wavelet transforms [42], genetic algorithms [51][49][60], artificial neural networks [59][37] and many other signal analysis algorithms which is used in software applications such as smartphone applications. The time interval between successive heartbeat is measured in milliseconds (ms), and it is called the R-R interval or inter-beat interval (IBI) “since it is the time interval between R points”. A series of R-R intervals contains important information about the physiological state of an individual.
2.2 Heart Rate Variability

While heart rate defines the number of beats per minute, heart rate variability (HRV) quantifies the variation in time between consecutive heartbeats and is measured based on R-R interval time series as it is the time interval between successive R points of QRS complex of the ECG. The clinical significance of heart rate variability was first determined by Hon and Lee [31] in 1965. They noted that fetal distress was associated with reduced beat-to-beat variation before any notable change happened in the heart rate itself. In 1977 Wolf [69] was first reported the association between increased mortality and decreased HRV in patients with myocardial infarction. This was confirmed later by several studies indicating that reduced HRV is associated with higher mortality in patients with acute myocardial infarction [12] [47], and HRV is an important indicator of sudden death [63]. However, high HRV in a resting state and low HRV in an active state are assumed more favorable for a body.

In recent years, heart rate variability has gained great interest in clinical and physiological research due to ongoing interactions between scientists, physicians and physiologists [35]. It has been used as an indicator of many cardiovascular conditions, including hypertension [29], chronic heart failure, and myocardial infarction, and it is applied to predict of mortality, autonomic balance, exercise response, sleep disorder, and many more [16]. Also, Heart rate variability (HRV) is a useful metric to analyze the functionality of the autonomic nerve system. Previous studies have proposed the correlation between HRV and ANS [61]. Generally, normal HR and its variation are associated with the regulation of an autonomic nervous system (ANS) [38]. Increasing HRV reflects better adaption status, while the reduction in HRV demonstrates stress and a worse recovery status. There are benefits to understanding the state of ANS at any moment by measuring HRV. For example, researchers have claimed that the ANS has a significant role in the sudden cardiac death [65] [55]. Therefore, HRV is a useful metric for the cognition of overall health, resilience, and ability to coping with stress and managing it from all sources, and is a good indicator in the classification of stress versus relaxation, estimation of ANS balance, exercise responses, sleep disorder, assessment of mental or physical workload. In this research, heart rate variability (HRV) is recorded during three activities comprising sleep, exam, and exercise. The main purpose of this study is to improve automatic segmentation of HRV using machine learning techniques.

2.2.1 Heart Rate Variability Analysis

Heart rate variability is evaluated by a number of methods which are categorized as time-domain, spectral or frequency domain, geometric, and nonlinear methods. This study concentrates on time-domain measurements, which are used in most research studies of HRV analysis. The time-domain measures the change in normal R wave to normal R wave (NN) intervals over time and states the activity of circulation system [2].
Table 2.1: Time domain measures of HRV

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSSD</td>
<td>ms</td>
<td>The root-mean-square of successive differences</td>
</tr>
<tr>
<td>SDNN</td>
<td>ms</td>
<td>The standard deviation</td>
</tr>
<tr>
<td>SDANN</td>
<td>ms</td>
<td>The standard deviation of mean values of intervals</td>
</tr>
<tr>
<td>SDANNi</td>
<td>ms</td>
<td>The mean standard deviation of intervals</td>
</tr>
<tr>
<td>SDSD</td>
<td>ms</td>
<td>The standard deviation of differences</td>
</tr>
<tr>
<td>PNN50</td>
<td>%</td>
<td>The percentage of differences greater than 50 (ms)</td>
</tr>
<tr>
<td>AutoCorrelation</td>
<td></td>
<td>the correlation of successive intervals, called lags</td>
</tr>
</tbody>
</table>

2.2.2 Statistical HRV Features

Seven commonly used statistical time-domain parameters [15][58] which are calculated from HRV segmentation during 5-minute recording windows as proposed in previous study [46], comprised of RMSSD, SDNN, SDANN, SDANNi, SDSD, PNN50, and AutoCorrelation, are considered in this study. Each of these HRV assessment techniques is described in Table 2.1 and the detail formula of them is described in more detail by the following equations.

Suppose that \( R_i, i = 1, 2, ..., N \) be the time intervals between successive R points of a heartbeat signal. (I.e., \( R_i \) is the interval between the \( i \)th R point and the \( i + 1 \)st R point.) Each of the measures below is typically computed over a fixed-size window, e.g. 5 minutes.

1. RMSSD refers to the root mean square differences of adjacent R-R intervals in a window.

\[
RMSSD = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N-1} (R_{i+1} - R_i)^2} \tag{2.1}
\]

2. SDNN refers to the standard deviation of the R-R intervals in a window.

\[
SDNN = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (R_i - \overline{R})^2} \tag{2.2}
\]

where \( \overline{R} \) (ms) is the arithmetic mean value of the normal R-R intervals computed as follow:

\[
\overline{R} = \frac{1}{N} \sum_{i=1}^{N} R_i \tag{2.3}
\]

3. SDANN is the standard deviation of average values of consecutive R—R intervals in a window.

4. SDANNi is defined by the mean standard deviation of consecutive R—R intervals within a window.
5. SDSD refers to the standard deviation of differences between the successive R−R intervals within a window.

\[
SDSD = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (dR_i - \bar{dR})^2}
\]  

where \(dR_i = R_{i+1} - R_i\) and \(\bar{dR}\) is the mean value of all \(dR_i\)

6. PNN50 calculates percentage of the differences between consecutive R-R intervals which are greater than a 50 ms.

7. AutoCorrelation

\[
CORR(\tau) = \frac{\sum_{i=1}^{N-\tau} (R_i - \bar{R})(R_{i+\tau} - \bar{R})}{\sum_{i=1}^{N} (R_i - \bar{R})^2}
\]

where \(\tau\) is a time lag

2.3 Machine Learning Techniques

Machine learning refers to a collection of techniques that provide computers with the ability to learn automatically and discover patterns among data from experience without direct human intervention. Several kinds of machine learning algorithms are applied to find patterns among data which leads to decision making based on the example that is provided. These algorithms are classified into different categories of supervised, unsupervised learning, semi-supervised learning, and active learning. Some methods fall into more than one category.

**Supervised Learning:** Making predictions by using a labelled set of training examples.

**Unsupervised Learning:** Finding patterns in unlabeled data.

**Semi-supervised Learning:** Identifying patterns from the combination of labelled and unlabeled data.

**Active Learning:** Selecting the most informative training examples to manually label them.

In recent years, artificial intelligence and machine learning techniques have led to a wide range of HRV analysis results. Supervised learning algorithms were applied in many clinical studies by analyzing heart rate variability. Although the neural network is a well-known technique of HRV analysis [57][8][23], different classification methods have been used for classification and prediction of clinical studies by analyzing the HRV signal. Some of them are described as follow: Maryam Mohebbi et. al
[48] applied a support vector machine classifier (SVM) to predict paroxysmal atrial fibrillation based on feature extraction including nonlinear analysis, spectrum and bispectrum features of HRV. The performance of the classifier in terms of sensitivity and specificity indicates the reliable and accurate classifier. P. Karthikeyan et. al [38] purposed a new method for detection of human stress through HRV and ECG signals analysis. Different time and frequency ranges were applied and compared to extract features from HRV and ECG signals. For example, statistical time domain measures were derived from HRV signals including SDNN, SENN (standard error of RR intervals), RMSSD, SDNN, and PNN50. The K-Nearest Neighbour (KNN) and probabilistic neural network (PNN) were considered as a classifier to determine a stress versus not stress. Argyro Kampouraki et al. [36] described a method which classifies heartbeat time series to distinguish healthy subjects from those with coronary artery disease with Support Vector Machines algorithm (SVM), learning vector quantization (LVQ) neural network and backpropagation neural network. To extract features from the heart rate variability, analysis techniques and statistical methods were applied. First, RR detection algorithm developed to get the RR features. Then, statistical methods such as standard deviation (SDNN), RMSSD, SDANN, SDANNi, SDSD, PNN50, and Autocorrelation were selected as features for the Gaussian kernel-based SVM classifier. To validate the performance of the algorithm, leave-one-out cross-validation was applied for SVM. The results of each classifier demonstrated that SVM with the accuracy of 100% performs better than both neural networks with the accuracies of 92% approaches. Alan Jovic et al. [34] proposed a novel feature extraction for binary and multiple classifications of ECG signal based on HRV analysis. Features were derived from a combination of linear and non-linear methods. Different learning algorithms were performed including K-means, Bayesian Network, Artificial neural network (ANN), Decision Tree, Random Forest, and SVM. It was shown that Random Forest classifier had a better performance in comparison with the other proposed algorithms.

2.3.1 Supervised Learning Algorithms

Supervised learning methods are commonly used in machine learning, and there are many applications of this model in practice. In this model, learning algorithms make predictions based on a set of examples (in our application, a sequence of R-R intervals) which are labelled with the desired values (in our application, the activity, i.e. exercise, exam, or sleep.) A supervised learning algorithm searches for best patterns among given labelled data and then use that pattern to make predictions of the desired values for unlabeled data. Supervised learning problems are grouped into regression (continuous-labeled) and classification (discrete-labeled) problems [22].

**Regression problem** is the task of predicting continues output which could be an integer or a floating-point value. Linear regression, regression tree, and support vector regression are some examples of regression techniques [22]. Linear regression is the simplest method but it is useful for a large number of applications. In this model the learning algorithm generates a function that maps given input variables $x^i$(also known as explanatory variables or features) to desired outputs (also known as target,
or response variables) \( y^i \) by looking at several observations \( (x^i, y^i) \) (i is an index into the training set). In the field of machine learning this function called hypothesis and usually presented as:

\[
\hat{h}_\theta(x) = \theta_0 + \theta_1 x
\]  

(2.6)

The accuracy of the hypothesis can be calculated by a cost function which measures the difference between actual value and the estimated value (the prediction).

\[
J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{h}_\theta(x^i) - y^i)^2
\]  

(2.7)

Where the \( \theta \) values are parameters which is called coefficients or weights, \( x^i \) is input of \( i^{th} \) data point , and \( m \) is the number of training examples.

This function is also called squared error function, or mean squared error, and the goal is to choose \( \theta \) values such that \( \hat{h}_\theta(x) \) is close to \( y \) for the training data \( (x, y) \).

Gradient descent is one of the most common optimization algorithms that minimize an objective function \( J(\theta_0, \theta_1) \) by simultaneously updating parameters in the inverse direction of the gradient of the cost function . The learning rate \( \alpha \) defines the size of each step to find a local optimal which should be adjusted for converges of the algorithm in a reasonable time [54]. If \( \alpha \) is too small, gradient descent could be very slow and if \( \alpha \) is too large, the gradient descent may fail to converge. With a fixed value of learning rate \( \alpha \) the gradient descent can converge to the local minimum since the gradient descent automatically takes smaller steps as the magnitude of the gradient shrinks near a minimum.

Multivariate linear regression is a linear regression with multiple features. The hypothesis function fitting with these multiple features is as follows:

\[
\hat{h}_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \ldots + \theta_n x_n
\]  

(2.8)

Using the definition of matrix multiplication, the hypothesis function can be determined for one training example as:

\[
\hat{h}_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \ldots + \theta_n x_n = \theta^T x
\]  

(2.9)

Gradient Descent for multiple variables is generally the same with repeating it for \( n \) features:

repeat until convergence:

\[
\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (\hat{h}_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \quad j = 1, \ldots, n
\]  

(2.10)

Where \( x_j^{(i)} \) is the value of feature \( j \) in the \( i^{th} \) training example.

**Classification problem** is the second task in the supervised learning which maps input variables into a discrete-valued output. In the simplest setting, the output may
have only two values which is called binary classification, or in the more complicated case with more than two values it is called multi-class classification. There are many possible classification learning methods that are used for predicting a discrete output.

In recent years, machine learning classification techniques led to many approaches to HRV analysis. Supervised learning algorithms were applied in many clinical studies by analyzing heart rate variability. In this study, heartbeat signals are used to classify each activity using most widely-used classification algorithms such as logistic regression, decision tree, random forest, and support vector machine. Each of them is described in the following section.

