EARLY PREDICTION OF CARDIAC ARREST THROUGH HEART RATE VARIABILITY ANALYSIS

A Thesis

presented to

the Faculty of California Polytechnic State University

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Luke Plewa

June 2015

COMMITTEE MEMBERSHIP

TITLE:	Early Prediction of Cardiac Arrest through Heart Rate Variability Analysis
AUTHOR:	Luke Plewa
DATE SUBMITTED:	June 2015
COMMITTEE CHAIR:	Assistant Professor Foaad Khosmood, Ph.D. Department of Computer Science
COMMITTEE MEMBER:	Professor John Clements, Ph.D. Department of Computer Science
COMMITTEE MEMBER:	Professor Alexander Dekhtyar, Ph.D. Department of Computer Science

ABSTRACT

Early Prediction of Cardiac Arrest through Heart Rate Variability Analysis

Luke Plewa

The increase in popularity for wearable technologies (see: Apple Watch and Microsoft Band) has opened the door for an Internet of Things solution to healthcare. One of the most prevalent healthcare problems today is the poor survival rate of out-of hospital sudden cardiac arrests (9.5% on 360,000 cases in the USA in 2013). It has been proven that heart rate derived features can give an early indicator of sudden cardiac arrest, and that providing an early warning has the potential to save many lives [19, 23, 2]. Many of these new wearable devices are capable of providing this warning through their heart rate sensors.

This thesis paper introduces a prospective dataset of physical activity heart rates collected via Microsoft Band. This dataset is representative of the heart rates that would be observed in the proposed Internet of Things solution. This dataset is combined with public heart rate datasets to provide a dataset larger than many of the ones used in related works and more representative of out-of-hospital heart rates. This paper introduces the use of LogitBoost as a classifier for sudden cardiac arrest prediction. Using this technique, a five minute warning of sudden cardiac arrest is provided with 96.36% accuracy and F-score of 0.9375. These results are better than existing solutions that only include in-hospital data.

Keywords: Heart Rate Variability, Machine Learning, Artificial Intelligence

ACKNOWLEDGMENTS

This thesis is inspired by the work of Michael C. Plewa, MD on sudden cardiac death prediction through ECGs [26]. This paper notes the required sample size for sudden cardiac death prediction as well as the ability and features to predict sudden cardiac death.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

According to the American Heart Association, approximately 360,00 cardiac arrests occurred outside of the hospital in the United States during 2013, with a survivor rate of only about 9.5% [9]. Survival rates increase dramatically for in-hospital cardiac arrest.

Researchers believe that cardiac arrest may be predicted in advance [34, 8, 5]. Specifically, through Heart Rate Variability (HRV) Analysis Algorithms, Machine Learning, the Internet of Things, and Big Data, we may be able to monitor at-risk individuals and give them advance warning (1-4 hours) to get to a hospital. There is a divide between papers focused on Electrocardiogram (ECG) features and HRV features. One such study conducted by researchers at Carnegie Mellon University and Chicago University predicted Code Blue, the callsign for someone going into cardiac arrest, with around 65% recall and 20% false positive rate on test data at 4 hours ahead of the event [34] that focuses on ECG features. Another study was able to make predictions with the accuracies of 99.73%, 96.52%, 90.37% and 83.96% for one, two, three, and four minutes prior to the event respectively with ECG features [8]. Murukesan et al. [23] achieve 96.36% accuracy through the use of a SVM and HRV features. This paper uses a five minute advance warning and two minutes of sample data. This is the baseline that this paper uses. HRV is the preferred feature set due to allow for a broader spectrum of potential commercial wearable devices. A significant benefit of such advance warning is corroborating evidence of the need to go to the hospital rather than ignoring symptoms [2]. Developments in wearable technology and advancements in non-intrusive heart rate monitors may allow for a future where people can stream their heart rate readings, with the readings automatically analyzed by robust machine learning algorithms which will alert cardiac arrest risk [22].

This field is new and there is room for progress through additional studies and the development of HRV Analysis Algorithms. This is supported by the increasing number of medical records stored electronically. The Carnegie Mellon University study, conducted in partnership with the NorthShore University HealthSystem, involved the use of patient-level information for about 133,000 in-hospital patients extracted from electronic medical records (EMR) from 2006 to 2011 [34]. This thesis utilizes the MIMIC II dataset from PhysioNet, which contains records dating back to 1970 [12].

Current machine learning methods for sudden cardiac arrest have not been tested against physically active heart rates. All tested samples come from in-hospital data where the individuals are at a resting state. This thesis hopes to advance the state of the art for cardiac arrest prediction for the wearable world through HRV analysis and develop a solution that can be reliable with high accuracy and f-score. Using previous works [23, 8, 19] as a baseline, new machine learning approaches are tested in this thesis.