Many machine learning algorithms make use of linearity. Linear classification algorithms assume that classes can be separated by a straight line (or its higher-dimensional analog). These include logistic regression and support vector machines.

**Baseline Linear Classifier**

Baseline linear classifier is the simplest classifier, which classifies input vector using hyper-plane (decision boundaries), the hyperplane with two-dimension is called a line, with three-dimensions called a plane. The decision boundary is created by the hypothesis function which is a linear function of the parameters $\theta$, and is calculated by the weighted sum of the input vector. The hypothesis maps all weighted sums larger than zero to class one and smaller than zero to class zero.

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \ldots + \theta_m x_m = \theta^T x_i$$  \hspace{1cm} (2.11)

$$\text{if } h_\theta(x) \geq 0 \rightarrow y = 1 \hspace{1cm} (2.12)$$

$$\text{if } h_\theta(x) < 0 \rightarrow y = 0 \hspace{1cm} (2.13)$$

However, this method does not always perform well for classification problems, as sometimes the true separator is not a linear function.

Furthermore, we may wish to estimate probabilities that convey our uncertainty in the classification. In order to address this problem the Sigmoid Function or Logistic Function is introduced, which constrains the outputs to lie between zero and one, $0 \leq h_\theta(x) \leq 1$.

**Logistic Regression Classifier**

Logistic regression is a statistical method which uses one or more explanatory variables to classify data into discrete outcomes. It estimates the probability of occurrence of an event by fitting the hypothesis $h_\theta(x)$ into a logistic function which satisfies the following inequality: $0 \leq h_\theta(x) \leq 1$

The sigmoid function is defined as follows and is represented in (Figure 2.3) :

$$g(z) = \frac{1}{1 + e^{-z}}$$  \hspace{1cm} (2.14)
Figure 2.3: The function $g(z)$, maps any real number to the $(0, 1)$ interval.

Plugging $\theta^T x$ into a logistic function results in:

$$\theta_0 + \theta_1 x_1 + \ldots + \theta_m x_m = \sum_{i=0}^{m} \theta_i x_i = \theta^T x = z \quad (2.15)$$

$$h_\theta(x) = g(\theta^T x) \quad (2.16)$$

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}} \quad (2.17)$$

In order to calculate the probability, the logistic regression applies the concept of odds-ratio, which is defined as follow:

$$\text{odds} = \frac{\Pr(y = 1|x)}{1 - \Pr(y = 1|x)} \quad (2.18)$$

Then the logistic equation is calculated by taking the natural logarithm of the odds ratio:

$$\logit(\Pr(x)) = \ln \left( \frac{\Pr(y = 1|x)}{1 - \Pr(y = 1|x)} \right) \quad (2.19)$$

The Logit of the probability is linear with respect to $x$, which means that:

$$\logit(\Pr(x)) = \theta_0 + \theta_1 x \quad (2.20)$$

So, the probability is:

$$\Pr(y = 1|x) = \frac{1}{1 + e^{-\theta^T x}} \quad (2.21)$$

Therefore the hypothesis function ($h_\theta(x)$) defines the conditional probability, which means that a certain output belongs to the class one given its features $x$. For example, $h_\theta(x) = 80\%$ means that the output has 80\% probability to the class one. The probability of class zero is the complement of the probability of class one.

$$h_\theta(x) = 1 - \Pr(y = 0|x; \theta) \quad (2.22)$$

$$\Pr(y = 1|x; \theta) + 1 - \Pr(y = 0|x; \theta) = 1 \quad (2.23)$$
In order to separate output values into class one or zero (decision boundary), the result of the hypothesis function can be interpreted as follows:

\[ h_\theta(x) \geq 0.5 \rightarrow y = 1 \quad \text{if } \theta^T x \geq 0 \]
\[ h_\theta(x) < 0.5 \rightarrow y = 0 \quad \text{if } \theta^T x < 0 \]  

(2.24)

So, the decision boundary is the property of the hypothesis and is defined by parameters of the hypothesis. The decision boundary for logistic regression is a hyperplane. The training set is used to fit the parameters \( \theta \) and finding the best parameters for logistic regression is not the same as the linear regression because there is no closed form, but the cost function is still convex and has a unique global optimum (Figure 2.4) as long as the data are not linearly separable or we use regularization (discussed below.) The log-likelihood is an objective function that is used to find the best parameters for a logistic regression model, because the logistic regression predicts conditional probabilities of the output, the likelihood function is used:

\[ l(\theta) = \prod_{i=1}^{N} \Pr(y^{(i)}|x^{(i)}, \theta) \]  

(2.25)

By taking the assumption of equation 2.21 and equation 2.25, the log-likelihood function is:

\[ \log \prod_{i=1}^{n} \Pr(Y^{(i)} = y^{(i)}|x^{(i)}) = \sum_{i=1}^{n} \left[ y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) \right] \]  

(2.26)

There are two equivalent approaches to find the best parameters of the logistic regression model, maximizing the log-likelihood (MLE) by using optimization algorithms such as gradient ascent or minimizing the logistic cost function using gradient descent. In this study, the second approach is described.

The cost function is given as follows and is visualized in Figure 2.5:

\[ \text{cost}(h_\theta(x^{(i)}), y^{(i)}) = \begin{cases} -\log(h_\theta(x^{(i)})) & \text{if } y = 1 \\ -\log(1 - h_\theta(x^{(i)})) & \text{if } y = 0 \end{cases} \]  

(2.27)

Where \( h_\theta(x) \) is a predicted value and \( y \) is an observed training value. After simplifying the equation 2.27 the cost function looks like:

\[ \text{cost}(h_\theta(x^{(i)}), y^{(i)}) = -y^{(i)} \log(h_\theta(x^{(i)})) - (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) \]  

(2.28)

The gradient descent optimization algorithm is the same as linear regression for logistic regression.
Overfitting and Underfitting in Classification

Generalization in learning algorithm refers to how well a learning algorithm learns from the training data and generalize to the new data which helps to make the prediction in future for unseen data. Overfitting and underfitting are two terminologies in machine learning model to see the performance of the model in a new data. Overfitting means that the model fits well on the training data but leads to poor model performance for new data. Typically, it happens when the model tries to learn the details of training data and separates data very precisely. Generally, large estimated coefficients are associated with overfitting (high variance). In contrast, underfitting (high bias) occurs when the model not only fails to fit the training data but also has a poor result on test data (Figure 2.6) In order to identify high bias (called underfitting) or high variance (called overfitting) of the model, training error and testing error is compared. The training error decreases when model complexity increases, however, the testing error tends to decreases first and then increases as the complexity of the model increases, the Figure 2.7 summarizes training error versus testing error.

Overfitting in Logistic Regression Classifier  In the logistic regression model, the data are said to be linearly separable when there exist coefficients θ such that
for all class one examples $\theta^T x \geq 0$ and for all class zero example $\theta^T x < 0$. In this case, both $\theta$ and the optimal decision boundary are not uniquely defined, and during the optimization process the values of $\theta$ tend toward positive or negative infinity. The Figure 2.8 shows that how the larger value of the $\theta$ leads to increasing the class probability.

Furthermore, the learner in logistic regression tries to find a decision boundary which separates data very well, and this may lead to a very complicated decision boundary that does not generalize well if the number of features is large compared to the number of training examples.

**Regularization**

Regularization is a technique which reduces the overfitting problem and can lead to better performance of learning algorithms. This technique adds a penalty on different parameters to prevent training a flexible model, therefore the model is less probable to fit the noise and the generalization capability is improved. The L1 regularization (Lasso) and the L2 regularization (Ridge) are two types of this method which avoid overfitting by penalizing large coefficients.

Lasso or L1 norm is sum of absolute value of parameters:

$$||\theta||_1 = |\theta_0| + |\theta_1| + ... + |\theta_n|$$  \hspace{1cm} (2.29)

Ridge or L2 norm is sum of squares of parameters:

$$||\theta||_2^2 = \theta_0^2 + \theta_1^2 + ... + \theta_n^2$$  \hspace{1cm} (2.30)

**Regularization in Logistic Regression** As discussed above, logistic regression is prone to overfitting with high order polynomial features and large coefficients. In order to penalize the coefficients, the cost function can be modified as follows for L2 regularized method:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{n} [y^{(i)} \log(h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))] + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2$$  \hspace{1cm} (2.31)
Figure 2.8: The class probability increases when $\theta$ is very large
Where \( \lambda \) is a tuning parameter that balances fit and magnitude of coefficients, if the value of \( \lambda \) is too large, it causes underfitting and if it is too small (zero), it results in overfitting. So, there should be an area for \( \lambda \) between zero and infinity that balances the fitting model against the magnitude of parameters \( \theta \). Choosing a \( \lambda \) can be done by cross-validation method which is explained in the evaluation section [71].

**Support Vector Machine (SVM) Classifier**

Support vector machine (SVM) is another powerful supervised learning algorithm. This algorithm is related to logistic regression. The logistic regression classifier predicts the probability of class one for input \( x \) if and only if \( \theta^T x \geq 0 \) and similarly, predicts the probability of the class zero for input \( x \) if and only if \( \theta^T x < 0 \). The larger value of \( \theta^T x \) means a higher degree of confidence to predict class one and a small value of \( \theta^T x \) corresponds to a highly confident prediction of class zero. In this model, the larger distance of training examples from the decision boundary results in the higher probability to be class one or zero, and examples near the boundary are less certain. Unlike logistic regression, SVM does not output a probability. Instead, it looks for a classifier that can separate classes by building a hyperplane (linearly separable) which maximizes the margin. The margin is the space between the hyperplane and the closest data points on each side to the decision boundary, these points are called support vectors and are shown in Figure(2.9). Essentially, the SVM avoids having training points that are uncertain, that is, points that are near the separating hyperplane. In order to classify the positive examples from negative examples with

![Figure 2.9: Maximum margin classification](image)

a hyper-plane, the decision boundary is denoted as \( \theta^T x + \theta_0 = 0 \), and the hypothesis function for a given training example \( (x^{(i)}, y^{(i)}) \) where \( y \in \{1, -1\} \) is defined as:

\[
h_\theta(x) = \begin{cases} 
1 & \text{if } \theta^T x + \theta_0 \geq 0 \\
-1 & \text{if } \theta^T x + \theta_0 < 0 
\end{cases}
\]  

(2.32)
With some calculation, the distances between the two lines called margin is measured as:

\[
\frac{2}{||\theta||}
\]  

(2.33)

The objective function looks for the decision boundary that maximizes the margin while classifying positive and negative examples correctly, which means that:

\[
\max_{\theta} \frac{2}{||\theta||} \quad \text{while} \quad y^{(i)}(\theta^T x^{(i)} + \theta_0) \geq 1 \; \forall i
\]  

(2.34)

The equation 2.34 is equivalent to the following equation which can be solved by quadratic programming:

\[
\min_{\theta} \frac{1}{2}||\theta||^2 \quad \text{while} \quad y^{(i)}(\theta^T x^{(i)} + \theta_0) \geq 1 \; \forall i
\]  

(2.35)

For mislabeled examples, the Soft Margin method is introduced which is a modified version of maximum margin. The soft margin chooses a hyperplane that split data points while maximizing the margin in misclassification problems. It uses the slack variable \(\xi\) which measures how much point \(x^{(i)}\) is on the wrong side of the margin. The objective function penalizes non-zero \(\xi\) and the optimization is regularized by the trade-off between a small error penalty and a large margin [17] [14].

\[
\min_{\theta, \xi} \left\{ \frac{1}{2}||\theta||^2 + C \sum_{i=1}^{n} \xi_i \right\} \quad \text{while} \quad y^{(i)}(\theta^T x^{(i)} + \theta_0) \geq 1 - \xi_i \; \forall i
\]  

(2.36)

The equation 2.36 and its constrains can be solved by introducing a Lagrange multipliers and transforming it into dual maximization problem of the following equations:

\[
\theta = \sum_i \alpha_i y_i x_i
\]  

(2.37)

\[
\max_{\alpha_i} \left\{ \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i \cdot x_j \right\} \quad \text{while} \quad \sum_i \alpha_i y_i = 0, \; \alpha_i \geq 0
\]  

(2.38)

where nonnegative \(\alpha_i\) are Lagrange multipliers associated with the constrains:

\[y^{(i)}(\theta^T x^{(i)} + \theta_0) \geq 1 - \xi_i\]

Note that \(\alpha_i\) are mostly zero satisfying the constrains:

\[\sum_i \alpha_i y_i = 0, \; \alpha_i \geq 0\]

The data points for which the corresponding non-zero \(\alpha_i\) are called support vectors.

However in most real problems, there is no hyperplane that classify positive examples from negative examples and the decision boundaries are nonlinear. The general idea for creating nonlinear classifiers that maximize the margin hyperplanes is using
a function $\phi : x \rightarrow \phi(x)$ that maps lower-dimensional input feature space to higher-dimensional feature space [67] which leads to kernel-based SVMs [41]. Therefore, the optimization problem for new points can be summarized as follow:

$$\min_{\theta, \xi} \left\{ \frac{1}{2} \|\theta\|^2 + C \sum_{i=1}^{n} \xi_i \right\} \quad \text{while} \quad y^{(i)}(\theta^T \phi(x^{(i)}) + \theta_0) \geq 1 - \xi_i \quad \forall i \quad (2.39)$$

Similar to the linear case the equation 2.39 is solved by Lagrange multipliers and transforming it into dual maximization problem as follow:

$$\max_{\alpha_i} \left\{ \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j K(x_i, x_j) \right\} \quad \text{while} \quad \sum_i \alpha_i y_i = 0, \alpha \geq 0 \quad (2.40)$$

where $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ which creates a nonlinear decision boundary in the original feature space. Several kernel function are described by Genton [25]. However, some typical kernel functions are Radial Basis Function (RBF), Polynomial, Hyperbolic tangent, and Sigmoid and choosing an appropriate kernel function can be performed by cross-validation method. Finally, the decision function becomes:

$$f(x) = \text{sign}\left( \sum_i (\alpha_i y_i K(x, x_i) + \theta_0) \right) \quad (2.41)$$

**Decision Tree Classifier**

One of the algorithms which are considered for classifying the data in this project is a Decision Tree (DT) method. This method is very useful when the data has a large number of features which might predict the output through complex, non-linear, interacting relationships.

A decision tree classifies examples by asking questions about the value of different features that are either true or false. The result of a question determines the next question that is asked. Questions are organized into a tree structure as it shown in Figure 2.10. If a decision tree classifier has no more questions, it produces a class label called the leaf. The quality of a question (its ability to separate positive examples from negative examples) is measured by a statistical property called information gain, which is based on a change in entropy from before to after asking the question.

**Entropy** Entropy is the amount of uncertainty that an event would occur. For example, if the probability of occurring an event is equal to one (all examples belong to the same class), then that event would have zero uncertainty or zero entropy. On the other hand, if the probability of occurring is 0.5 (each class in a binary classification problem has an equal number of examples), then we would be very uncertain about the occurrence of that event, and entropy is one. Thus, distributions with high probability events should have relatively low entropy and distributions with evenly distributed probability should have relatively high entropy. Shannon developed
Figure 2.10: Decision Tree

a model which defines that the empirical entropy of a set is a logarithmic function of the proportion of positive and negative examples and mathematically it is defined as:

\[
Entropy(S) = -(A \log A + B \log B)
\]  

(2.42)

Where \(S\) is a set of examples, \(A\) is the proportion of positive examples, and \(B\) is the proportion of negative examples.

**Information Gain** Information gain is a simple mathematical way to measure the amount of information that is gained by asking a question using a particular feature. It is the expected decrease in entropy of a training set after testing a descriptive feature. If each class label is pure (do not contain a mix labels) after asking the question, the feature would provide maximum information about the label at the leaf. The information gain based on a set of examples \((S)\) associated with asking a question about a certain feature \(X\) is defined as:

\[
Gain(S, X) = Entropy(S) - \sum_i \frac{S_i}{S} \cdot Entropy(S_i)
\]  

(2.43)

In which, \(S_i\) is a subset of \(S\) that feature \(X\) has a value of \(i\) [43]. If \(X\) is continuous, typically features of the form \(X < x\) are considered for different values of \(x\), and these are used to build the tree.

**Overfitting in Decision Tree** Decision Trees are highly prone to overfitting, even if the number of features is small. As discussed previously, overfitting happens when the training error decreases as the model complexity increases, but the true error goes down and then climbs back up. In the context of decision trees, overfitting happens
when the depth of the tree increases, which means that the decision boundary becomes more and more complex. There are two approaches to avoid overfitting in decision trees, consisting of early stopping in learning the decision tree before tree gets very complex, and pruning the tree, which means simplify the tree after terminating the learning algorithm.

The following procedures are commonly used to prevent overfitting:

- Restrict the maximum depth of the tree
- Define a minimum node size which means stop splitting when the number of data points is too small
- Calculate classification error and do not grow the tree further if the error does not decrease significantly (often measured by a hypothesis test)

The advantages of decision trees are that they are easy to interpret and visualize, also they are a useful method for identifying patterns among data that cannot be expressed by linear methods such as logistic regression and linear SVMs. However, small changes in training data can lead to decision trees with very different structures, which is sometimes undesirable [52].

**Random Forest Classifier**

Random Forest (RF) which was proposed by Breiman [13] is known as random decision forests, it is based on the collection of decision trees. This data mining algorithm addresses the problem of decision tree overfitting.

Similar to decision trees, random forest makes binary splits to create subgroups by applying simple rules repeatedly. This model first selects the subset of features randomly from a random subset of data, then find a feature among subset that has the large association with response variable (based on Gini index, which is an approximation of Entropy) to create subgroups or segmentation. After first segmentation, the new subset is selected at random and this procedure continues until the tree is completely created. The number of trees that are created is associated with the accuracy of the model. Finally, random forest classifier makes a prediction for a new data based on the averaging from the outputs of several trees which will result in significant decrease of the variance in comparison to a single Decision Tree.

### 2.3.2 Unsupervised Learning Algorithms

In contrast with supervised learning, instances are not labelled in unsupervised learning models and the learning algorithm looks for unknown but useful structure within the collection of data points. This method is used for searching for similarities, finding patterns, detecting outliers, and reducing dimension. High dimensional data refers to the data that need more than two, or three dimensions to represent. Exploring such data visually to observe the distributions of the specific variable or viewing the potential correlations among clusters as well as data points is challenging. t-SNE and
Principle Component Analysis (PCA) are two unsupervised learning techniques that are widely used for reducing the dimensionality of data prior to visualization.

**t-SNE Visualization Method**

t-SNE is a new popular method of high dimensional data visualization which embeds high dimensional data points into a two or three dimensional space by converting the similarities of data points to joint probabilities (the likelihood of occurring two events at the same time) and minimizing the relative entropy between the joint probabilities of high and low dimensional data points [39].

This method is a sort of Stochastic Neighbor Embedding with relatively better optimization and visualization that reduces the trend of data points concentration. Unlike traditional dimensionality reduction method which use linear approaches to separate dissimilar data points, the t-SNE method focuses on grouping similar data points by taking advantage of non-linear techniques [44].

**Principle Component Analysis Visualization Method**

Another dimensionality reduction method is the Principle Component Analysis (PCA) which projects the high dimensional features \( (n) \) to the lower dimensional features \( (k) \). It finds a lower dimensional linear sub-space \( (k \) vectors) while the average squared projection error of the projected data is minimized. The projection error is computed by the distances between the original data points and the projected version. In other words, it says how far on average are the training data from the projected data when using \( k \) dimensions. It should be noted that before applying PCA the mean normalization or feature scaling is needed to compute for the original feature vectors. The procedure to reduce data from \( n \)-dimensions to \( k \)-dimensions is as follows:

- Normalize the data (For each original dimension, subtract the mean, divide by standard deviation.)
- From the covariance matrix (equation 2.44 ) compute the Eigenvectors (the principal components) and Eigenvalues (length of Eigenvectors)

\[
\Sigma = \frac{1}{m} \sum_{i=1}^{m} (x^{(i)}) (x^{(i)})^T 
\]  

\[
eig(\Sigma) 
\]  

- Choose the \( k \) eigenvectors corresponding to the \( k \) largest eigenvalues
- From the \( k \) eigenvectors, construct the projection matrix
- Convert the original \( n \)-dimensional data set via the projection matrix to \( k \) dimensional subspace
2.3.3 Evaluating Learning Algorithms

As discussed in the previous section, supervised learning algorithms try to find parameters $\theta$ which minimize the training error, however, small training error does not necessarily indicate a good hypothesis because it may not fit well for new examples. Generally, training error is not a good metric to evaluate the hypothesis for unseen data. The standard way to evaluate hypothesis is to use statistical techniques (called resampling) which estimate the performance of the model on unseen data. Many different methods may be used evaluate the hypothesis; for example:

- Train and Test Set
- K-Fold Cross Validation
- Leave One Out Cross Validation
- Repeated Random Test/Train and Test set

Train and Test Set

The simplest approach for evaluating the performance of the learning algorithm is to use different training and testing data, which means that the original dataset is split into two partitions. The algorithm is learned on the first partition called training set, and predictions are made on the second partition called the testing set, and then the model is evaluated against the expected results. Although the size of training and testing depends on the size of the dataset, typically 70% of data is assigned for the training set, and 30% is assigned for testing set. This evaluation technique is very fast especially for a large dataset, however, this technique has a high variance if the size of the dataset is small. There are some metrics for interpreting the test error which will be introduced later.

K-Fold Cross Validation

Cross-Validation is a method that estimates the performance of a model with less variance if the data set is small. It splits the dataset into $k$ equal parts or folds, for example, $k = 5$ or $k = 10$. Then the algorithm is trained on $k - 1$ folds and tested using the remaining fold. This procedure is repeated $k$ times. The Figure 2.11 demonstrates the procedure of this technique. Running cross-validation results in $k$ different performance scores, which can be summarized by mean and standard deviation. This method typically has lower variance than a single train/test split, since the model is trained and assessed multiple times on various data. Choosing the size of $k$ depends on the size of the dataset, usually for large datasets $k$ values of three, five, and ten are common.

Leave One Out Cross Validation

When the size of each fold is chosen to be 1 (equivalently, $k$ is assumed to the number of observations) the method is called leave one out cross-validation. There are pros
and cons to this method. The advantage is that almost all of the data may be used for training which is important for small datasets. The drawback is that it is more expensive computationally, and may give a higher variance estimate of performance than 5- or 10-fold.

In our study, described in the next chapter, each person produces many data points. In order to evaluate the performance, we use a variant of leave one out: we hold one person out and train on the rest, and test on the held-out person.

**Metrics For Evaluation Of Hypothesis**

Measuring the error or accuracy of the classifier was explained in the previous section. However, in many real classification problems, error or accuracy is not a good measure to examine the hypothesis [20]. Machine learning algorithms work well when the classes are balanced, the number of examples in each class are equal, however, many classification problems in practice are imbalanced, there are a lot more samples on one class (majority class) versus the other (minority class) [20]. In this situation, many machine learning techniques ignore the minority class. The confusion matrix is a useful method to understand the performance of classifiers on imbalanced data, and summarizes the accuracy of the hypothesis with four performance metrics as follow:

- **True positive** - (TP): The positive label correctly predicted as positive
- **True negative** - (TN): The negative label correctly predicted as negative
- **False positive** - (FP): The negative label incorrectly predicted as positive (type 1 error)
- **False negative** - (FN): The positive label incorrectly predicted as negative (type 2 error)

These figures are demonstrated in a $2 \times 2$ matrix as shown in table 2.2:
<table>
<thead>
<tr>
<th>Predicted −</th>
<th>Predicted +</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual −</td>
<td>TN</td>
</tr>
<tr>
<td>Actual +</td>
<td>FN</td>
</tr>
</tbody>
</table>

Table 2.2: Confusion Matrix

Smaller values in the off diagonal and higher values of the diagonal in the confusion matrix indicate better model performance. A high value of false negatives means classifying positive class is problematic and high values of false positive means classifying negative class is problematic. From the confusion matrix, many indicators can be calculated that can be directly used to choose between models [9]:

- **Classification Accuracy**
  \[
  \frac{TP + TN}{TP + TN + FP + FN} \tag{2.46}
  \]

- **Misclassification Rate**
  \[
  \frac{FP + FN}{TP + TN + FP + FN} \tag{2.47}
  \]

- **Recall - True Positive Rate**
  \[
  \frac{TP}{TP + FN} \tag{2.48}
  \]

- **Specificity – True Negative Rate**
  \[
  \frac{TN}{TN + FP} \tag{2.49}
  \]

- **Precision – Positive Predicted Value**
  \[
  \frac{TP}{TP + FP} \tag{2.50}
  \]

- **F1 Score**
  \[
  \frac{2TP}{2TP + FP + FN} \tag{2.51}
  \]

Precision and recall are two important types of metrics to evaluate the performance of the classifier. The portion of the positive prediction that is actually positive is called precision, and the portion of positive instances predicted to be positive is called recall. A classifier which predicts everything as positive has a high recall and low precision since some of the true negative instances are label as a positive. For addressing this problem, the precision can be increased while decreasing the recall, which means that positive instances are predicted as positive only when very sure, which result
Figure 2.12: Higher value of AUC represents better performance of the classifier in losing many positive examples [9]. Therefore, balancing precision and recall helps to find the right model. This can be done by visualizing the precision-recall curves, which are extremely useful to understanding how well a classifier is performing [9]. An alternative way to visualize the performance of a binary classifier is called ROC curve.

**Receiver Operating Characteristic (ROC) curve** plots true positive rate against the false positive rate for all classification threshold, the metric which is considered is the Area Under the Curve which is called AUROC. The Figure 2.12 illustrates the Area Under the curve of the ROC with a different threshold. The ideal classifier would reach the upper left corner of this plot, and the higher value of AUC demonstrates the better classifier which is used as a single number to evaluate the performance of the classifier. This metric indicates the likelihood that the classifier devotes a higher predicted probability to the positive instances [20].
Chapter 3

Methods

In this chapter we present rate variability segmentation and classification methods, and a novel classification technique which increases the performance of the classifiers that are introduced in this chapter.

3.1 Heart Rate Variability (HRV) Recording

Although measuring HRV requires more accuracy than measuring heart rate, advances in computer technology and signal processing algorithms accelerated the science of understanding of the heart rate signals produced by electrocardiograph (ECG) devices. A wide range of wearable heartbeat measuring devices has been designed and developed as a simpler alternative to full ECG. In this study, a product supplied by Firstbeat (https://www.firstbeat.com) company has been used. This product assisted us in deriving heart rate of 39 individuals for three different activities consisting of sleep, exam, and exercise. The 39 participants consisted of 23 men and 16 women. Relatively similar conditions governed the three activities of all participants. For example, all of them experienced a similar night time sleeping condition and similar exercising activities. Also, the typical duration of exams was from 1 to 2 hours for all participants.

For the purpose of data acquisition, an electronic chip is attached to the chest of participants and their heart activity is recorded over the time and saved in onboard memory. Later, the logged data on the memory card is downloaded. Firstbeat analysis software is used to detect QRS complex and R-R intervals from heart signal. Finally, data sets are created by collecting the R-R intervals and corresponding activity recordings for each individual.

3.1.1 Segmentation of Heart Rate Variability

The analysis procedure used in this study is shown in Figure 3.1. Firstbeat analysis software detected QRS complexes from heart rate inputs and extracts R-R intervals in every millisecond. Then, the recording of each individual was divided into five-minute windows which is seems reasonable based on existing literature [46]. In next
step, seven statistical features including RMSSD, SDNN, SDANN, SDANNi, SDSD, PNN50, and AutoCorrelation were calculated for the five-minute window. As an example, in order to calculate (equation 2.1) statistical feature over a five minute window, the following procedure has been applied:

- The squared difference between consequent R-R intervals are calculated.
- Accumulative squared difference values are computed for every five minutes
- RMSSD is measured for each five minutes as mean squared root of accumulative values

### 3.1.2 Binary Classification of Activities

The resulting 5-minute windows were then used for binary classification of each activity by machine learning techniques. Activities are labeled as sleep or not sleep, exam or not exam, exercise or not exercise. For each individual, a unique identification number (id) is assigned. Different statistical features are computed as discussed before and recorded in a comma separated value file, as shown in the table 3.1 and appendix A. The table consists of person id, features and the corresponding window id which indicates five minute intervals.

<table>
<thead>
<tr>
<th>Person id</th>
<th>Window id</th>
<th>RMSSD</th>
<th>SDNN</th>
<th>SDANN</th>
<th>SDANNi</th>
<th>SDSD</th>
<th>PNN50</th>
<th>AutoCorrelation</th>
<th>Sleep</th>
<th>Exam</th>
<th>Exercise</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>10.3041303</td>
<td>52.1289705</td>
<td>51.3041303</td>
<td>33.90453136</td>
<td>8.032826262</td>
<td>0.004885993</td>
<td>0.550448534</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4.481905972</td>
<td>23.65754239</td>
<td>15.68495122</td>
<td>17.12199659</td>
<td>0.814988136</td>
<td>0</td>
<td>0.541422255</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>3.956323125</td>
<td>20.40099581</td>
<td>14.17543118</td>
<td>14.65812411</td>
<td>2.710140032</td>
<td>0</td>
<td>0.528641228</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>7.795485873</td>
<td>55.06107565</td>
<td>52.52526412</td>
<td>27.27875136</td>
<td>6.533259288</td>
<td>0.0016</td>
<td>0.544489988</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>12.21512314</td>
<td>34.61848864</td>
<td>19.30456854</td>
<td>28.57027353</td>
<td>9.489797485</td>
<td>0.01002004</td>
<td>0.579442004</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>14.16016993</td>
<td>37.61741279</td>
<td>14.24265899</td>
<td>26.19548583</td>
<td>11.3163363</td>
<td>0.00541250</td>
<td>0.594975608</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3.1: Partial Data Set

Before performing machine learning techniques to classify activities within each 5-minutes window, two tests were examined, consisting of P-values and null accuracy. P-values are computed to test the null hypothesis that there is no difference between the average feature values within each class, and the null accuracy is the accuracy of
the predicting the most frequent class in the testing set, which is used as a baseline for the classifier accuracy. If the null accuracy is the same as classifier accuracy the model is not useful for prediction, since the predictions made are no better than always predicting the more prevalent class.

Logistic regression, Support vector machine, decision tree, and random forest are the methods that have been applied in this experiment to classify activities. For all applied methods, hyper-parameters are tuned to prevent overfitting and find the best-fitted model. Hyper-parameters are parameters which control the model behaviour. Unlike main algorithm parameters (like $\theta$ for logistic regression), hyperparameters cannot be directly learned from training data. For example, C, kernel and gamma for support vector classifier, maximum depth for decision tree are some of the typical hyper-parameters. These parameters which are described later in this chapter must be tuned using cross-validation error as the target. The Bayesian optimization algorithm is currently a common optimization approach to find the best hyper-parameters. This method relays on the Gaussian Process (GP) to minimize and estimate the error function of hyper-parameters. However, this method is not efficient, since it requires a large number of evaluations. [45].

A novel and efficient approach is known as HORD introduced by Ilievski et. al [32]. This method employs RBF (Radial Basis Function) interpolation model as error surrogates. "HORD searches the surrogate for the most promising hyper-parameter values through dynamic coordinate search and requires many fewer function evaluations " [32]. The HORD algorithm is implemented in an open-source optimization software toolbox called pySOT which is used in this study to find the best hyper-parameters. PySOT is a toolbox based on asynchronous parallel optimization methods. The most important application of this tool is to optimize objective functions containing the high volume of integer or continuous variables and need to be processed by several processors. In order to use pySOT for a bounded set of optimization problems the following steps are applied:

- Declaring the optimization problem
- Creating pySOT objects from its module
- Running the optimization in serial

Leave one person out cross validation method is applied for tuning the hyperparameters of all classifiers, which means that the algorithm is trained on all individuals with one left out and test on the held back repeatedly. After running leave one person out cross validation, the 39 different performance scores are summarized using mean and standard deviation.

The confusion matrix function in the metric module of scikit-learn is used to describe the performance of the classifiers. Also, receiver operating characteristic (ROC) curve is plotted to visualize the performance of the learning algorithm and analyze how well the model separates two classes from each other. AUC is computed to evaluate the probability of the classifier correctly distinguishes between pairs of instances with different labels.
**Logistic Regression Estimator:** LogisticRegression classifier on the top of sklearn library estimates the coefficients $\theta$ values by using "l2" penalty (regularized classifiers) and the optimization algorithm is solver with the values of "liblinear" which is an open source library for large-scale linear classification based on coordinate descent optimization algorithm [19]

**Support Vector Machine (SVM):** The following parameters are tuned for SVM to find the optimal hyper-parameters.

- kernels range: [‘linear’, ‘rbf’, ‘sigmoid’]
- C range: [0.01, 1]
  - C controls the trade of between small error penalty and large margin
- gamma range: [0.01, 1]
  - gamma defines how far the influence of a single training example reaches and the small value means the training examples far to the decision boundary has low weights and the large value means the training examples close to the decision boundary has a lot weights.

For the parameters with string values a dictionary is defined with the key and its associated value. For example, kernels range is defined as a dictionary as follows:

$$\text{kernels} = \{0: \text{'rbf'}, 1: \text{'linear'}, 2: \text{'sigmoid'}\}$$

(3.1)

**Decision Tree:** The following parameters are tuned for Decision Tree to find the optimal hyper-parameters.

- max depth range: [3, 7]
  - max depth defines the maximum depth of the tree
- criterion: [‘gini’, ‘entropy’]
  - criterion measure the quality of a split
- max leaf node range: [5, 20]
  - by defining max leaf node the number of leaf nodes that are created can be controlled.
- min samples leaf range: [3, 7]
  - min samples leaf defines the minimum number of samples required to be at a leaf node
• min sample split range: $[4, 8]$
  
  - min sample split defines the minimum number of samples required to split
  an internal node

**Random Forest**: The following parameters are tuned for Random Forest to find the
optimal hyper-parameters.

• number of estimators range: $[1, 300]$
  
  - number of estimators defines the number of trees in the forest

• criterion range: ['gini', 'entropy']
  
  - criterion measure the quality of a split

• max depth range: $[1, 12]$
  
  - max depth defines the maximum depth of the tree

• min samples leaf range: $[1, 10]$
  
  - min samples leaf defines the minimum number of samples required to be
  at a leaf node

• min sample split range: $[2, 10]$
  
  - min sample split defines the minimum number of samples required to split
  an internal node

Scikit-Learn implementation of t-SNE and PCA are applied for dimensionality
reduction in the dataset while preserving most information. The Barnes-Hut algo-
rithm has been proposed which accelerates the computation with t-SNE [64]. Then
plotting the results of t-SNE and PCA by scatterplot of the two, three dimensions
and colouring each of sample by its corresponding label.
3.2 Baseline Assisted Classification

As described above, typically in supervised learning we want to classify an object (in our case 5-minute window) based on its features. Typically, supervised learning techniques assume that examples are independent and identically distributed (iid). This is not true in our application because examples are grouped by person in the dataset. Also, it is known that there is significant variability in ECG signal from person to person. So, the differences between windows can be caused by differences in activity but also differences in person.

As we will show in the next chapter, classifiers built using the standard supervised learning approach do not generalize very well to other people. Therefore, we propose a new method to build a classifier that is person-independent which can be generalized to new people who are not in the data set. For this purpose, a new feature vector is augmented to the data set which is the sleep windows from the same person. In other words, the new feature vector consist of two vector-valued inputs, the first input is the features for the 5-minute window that needs to be classified, and the second one is the features for a 5-minutes window of sleep from that same individual. Table 3.2 is an example of two persons with a sleep, exam, and exercise segmentation and the augmented portion derived from their sleep windows.

<table>
<thead>
<tr>
<th>Person id</th>
<th>Sequence id</th>
<th>Feature Vector I</th>
<th>Feature Vector II</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Sleep_{11}</td>
<td>Sleep_{11}</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>Sleep_{21}</td>
<td>Sleep_{11}</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>Sleep_{31}</td>
<td>Sleep_{11}</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Exam_{11}</td>
<td>Sleep_{11}</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>Exam_{21}</td>
<td>Sleep_{11}</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>Exam_{31}</td>
<td>Sleep_{11}</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Exercise_{11}</td>
<td>Sleep_{11}</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>Exercise_{21}</td>
<td>Sleep_{11}</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>Exercise_{31}</td>
<td>Sleep_{11}</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Sleep_{21}</td>
<td>Sleep_{21}</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Sleep_{22}</td>
<td>Sleep_{21}</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>Sleep_{23}</td>
<td>Sleep_{21}</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Exam_{21}</td>
<td>Sleep_{21}</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Exam_{22}</td>
<td>Sleep_{21}</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>Exam_{23}</td>
<td>Sleep_{21}</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Exercise_{21}</td>
<td>Sleep_{21}</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Exercise_{22}</td>
<td>Sleep_{21}</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>Exercise_{23}</td>
<td>Sleep_{21}</td>
</tr>
</tbody>
</table>

Table 3.2: Feature Augmentation

Then all standard learners are applied to this new augmented data set as described in previous section. Note that to use this new classifier for a new window with unknown activity, it must be provided with a window from the same person that is known to
be sleep and note that, this new feature vector has noting to do with individuals. However, sleep windows are often easy to identify manually based on heart rate and time of day so we think that is not a significant problem. In next chapter, the results from both approaches are reported.
Chapter 4

Results

In this section we present a summary of the data, followed by visualizations, followed by the results of our supervised learning algorithms. Logistic regression, Support Vector Machine (SVM), Decision Tree, and Random Forests classifiers are applied to classify each activity, based on the segmentation of heart rate variability. Our novel approach, baseline assisted classifier (BAC), is performed with all the classifiers. The results of these methods are explained in this chapter.

4.1 Baseline metrics and P-Value

A baseline metric is defined to compare against the performance of all classifiers. As discussed in Chapter 2 Section 2.3.3, the accuracy of the classifier is the percentage of correct predictions, and the null accuracy is defined as the accuracy of predicting the most frequent class. Note that because of the way the additional examples in the BAC dataset is constructed, the null accuracy is changed from the original. The null accuracy of three activities and two methods are computed as follows:

4.1.1 Null accuracy of HRV
- Null accuracy for sleep: 0.551855895197
- Null accuracy for exam: 0.573144104803
- Null accuracy for exercise: 0.875

4.1.2 Null accuracy of BAC
- Null accuracy for sleep: 0.520712528554
- Null accuracy for exam: 0.579291864347
- Null accuracy for exercise: 0.89


### 4.1.3 Features

Table 4.1 on itemized the features x1 to x14 and its respective label.

<table>
<thead>
<tr>
<th>Features</th>
<th>label of Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>RMSSD</td>
</tr>
<tr>
<td>x2</td>
<td>SDNN</td>
</tr>
<tr>
<td>x3</td>
<td>SDANN</td>
</tr>
<tr>
<td>x4</td>
<td>SDNNi</td>
</tr>
<tr>
<td>x5</td>
<td>SDSD</td>
</tr>
<tr>
<td>x6</td>
<td>PNN50</td>
</tr>
<tr>
<td>x7</td>
<td>Autocorrelation</td>
</tr>
<tr>
<td>x8</td>
<td>RMSSD-sleep</td>
</tr>
<tr>
<td>x9</td>
<td>SDNN-sleep</td>
</tr>
<tr>
<td>x10</td>
<td>SDANN-sleep</td>
</tr>
<tr>
<td>x11</td>
<td>SDNNi-sleep</td>
</tr>
<tr>
<td>x12</td>
<td>SDSD-sleep</td>
</tr>
<tr>
<td>x13</td>
<td>PNN50-sleep</td>
</tr>
<tr>
<td>x14</td>
<td>Autocorrelation-sleep</td>
</tr>
</tbody>
</table>

Table 4.1: Features and corresponding label

### 4.1.4 HRV P-Value

The p-values are computed for the hypothesis that the mean value of each feature is the same in the positive class and negative class. This gives us an idea of which features may be relevant for discrimination on their own. They are shown in table 4.2, and table 4.3, table 4.4. The result from P-values shows that all features except for SDSD (x5), are significantly associated with the class label for the sleep, and all features are significantly associated with the class label for the exam. However, the only significant feature is SDNNi (x4) for the exercise.

<table>
<thead>
<tr>
<th></th>
<th>coef</th>
<th>std err</th>
<th>z</th>
<th>p &gt;</th>
<th>z</th>
<th>0.025</th>
<th>0.975</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>4.6830</td>
<td>1.187</td>
<td>3.945</td>
<td>0.000</td>
<td>2.356</td>
<td>7.010</td>
<td></td>
</tr>
<tr>
<td>x2</td>
<td>4.0782</td>
<td>0.753</td>
<td>5.415</td>
<td>0.000</td>
<td>2.602</td>
<td>5.554</td>
<td></td>
</tr>
<tr>
<td>x3</td>
<td>-1.5302</td>
<td>0.317</td>
<td>-4.821</td>
<td>0.000</td>
<td>-2.152</td>
<td>-0.908</td>
<td></td>
</tr>
<tr>
<td>x4</td>
<td>-5.5595</td>
<td>0.615</td>
<td>-9.044</td>
<td>0.000</td>
<td>-6.764</td>
<td>-4.355</td>
<td></td>
</tr>
<tr>
<td>x5</td>
<td>1.1225</td>
<td>0.854</td>
<td>1.315</td>
<td>0.189</td>
<td>-0.551</td>
<td>2.796</td>
<td></td>
</tr>
<tr>
<td>x6</td>
<td>-1.4151</td>
<td>0.429</td>
<td>-3.301</td>
<td>0.001</td>
<td>-2.255</td>
<td>-0.575</td>
<td></td>
</tr>
<tr>
<td>x7</td>
<td>-1.3387</td>
<td>0.099</td>
<td>-13.547</td>
<td>0.000</td>
<td>-1.532</td>
<td>-1.145</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2: HRV Sleep P-value
<table>
<thead>
<tr>
<th></th>
<th>coef</th>
<th>std err</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>1.6428</td>
<td>0.643</td>
<td>2.554</td>
<td>0.011</td>
<td>0.382</td>
<td>2.903</td>
</tr>
<tr>
<td>x2</td>
<td>-3.7610</td>
<td>0.658</td>
<td>-5.714</td>
<td>0.000</td>
<td>-5.051</td>
<td>-2.471</td>
</tr>
<tr>
<td>x3</td>
<td>0.8519</td>
<td>0.264</td>
<td>3.223</td>
<td>0.001</td>
<td>0.334</td>
<td>1.370</td>
</tr>
<tr>
<td>x4</td>
<td>5.2509</td>
<td>0.531</td>
<td>9.888</td>
<td>0.000</td>
<td>4.210</td>
<td>6.292</td>
</tr>
<tr>
<td>x5</td>
<td>-3.3322</td>
<td>0.540</td>
<td>-6.166</td>
<td>0.000</td>
<td>-4.391</td>
<td>-2.273</td>
</tr>
<tr>
<td>x6</td>
<td>-0.9739</td>
<td>0.250</td>
<td>-3.889</td>
<td>0.000</td>
<td>-1.465</td>
<td>-0.483</td>
</tr>
<tr>
<td>x7</td>
<td>0.8857</td>
<td>0.071</td>
<td>12.425</td>
<td>0.000</td>
<td>0.746</td>
<td>1.025</td>
</tr>
</tbody>
</table>

Table 4.3: HRV Exam P-value

<table>
<thead>
<tr>
<th></th>
<th>coef</th>
<th>std err</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>-0.4421</td>
<td>0.540</td>
<td>-0.818</td>
<td>0.413</td>
<td>-1.501</td>
<td>0.617</td>
</tr>
<tr>
<td>x2</td>
<td>0.3529</td>
<td>0.496</td>
<td>0.711</td>
<td>0.477</td>
<td>-0.620</td>
<td>1.326</td>
</tr>
<tr>
<td>x3</td>
<td>0.2955</td>
<td>0.206</td>
<td>1.436</td>
<td>0.151</td>
<td>-0.108</td>
<td>0.699</td>
</tr>
<tr>
<td>x4</td>
<td>-0.8910</td>
<td>0.391</td>
<td>-2.276</td>
<td>0.023</td>
<td>-1.658</td>
<td>-0.124</td>
</tr>
<tr>
<td>x5</td>
<td>0.3322</td>
<td>0.431</td>
<td>0.771</td>
<td>0.441</td>
<td>-0.512</td>
<td>1.177</td>
</tr>
<tr>
<td>x6</td>
<td>0.0947</td>
<td>0.209</td>
<td>0.453</td>
<td>0.650</td>
<td>-0.315</td>
<td>0.504</td>
</tr>
<tr>
<td>x7</td>
<td>0.0654</td>
<td>0.054</td>
<td>1.213</td>
<td>0.225</td>
<td>-0.040</td>
<td>0.171</td>
</tr>
</tbody>
</table>

Table 4.4: HRV Exercise P-value

4.1.5 BAC P-Value

The p-values are computed for the baseline assisted classification and it is shown in table 4.5, table 4.6, and table 4.7. Note that because the BAC data set is much larger than the original, the p-values are much smaller. The absolute z values can be used as a rough measure of how much the feature is associated with the class.
### Table 4.5: BAC Sleep P-value

<table>
<thead>
<tr>
<th>coef</th>
<th>std err</th>
<th>z</th>
<th>p &gt;</th>
<th>z</th>
<th>[0.025</th>
<th>0.975</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>5.1978</td>
<td>0.319</td>
<td>16.281</td>
<td>0.000</td>
<td>4.572</td>
<td>5.823</td>
</tr>
<tr>
<td>x2</td>
<td>6.5430</td>
<td>0.205</td>
<td>31.938</td>
<td>0.000</td>
<td>6.141</td>
<td>6.945</td>
</tr>
<tr>
<td>x3</td>
<td>-2.2442</td>
<td>0.082</td>
<td>-27.493</td>
<td>0.000</td>
<td>-2.404</td>
<td>-2.084</td>
</tr>
<tr>
<td>x4</td>
<td>-8.1861</td>
<td>0.173</td>
<td>-47.359</td>
<td>0.000</td>
<td>-8.525</td>
<td>-7.847</td>
</tr>
<tr>
<td>x5</td>
<td>2.0137</td>
<td>0.238</td>
<td>8.450</td>
<td>0.000</td>
<td>1.547</td>
<td>2.481</td>
</tr>
<tr>
<td>x6</td>
<td>-1.0464</td>
<td>0.112</td>
<td>-9.345</td>
<td>0.000</td>
<td>-1.266</td>
<td>-0.827</td>
</tr>
<tr>
<td>x7</td>
<td>-1.3247</td>
<td>0.023</td>
<td>-57.546</td>
<td>0.000</td>
<td>-1.370</td>
<td>-1.280</td>
</tr>
<tr>
<td>x8</td>
<td>0.7851</td>
<td>0.220</td>
<td>3.564</td>
<td>0.000</td>
<td>0.353</td>
<td>1.217</td>
</tr>
<tr>
<td>x9</td>
<td>-1.9110</td>
<td>0.196</td>
<td>-9.748</td>
<td>0.000</td>
<td>-2.295</td>
<td>-1.527</td>
</tr>
<tr>
<td>x10</td>
<td>0.3885</td>
<td>0.077</td>
<td>5.025</td>
<td>0.000</td>
<td>0.237</td>
<td>0.540</td>
</tr>
<tr>
<td>x11</td>
<td>2.5268</td>
<td>0.160</td>
<td>15.780</td>
<td>0.000</td>
<td>2.213</td>
<td>2.841</td>
</tr>
<tr>
<td>x12</td>
<td>-1.3354</td>
<td>0.170</td>
<td>-7.854</td>
<td>0.000</td>
<td>-1.669</td>
<td>-1.002</td>
</tr>
<tr>
<td>x13</td>
<td>-1.6655</td>
<td>0.082</td>
<td>-20.224</td>
<td>0.000</td>
<td>-1.827</td>
<td>-1.504</td>
</tr>
<tr>
<td>x14</td>
<td>1.2057</td>
<td>0.020</td>
<td>59.552</td>
<td>0.000</td>
<td>1.166</td>
<td>1.245</td>
</tr>
</tbody>
</table>

### Table 4.6: BAC Exam P-value

<table>
<thead>
<tr>
<th>coef</th>
<th>std err</th>
<th>z</th>
<th>p &gt;</th>
<th>z</th>
<th>[0.025</th>
<th>0.975</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>1.9971</td>
<td>0.146</td>
<td>13.639</td>
<td>0.000</td>
<td>1.710</td>
<td>2.284</td>
</tr>
<tr>
<td>x2</td>
<td>-4.4325</td>
<td>0.150</td>
<td>-29.530</td>
<td>0.000</td>
<td>-4.727</td>
<td>-4.138</td>
</tr>
<tr>
<td>x3</td>
<td>0.9014</td>
<td>0.057</td>
<td>15.739</td>
<td>0.000</td>
<td>0.789</td>
<td>1.014</td>
</tr>
<tr>
<td>x4</td>
<td>5.9263</td>
<td>0.123</td>
<td>48.286</td>
<td>0.000</td>
<td>5.686</td>
<td>6.167</td>
</tr>
<tr>
<td>x5</td>
<td>-3.4905</td>
<td>0.122</td>
<td>-28.717</td>
<td>0.000</td>
<td>-3.729</td>
<td>-3.252</td>
</tr>
<tr>
<td>x6</td>
<td>-1.6221</td>
<td>0.059</td>
<td>-27.550</td>
<td>0.000</td>
<td>-1.737</td>
<td>-1.507</td>
</tr>
<tr>
<td>x7</td>
<td>0.7905</td>
<td>0.015</td>
<td>51.999</td>
<td>0.000</td>
<td>0.761</td>
<td>0.820</td>
</tr>
<tr>
<td>x8</td>
<td>-0.6883</td>
<td>0.114</td>
<td>-6.060</td>
<td>0.000</td>
<td>-0.911</td>
<td>-0.466</td>
</tr>
<tr>
<td>x9</td>
<td>0.5447</td>
<td>0.130</td>
<td>4.185</td>
<td>0.000</td>
<td>0.290</td>
<td>0.800</td>
</tr>
<tr>
<td>x10</td>
<td>-0.0171</td>
<td>0.050</td>
<td>-0.342</td>
<td>0.733</td>
<td>-0.115</td>
<td>0.081</td>
</tr>
<tr>
<td>x11</td>
<td>-0.7571</td>
<td>0.107</td>
<td>-7.065</td>
<td>0.000</td>
<td>-0.967</td>
<td>-0.547</td>
</tr>
<tr>
<td>x12</td>
<td>0.0739</td>
<td>0.095</td>
<td>0.781</td>
<td>0.435</td>
<td>-0.112</td>
<td>0.259</td>
</tr>
<tr>
<td>x13</td>
<td>1.2204</td>
<td>0.046</td>
<td>26.392</td>
<td>0.000</td>
<td>1.130</td>
<td>1.311</td>
</tr>
<tr>
<td>x14</td>
<td>-0.5384</td>
<td>0.013</td>
<td>-40.730</td>
<td>0.000</td>
<td>-0.564</td>
<td>-0.512</td>
</tr>
</tbody>
</table>

### 4.2 Unsupervised Dimensionality Reduction and Visualization

Here, we present visualizations of our data after performing dimensionality reduction. In these visualizations the algorithms do not use the training labels. However, when we plot the data points, we indicate the labels using colour. This shows whether the chosen features could "naturally" divide the data according to the labels or not.
|   | coef    | std err | z     | \(p \geq |z|\) | 0.025  | 0.975   |
|---|---------|---------|-------|----------------|--------|---------|
|x1 | -0.4426 | 0.111   | -3.984| 0.000          | -0.660 | -0.225  |
|x2 | 0.6201  | 0.106   | 5.869 | 0.000          | 0.413  | 0.827   |
|x3 | 0.1416  | 0.042   | 3.362 | 0.001          | 0.059  | 0.224   |
|x4 | -0.9941 | 0.084   | -11.809| 0.000         | -1.159 | -0.829  |
|x5 | 0.1718  | 0.088   | 1.945 | 0.052          | -0.001 | 0.345   |
|x6 | 0.2090  | 0.045   | 4.603 | 0.000          | 0.120  | 0.298   |
|x7 | 0.1205  | 0.011   | 10.606| 0.000          | 0.098  | 0.143   |
|x8 | -0.1879 | 0.091   | -2.068| 0.039          | -0.366 | -0.010  |
|x9 | 0.2981  | 0.104   | 2.871 | 0.004          | 0.095  | 0.502   |
|x10| -0.1106 | 0.040   | -2.757| 0.006          | -0.189 | -0.032  |
|x11| -0.3353 | 0.086   | -3.899| 0.000          | -0.504 | -0.167  |
|x12| 0.5492  | 0.077   | 7.162 | 0.000          | 0.399  | 0.700   |
|x13| 0.0571  | 0.036   | 1.584 | 0.113          | -0.014 | 0.128   |
|x14| -0.1885 | 0.011   | -17.688| 0.000         | -0.209 | -0.168  |

Table 4.7: BAC Exercise P-value

### 4.2.1 Results of t-SNE

For visualizing our high dimensional dataset, the t-SNE method based on the Barnes-Hut-SNE algorithm which reduces the computation from \(O(n^2)\) to \(O(n \log n)\) [64] is used for HRV and BAC dataset.

**HRV results**

- Visualizations of high dimensional HRV data along with the labels for sleep, exam, and exercise are shown using t-SNE is shown in Figures 4.1, 4.2, 4.3. In each figure, the green markers represent the positive class. Note that the labels were not used by the t-SNE algorithm.

**BAC results**

- Visualizations of high dimensional BAC data along with the labels for sleep, exam, and exercise using t-SNE are shown in Figures 4.4, 4.5, 4.6. In each figure, the green markers represent the positive class. Note that the labels were not used by the t-SNE algorithm.
Figure 4.1: HRV data visualized with t-SNE. Green markers are sleep. Although the t-SNE do not use the training labels, data are separating into two groups naturally.

Figure 4.2: HRV data visualized with t-SNE. Green markers are exam. Although the t-SNE do not use the training labels, it seems data are separating into two groups naturally.
Figure 4.3: HRV data visualized with t-SNE. Green markers are exercise. Less number of observation for exercise results in more overlaps with this activity.

Figure 4.4: BAC data visualized with t-SNE. Green markers are sleep. Although the t-SNE do not use the training labels, data are separating into two groups naturally.
Figure 4.5: BAC data visualized with t-SNE. Green markers are exam. Although the t-SNE do not use the training labels, it seems data are separating into two groups naturally.

Figure 4.6: BAC data visualized with t-SNE. Green markers are exercise. Less number of observation for exercise results in more overlaps with this activity.
4.2.2 Results of PCA

We also apply the PCA algorithm for dimensionality reduction and visualization of HRV and BAC.

HRV results

- Visualizations of high dimensional HRV data along with the labels for sleep, exam, and exercise are shown using PCA is shown in Figures 4.7, 4.8, 4.9, 4.10, 4.11, 4.12. In each figure, the green markers represent the positive class. Note that the labels were not used by the PCA algorithm.

![HRV data visualized with PCA, 2 dimensions. Green markers are sleep.](image)

**Figure 4.7**: HRV data visualized with PCA, 2 dimensions. Green markers are sleep.

BAC results

- Visualizations of high dimensional BAC data along with the labels for sleep, exam, and exercise are shown using PCA is shown in Figures 4.13, 4.14, 4.15, 4.16, 4.17, 4.18. In each figure, the green markers represent the positive class. Note that the labels were not used by the PCA algorithm.
Figure 4.8: HRV data visualized with PCA, 3 dimensions. Green markers are sleep.

Figure 4.9: HRV data visualized with PCA, 2 dimensions. Green markers are exam.
Figure 4.10: HRV data visualized with PCA, 3 dimensions. Green markers are exam.

Figure 4.11: HRV data visualized with PCA, 2 dimensions. Green markers are exercise.
Figure 4.12: HRV data visualized with PCA, 3 dimensions. Green markers are sleep.

Figure 4.13: BAC data visualized with PCA, 2 dimensions. Green markers are sleep.
Figure 4.14: BAC data visualized with PCA, 3 dimensions. Green markers are sleep.

Figure 4.15: BAC data visualized with PCA, 2 dimensions. Green markers are exam.
Figure 4.16: BAC data visualized with PCA, 3 dimensions. Green markers are exam.

Figure 4.17: BAC data visualized with PCA, 2 dimensions. Green markers are exercise.
Figure 4.18: BAC data visualized with PCA, 3 dimensions. Green markers are exercise.
4.3 Supervised Learning Results

In this section, we present the performance results for each of the algorithms, both for the original HRV data and the new BAC data. The complete results are presented here, and are summarized in Chapter 5.

4.3.1 Results of Logistic Regression

Logistic Regression learning algorithm is applied, the accuracy of the model is computed with leave one person out, the confusion matrix is plotted and Area Under the Curve (AUC) is measured for both methods. The quality and quantity of the classifier is measured with precision and recall. Note that, the support is the number of true response that falls in that class and average total value is a weighted average (support values) of precision, recall and f1-score. The following results are obtained:

Logistic Regression HRV classification of sleep versus not sleep:

Applying leave one person out cross-validation results in the accuracy of 0.833 (+/- 0.223) and AUC accuracy of 0.8272.

The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.8. Visualizing Recall and Precision are shown by two metrics as follow:

- Normalized confusion is shown in Figure 4.19
- Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.20.

Logistic Regression BAC classification of sleep versus not sleep:

Applying leave one person out cross-validation results in the accuracy of 0.883 (+/- 0.145) and AUC accuracy of 0.8604.

The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.9. Visualizing Recall and Precision are shown by two metrics as follow:

- Normalized confusion is shown in Figure 4.21.
- Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.22.

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.84</td>
<td>0.86</td>
<td>0.85</td>
<td>1011</td>
</tr>
<tr>
<td>1</td>
<td>0.82</td>
<td>0.80</td>
<td>0.81</td>
<td>821</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
<td>1832</td>
</tr>
</tbody>
</table>

Table 4.8: Sleep Precision and Recall with Logistic Regression
Logistic Regression HRV classification of exam versus not exam:

Applying leave one person out cross-validation results in the accuracy of 0.777 (+/- 0.205) and AUC accuracy of 0.7443. The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.10.
Figure 4.21: Normalized confusion matrix for sleep

Figure 4.22: Receiver Operating Characteristic (ROC) curve for sleep

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.77</td>
<td>0.80</td>
<td>0.79</td>
<td>1050</td>
</tr>
<tr>
<td>1</td>
<td>0.72</td>
<td>0.69</td>
<td>0.70</td>
<td>782</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
<td>1832</td>
</tr>
</tbody>
</table>

Table 4.10: Exam Precision and Recall with Logistic Regression

Visualizing Recall and Precision are shown by two metrics as follow:

- Normalized confusion is shown in Figure 4.23
- Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.24.
Logistic Regression BAC classification of exam versus not exam:

Applying leave one person out cross-validation results in the accuracy of 0.787 (+/- 0.188) and AUC accuracy of 0.7507.

The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.11:

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.78</td>
<td>0.81</td>
<td>0.80</td>
<td>26374</td>
</tr>
<tr>
<td>1</td>
<td>0.73</td>
<td>0.69</td>
<td>0.71</td>
<td>19154</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.76</td>
<td>0.76</td>
<td>0.76</td>
<td>45528</td>
</tr>
</tbody>
</table>

Table 4.11: Exam Precision and Recall with Logistic Regression
Visualizing Recall and Precision are shown by two metrics as follow:

- Normalized confusion is shown in Figure 4.25.

![Normalized confusion matrix](image)

Figure 4.25: Normalized confusion matrix for exam

- Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.26.

![ROC curve](image)

Figure 4.26: Receiver Operating Characteristic (ROC) curve for exam

**Logistic Regression HRV classification of exercises versus not exercises:**

Applying leave one person out cross-validation results in the accuracy of 0.865 (+/- 0.179) and AUC accuracy of 0.6179.

The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.12.

Visualizing Recall and Precision are shown by two metrics as follow:
<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.89</td>
<td>0.96</td>
<td>0.93</td>
<td>1603</td>
</tr>
<tr>
<td>1</td>
<td>0.41</td>
<td>0.19</td>
<td>0.26</td>
<td>229</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.83</td>
<td>0.86</td>
<td>0.84</td>
<td>1832</td>
</tr>
</tbody>
</table>

Table 4.12: Exercises Precision and Recall with Logistic Regression

- Normalized confusion is shown in Figure 4.27

![Normalized confusion matrix for exercise](image)

Figure 4.27: Normalized confusion matrix for exercise

- Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.28.

![Receiver Operating Characteristic (ROC) curve for exercise](image)

Figure 4.28: Receiver Operating Characteristic (ROC) curve for exercise
Logistic Regression BAC classification of exercises versus not exercises:

Applying leave one person out cross-validation results in the accuracy of 0.868 (±0.186) and AUC accuracy of 0.6626.

The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.13:

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.93</td>
<td>0.95</td>
<td>0.94</td>
<td>40975</td>
</tr>
<tr>
<td>1</td>
<td>0.47</td>
<td>0.37</td>
<td>0.42</td>
<td>4553</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.89</td>
<td>0.90</td>
<td>0.89</td>
<td>45528</td>
</tr>
</tbody>
</table>

Table 4.13: Exercises Precision and Recall with Logistic Regression

Visualizing Recall and Precision are shown by two metrics as follow:

- Normalized confusion is shown in Figure 4.29.

![Normalized confusion matrix](image)

Figure 4.29: Normalized confusion matrix for exercise

- Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.30.

4.3.2 Results of Support Vector Machine

Support Vector Machine learning algorithm is applied, the accuracy of the model is computed with leave one person out, the confusion matrix is plotted and Area Under the Curve (AUC) is measured for both methods. The quality and quantity of the classifier is measured with precision and recall. Note that, the support is the number of true response that falls in that class and average total value is a weighted average (support values) of precision, recall and f1-score. The following results are obtained:
Figure 4.30: Receiver Operating Characteristic (ROC) curve for exercise

**SVM HRV classification of sleep versus not sleep:**

Running the optimization in serial with pySOT results in good hyper-parameters as follow:

- C: 0.87753
- gamma: 0.21019
- kernel: rbf

Applying leave one person out cross-validation results in the accuracy of 0.855 (+/- 0.215) and AUC accuracy of 0.8205.

The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.14:

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.83</td>
<td>0.87</td>
<td>0.85</td>
<td>1011</td>
</tr>
<tr>
<td>1</td>
<td>0.83</td>
<td>0.77</td>
<td>0.80</td>
<td>821</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.83</td>
<td>0.83</td>
<td>0.82</td>
<td>1832</td>
</tr>
</tbody>
</table>

Table 4.14: Sleep Precision and Recall with SVM

Visualizing Recall and Precision are shown by three metrics as follow:

- Normalized confusion matrix is shown in Figure 4.31
- Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.32

**SVM BAC classification of sleep versus not sleep:**

Running the optimization in serial with pySOT results in good hyper-parameters as follow:
Applying leave one person out cross-validation results in the accuracy of 0.903 (+/-0.146) and AUC accuracy of 0.8893.

The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.15:

Visualizing Recall and Precision are shown by three metrics as follow:

- Normalized confusion matrix is shown in Figure 4.33
- Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.34
Table 4.15: Sleep Precision and Recall with SVM

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.92</td>
<td>0.79</td>
<td>0.85</td>
<td>23707</td>
</tr>
<tr>
<td>1</td>
<td>0.81</td>
<td>0.93</td>
<td>0.86</td>
<td>21821</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.87</td>
<td>0.86</td>
<td>0.86</td>
<td>45528</td>
</tr>
</tbody>
</table>

Figure 4.33: Normalized confusion matrix for sleep

Figure 4.34: Receiver Operating Characteristic (ROC) curve for sleep

**SVM HRV classification of exam versus not exam:**

Running the optimization in serial with pySOT results in good hyper-parameters as follow:

- C: 0.92163
• gamma: 0.74097

• kernel: linear

Applying leave one person out cross-validation results in the accuracy of 0.764 (+/- 0.033) and AUC accuracy of 0.741.
The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.16:

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.78</td>
<td>0.79</td>
<td>0.78</td>
<td>1050</td>
</tr>
<tr>
<td>1</td>
<td>0.71</td>
<td>0.69</td>
<td>0.70</td>
<td>782</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
<td>1832</td>
</tr>
</tbody>
</table>

Table 4.16: Exam Precision and Recall with SVM

Visualizing Recall and Precision are shown by three metrics as follow:

• Normalized confusion matrix is shown in Figure 4.35

![Normalized confusion matrix for exam](image)

Figure 4.35: Normalized confusion matrix for exam

• Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.36

SVM BAC classification of exam versus not exam:

Running the optimization in serial with pySOT results in good hyper-parameters as follow:

• C: 0.15142857
Figure 4.36: Receiver Operating Characteristic (ROC) curve for exam

- gamma: 0.43428571
- kernel: linear

Applying leave one person out cross-validation results in the accuracy of 0.793 (+/-0.194) and AUC accuracy of 0.7619. The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.17:

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.79</td>
<td>0.82</td>
<td>0.81</td>
<td>26374</td>
</tr>
<tr>
<td>1</td>
<td>0.74</td>
<td>0.70</td>
<td>0.72</td>
<td>19154</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
<td>45528</td>
</tr>
</tbody>
</table>

Table 4.17: Exam Precision and Recall with SVM

Visualizing Recall and Precision are shown by three metrics as follow:
- Normalized confusion matrix is shown in Figure 4.37
- Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.38

**SVM HRV classification of exercise versus not exercise:**

Running the optimization in serial with pySOT results in good hyper-parameters as follow:
- C: 0.47289701
- gamma: 0.74373271
- kernel: rbf
Applying leave one person out cross-validation results in the accuracy of 0.901 (+/- 0.155) and AUC accuracy of 0.6313.
The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.18:

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.91</td>
<td>0.97</td>
<td>0.94</td>
<td>1603</td>
</tr>
<tr>
<td>1</td>
<td>0.62</td>
<td>0.29</td>
<td>0.39</td>
<td>229</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.87</td>
<td>0.89</td>
<td>0.87</td>
<td>1832</td>
</tr>
</tbody>
</table>

Table 4.18: Exam Precision and Recall with SVM

Visualizing Recall and Precision are shown by three metrics as follow:

- Normalized confusion matrix is shown in Figure 4.39
Figure 4.39: Normalized confusion matrix for exercise

- Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.40

Figure 4.40: Receiver Operating Characteristic (ROC) curve for exercise

**SVM BAC classification of exercise versus not exercise:**

Running the optimization in serial with pySOT results in good hyper-parameters as follow:

- C: 0.71714286
- gamma: 0.85857143
- kernel: rbf
Applying leave one person out cross-validation results in the accuracy of 0.856 (+/-0.170) and AUC accuracy of 0.5738. The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.19:

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.91</td>
<td>0.90</td>
<td>0.91</td>
<td>40975</td>
</tr>
<tr>
<td>1</td>
<td>0.22</td>
<td>0.24</td>
<td>0.23</td>
<td>4553</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.85</td>
<td>0.84</td>
<td>0.84</td>
<td>45528</td>
</tr>
</tbody>
</table>

Table 4.19: Exercise Precision and Recall with SVM

Visualizing Recall and Precision are shown by three metrics as follow:

• Normalized confusion matrix is shown in Figure 4.41

![Normalized confusion matrix](image)

Figure 4.41: Normalized confusion matrix for exercise

• Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.42

4.3.3 Results of Decision Tree

Decision Tree learning algorithm is applied, the accuracy of the model is computed with leave one person out, the confusion matrix is plotted and Area Under the Curve is measured for both methods. The quality and quantity of the classifier is measured with precision and recall. Note that, the support is the number of true response that falls in that class and average total value is a weighted average (support values) of precision, recall and f1-score. The following results are obtained:
Figure 4.42: Receiver Operating Characteristic (ROC) curve for exercise

**Decision Tree HRV classification of sleep versus not sleep:**

Running the optimization in serial with pySOT results in good hyper-parameters as follow:

- criterion: gini
- max depth: 6
- min samples leaf: 5
- min sample split: 6
- max leaf nodes: 13

Applying leave one person out cross-validation results in the accuracy of 0.807 (+/- 0.241) and AUC accuracy of 0.8054.

The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.20:

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.81</td>
<td>0.86</td>
<td>0.83</td>
<td>1011</td>
</tr>
<tr>
<td>1</td>
<td>0.81</td>
<td>0.75</td>
<td>0.78</td>
<td>821</td>
</tr>
<tr>
<td><strong>avg / total</strong></td>
<td><strong>0.81</strong></td>
<td><strong>0.81</strong></td>
<td><strong>0.81</strong></td>
<td><strong>1832</strong></td>
</tr>
</tbody>
</table>

Table 4.20: Sleep Precision and Recall with Decision Tree

Visualizing Recall and Precision are shown by three metrics as follow:

- Normalized confusion is shown in Figure 4.43
- Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.44
Decision Tree BAC classification of sleep versus not sleep:

Running the optimization in serial with pySOT results in good hyper-parameters as follow:

- criterion: gini
- max depth: 5
- min samples leaf: 7
- min sample split: 4
- max leaf nodes: 12

Applying leave one person out cross-validation results in the accuracy of 0.895 (+/-0.174) and AUC accuracy of 0.861.
The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.21:

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.92</td>
<td>0.79</td>
<td>0.85</td>
<td>23707</td>
</tr>
<tr>
<td>1</td>
<td>0.81</td>
<td>0.93</td>
<td>0.86</td>
<td>21821</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.87</td>
<td>0.86</td>
<td>0.86</td>
<td>45528</td>
</tr>
</tbody>
</table>

Table 4.21: Sleep Precision and Recall with Decision Tree

Visualizing Recall and Precision are shown by three metrics as follow:

- Normalized confusion is shown in Figure 4.45

![Normalized confusion matrix for sleep](image)

- Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.46

Decision Tree HRV classification of exam versus not exam:

Running the optimization in serial with pySOT results in good hyper-parameters as follow:

- criterion: gini
- max depth: 5
- min samples leaf: 6
- min sample split: 5
Applying leave one person out cross-validation results in the accuracy of 0.691 (+/- 0.277) and AUC accuracy of 0.6608.

The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.22:

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.71</td>
<td>0.72</td>
<td>0.71</td>
<td>1050</td>
</tr>
<tr>
<td>1</td>
<td>0.62</td>
<td>0.60</td>
<td>0.61</td>
<td>782</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.67</td>
<td>0.67</td>
<td>0.67</td>
<td>1832</td>
</tr>
</tbody>
</table>

Table 4.22: Exam Precision and Recall with Decision Tree

Visualizing Recall and Precision are shown by three metrics as follow:

- Normalized confusion is shown in Figure 4.47
- Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.48

**Decision Tree BAC classification of exam versus not exam:**

Running the optimization in serial with pySOT results in good hyper-parameters as follow:

- criterion: gini
- max depth: 6
- min samples leaf: 5
- min sample split: 8
Figure 4.47: Normalized confusion matrix for exam

Figure 4.48: Receiver Operating Characteristic (ROC) curve for exam

- max leaf nodes: 15

Applying leave one person out cross-validation results in the accuracy of 0.758 (+/-0.216) and AUC accuracy of 0.697.

The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.23:

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.72</td>
<td>0.83</td>
<td>0.77</td>
<td>26374</td>
</tr>
<tr>
<td>1</td>
<td>0.71</td>
<td>0.56</td>
<td>0.63</td>
<td>19154</td>
</tr>
<tr>
<td><strong>avg / total</strong></td>
<td><strong>0.72</strong></td>
<td><strong>0.72</strong></td>
<td><strong>0.71</strong></td>
<td><strong>45528</strong></td>
</tr>
</tbody>
</table>

Table 4.23: Sleep Precision and Recall with Decision Tree
Visualizing Recall and Precision are shown by three metrics as follow:

- Normalized confusion is shown in Figure 4.49

![Normalized confusion matrix for exam](image)

**Figure 4.49: Normalized confusion matrix for exam**

- Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.50

![Receiver Operating Characteristic (ROC) curve for exam](image)

**Figure 4.50: Receiver Operating Characteristic (ROC) curve for exam**

**Decision Tree HRV classification of exercise versus not exercise:**

Running the optimization in serial with pySOT results in good hyper-parameters as follow:

- criterion: entropy
- max depth: 4
- min samples leaf: 6
- min sample split: 7
- max leaf nodes: 10

Applying leave one person out cross-validation results in the accuracy of 0.890 (+/- 0.173) and AUC accuracy of 0.6928.

The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.24:

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.92</td>
<td>0.94</td>
<td>0.93</td>
<td>1603</td>
</tr>
<tr>
<td>1</td>
<td>0.53</td>
<td>0.44</td>
<td>0.48</td>
<td>229</td>
</tr>
<tr>
<td><strong>avg / total</strong></td>
<td><strong>0.87</strong></td>
<td><strong>0.88</strong></td>
<td><strong>0.88</strong></td>
<td><strong>1832</strong></td>
</tr>
</tbody>
</table>

Table 4.24: Sleep Precision and Recall with Decision Tree

Visualizing Recall and Precision are shown by three metrics as follow:

- Normalized confusion is shown in Figure 4.51

![Decision Tree Normalized confusion matrix](image)

Figure 4.51: Normalized confusion matrix for exercise

- Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.52
Figure 4.52: Receiver Operating Characteristic (ROC) curve for exercise

**Decision Tree BAC classification of exercise versus not exercise:**

Running the optimization in serial with pySOT results in good hyper-parameters as follow:

- criterion: gini
- max depth: 7
- min samples leaf: 5
- min sample split: 7
- max leaf nodes: 20

Applying leave one person out cross-validation results in the accuracy of 0.878 (+/-0.164) and AUC accuracy of 0.6513.

The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.25:

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.93</td>
<td>0.95</td>
<td>0.94</td>
<td>40975</td>
</tr>
<tr>
<td>1</td>
<td>0.43</td>
<td>0.36</td>
<td>0.39</td>
<td>4553</td>
</tr>
<tr>
<td><strong>avg / total</strong></td>
<td><strong>0.88</strong></td>
<td><strong>0.89</strong></td>
<td><strong>0.88</strong></td>
<td><strong>45528</strong></td>
</tr>
</tbody>
</table>

Table 4.25: Sleep Precision and Recall with Decision Tree

Visualizing Recall and Precision are shown by three metrics as follow:

- Normalized confusion is shown in Figure 4.53
- Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.54
4.3.4 Results of Random Forest

Random Forest learning algorithm is applied, the accuracy of the model is computed with leave one person out, the confusion matrix is plotted and Area Under the Curve is measured for both methods. The quality and quantity of the classifier is measured with precision and recall. Note that, the support is the number of true response that falls in that class and average total value is a weighted average (support values) of precision, recall and f1-score. The following results are obtained:

Random Forest HRV classification of sleep versus not sleep:

Running the optimization in serial with pySOT results in good hyper-parameters as follow:

- number of estimators: 189
• criterion: gini
• max depth: 5
• min samples leaf: 5
• min sample split: 5

Applying leave one person out cross-validation results in the accuracy of 0.822 (+/- 0.227) and AUC accuracy of 0.8248.

The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.26:

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.83</td>
<td>0.86</td>
<td>0.85</td>
<td>1011</td>
</tr>
<tr>
<td>1</td>
<td>0.82</td>
<td>0.79</td>
<td>0.80</td>
<td>821</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
<td>1832</td>
</tr>
</tbody>
</table>

Table 4.26: Sleep Precision and Recall with Random Forest

Visualizing Recall and Precision are shown by three metrics as follow:

• Normalized confusion is shown in Figure 4.55.

![Normalized confusion matrix for sleep](image)

Figure 4.55: Normalized confusion matrix for sleep

• Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.56.
Random Forest BAC classification of sleep versus not sleep:

Running the optimization in serial with pySOT results in good hyper-parameters as follow:

- number of estimators: 47
- criterion: gini
- max depth: 12
- min samples leaf: 5
- min sample split: 2

Applying leave one person out cross-validation results in the accuracy of 0.886 (+/- 0.181) and AUC accuracy of 0.8455.

The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.27:

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.87</td>
<td>0.83</td>
<td>0.85</td>
<td>23707</td>
</tr>
<tr>
<td>1</td>
<td>0.82</td>
<td>0.86</td>
<td>0.84</td>
<td>21821</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.85</td>
<td>0.84</td>
<td>0.84</td>
<td>45528</td>
</tr>
</tbody>
</table>

Table 4.27: Sleep Precision and Recall with Random Forest

Visualizing Recall and Precision are shown by three metrics as follow:

- Normalized confusion is shown in Figure 4.57.
- Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.58.
Feature selection helps us to determine which features best predict the response value. In this study, Random Forest classifier is used to find feature importance based on HRV dataset. The result is demonstrated in Figure 4.59 and Figure 4.60, which shows that Auto-correlation, PNN50, and RMSSD are most important features in comparison with the others for sleep data.

**Random Forest HRV classification of exam versus not exam:**

Running the optimization in serial with pySOT results in good hyper-parameters as follow:

- number of estimators: 28
- criterion: entropy
Figure 4.59: The feature importance of the forest

Figure 4.60: The feature importance of the forest by label

- max depth: 5
- min samples leaf: 10
- min sample split: 3

Applying leave one person out cross-validation results in the accuracy of 0.739 (+/- 0.259) and AUC accuracy of 0.7224.

The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.28:
Table 4.28: Exam Precision and Recall with Random Forest

Visualizing Recall and Precision are shown by three metrics as follow:

- Normalized confusion is shown in Figure 4.61.

![Normalized confusion matrix](image)

Figure 4.61: Normalized confusion matrix for exam

- Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.62.

![ROC curve](image)

Figure 4.62: Receiver Operating Characteristic (ROC) curve for exam
Random Forest BAC classification of exam versus not exam:

Running the optimization in serial with pySOT results in good hyper-parameters as follow:

- number of estimators: 148
- criterion: entropy
- max depth: 12
- min samples leaf: 7
- min sample split: 4

Applying leave one person out cross-validation results in the accuracy of 0.766 (+/- 0.204) and AUC accuracy of 0.6821.

The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.27:

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.72</td>
<td>0.84</td>
<td>0.77</td>
<td>26374</td>
</tr>
<tr>
<td>1</td>
<td>0.71</td>
<td>0.55</td>
<td>0.62</td>
<td>19154</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.72</td>
<td>0.72</td>
<td>0.71</td>
<td>45528</td>
</tr>
</tbody>
</table>

Table 4.29: Exam Precision and Recall with Random Forest

Visualizing Recall and Precision are shown by three metrics as follow:

- Normalized confusion is shown in Figure 4.63.

![Normalized confusion matrix](image)

Figure 4.63: Normalized confusion matrix for exam

- Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.64.
Figure 4.64: Receiver Operating Characteristic (ROC) curve for exam

Random Forest HRV classification of exercise versus not exercise:

Running the optimization in serial with pySOT results in good hyper-parameters as follow:

- number of estimators: 28
- criterion: gini
- max depth: 5
- min samples leaf: 10
- min sample split: 3

Applying leave one person out cross-validation results in the accuracy of 0.889 (+/- 0.167) and AUC accuracy of 0.6472.

The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.30:

<table>
<thead>
<tr>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.91</td>
<td>0.95</td>
<td>0.93</td>
</tr>
<tr>
<td>1</td>
<td>0.51</td>
<td>0.34</td>
<td>0.41</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.86</td>
<td>0.88</td>
<td>0.87</td>
</tr>
</tbody>
</table>

Table 4.30: Exercise Precision and Recall with Random Forest

Visualizing Recall and Precision are shown by three metrics as follow:

- Normalized confusion is shown in Figure 4.65.
- Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.66.
Running the optimization in serial with pySOT results in good hyper-parameters as follow:

- number of estimators: 28
- criterion: gini
- max depth: 5
- min samples leaf: 10
- min sample split: 3

Applying leave one person out cross-validation results in the accuracy of 0.877 (+/- 0.158) and AUC accuracy of 0.6586.

The quality and quantity of the classifier is measured with precision and recall and the result is shown in table 4.28:
Table 4.31: Exercise Precision and Recall with Random Forest

Visualizing Recall and Precision are shown by three metrics as follow:

- Normalized confusion is shown in Figure 4.67.

![Random Forest Normalized confusion matrix](image)

Figure 4.67: Normalized confusion matrix for exercise

- Receiver Operating Characteristic (ROC) curve is demonstrated in Figure 4.68.

![Receiver Operating Characteristic (ROC) curve](image)

Figure 4.68: Receiver Operating Characteristic (ROC) curve for exercise
Chapter 5

Discussion and Future Work

This chapter provides a brief overview of analyzing heart rate variability and a novel method to assist the performance of the classifier. The conclusion of this study will be reviewed based on our results. Finally, the potential area for future work will be discussed.

5.1 Discussion

The main purpose of this study was to develop an activity classifier for short-term (5 minutes) HRV signals. Initially, seven statistical time domain features from HRV are derived and used to predict the labelled classifier as sleep versus not sleep, exam versus not exam, and exercise versus not exercise.

First, we tested each feature individually for association with class label. For Sleep, all features except SDSD (x5) were significantly associated with class label. For Exam, all were significantly associated. For exercise, the only significant feature was SDNNi (x4). This may indicate that Exercise is a more difficult classification problem than the other two.

Our visualizations by t-SNE and PCA supports the idea that there are more overlaps with exercise which means that the data does not contain enough information for t-SNE and PCA to identify clearly between the classes. This could be due to the lesser number of observations for this activity or that the classification of exercise activity was more difficult. Also, there is less overlap in sleep data which indicates that the selected sleep features can be divided naturally according to their labels.

We used standard supervised learning using these features, and we proposed a new method based on creating a new feature vector which improves the performance of most classifiers. Table 5.1 summarises the classification accuracies (%) and AUC on two datasets (HRV and BAC) and the better performance is highlighted. We note that because of the small data set size (in terms of number of people) the differences are not statistically significant, but they are suggestive of improved performance for the Sleep and Exam tasks.

It can be seen that the categorization of the ECG into three different groups according to their HRV is most accurate by performing SVM and logistic regression in
Table 5.1: Comparison of the classification accuracy (%) and AUC with leave one person out on HRV and BAC datasets
the estimate of the best performance may have some optimistic bias. However, we expect the general trends in performance to hold. Furthermore, the small size of exercise data may be leading to less reliable results for those models in comparison with sleep and exam data.

5.2 Future Work

There are many opportunities to improve this study as well as further analysis of heart rate variability, some of which are provided in this section.

First, further verification and experimentation of the BAC technique with larger datasets would allow us to verify that it provides a performance boost. The idea may also be applicable to other domains where we have individual people who generate a significant amount of data that has individual characteristics.

Frequency domain, wavelet transform, and nonlinear methods of heart rate variability analysis is suggested to be compared to the proposed method. Also, neural network techniques such as LSTM which is the popular method in analyzing time series data [68] can be considered for future study. With both of these approaches we can combine our new baseline adaptive classifier idea to examine if it increases performance.

Finally, heart rate variability is a significant biological signal which can be used in many clinical studies. So, increasing the number of participants will be useful for further investigation of heart rate variability and its relationship with different clinical fields such as stress.
Bibliography


[70] R Wood. Resting heart rate chart.

Appendix A

The extra table belonging to the analysis in this research is provided here. The following table represents some partitions of the HRV segmentation during 5-minute recording windows for exam, sleep, and exercise.
Curriculum Vitae

Name: Elham Harirpoush

Post-Secondary Education:
Master of Science in Computer Science ,
The University of Western Ontario, London, Canada
2016 - 2018

Related Work Experience:
Teaching Assistant,
The University of Western Ontario
2016 - 2018

Research Assistant,
The University of Western Ontario
2016 - 2018