The contributions of this thesis are as follows:

- A LogitBoost approach to sudden cardiac arrest prediction.
- An implemented and proposed Internet of Things solution for collecting physical activity heart rate data and potentially warning users of the onset of sudden cardiac arrest.
- A LogitBoost classifier for the 0.16.1 version of the Scikit-Learn Python library that accomodates for different weak learner classes instead of multiples of the same learner [25].

Chapter 2

Background

2.1 The Heart

The heart undergoes states of depolarization and repolarization when it beats [11]. Echoes of these states are sent electronically throughout the body. Cardiologists use electrodes, sensitive receivers, to pick up those electronic echoes. In cardiology, these electrodes are named leads. In a standard 12 lead system, six leads will go onto the chest and six leads will be placed along the limbs. These leads give a multidimensional view of the heart. The recording of this electrical activity is called an electrocardiogram (ECG).

2.2 ECG Signal

There are five nodes on the ECG signal which are used to derive different features. These nodes are labeled P, Q, R, S, T, and U. These are broken into a P wave, QRS complex, a T wave, and a U wave [11]. One of these features is the RR interval, the period between two R nodes on an ECG. This interval is representative of a person's heart rate. The distance between two R waves represents a person's instantaneous heart rate. See Fig. 2.1 for more details on the ECG signal.

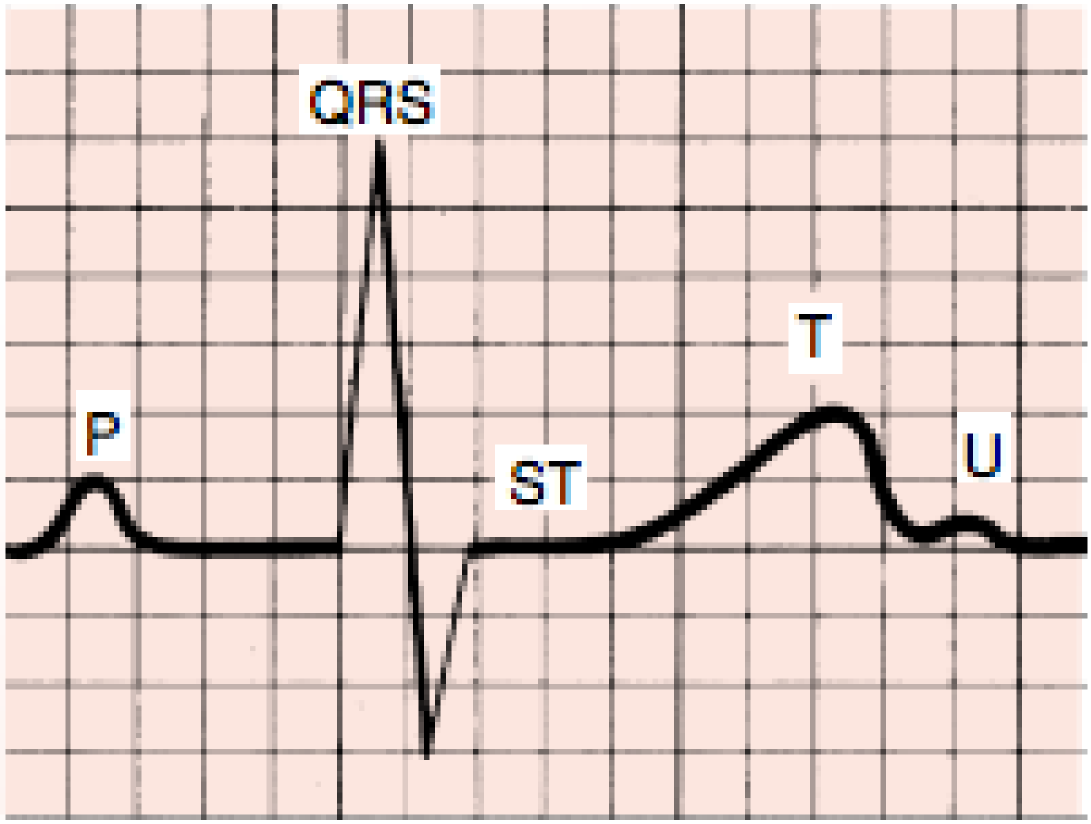

Figure 2.1: **ECG Signal for a Normal Sinus Rhythm**:
ECG Signal for a Normal Sinus Rhythm. The P wave, QRS complex, T wave, and U wave are labelled [11]. The P wave represents atrial depolarization. The QRS complex represents ventricular depolarization. The ST segment, T wave, and U wave represent ventricular repolarization.

2.3 Heart Rate Variability

Heart rate is a feature which can be derived from the ECG signal [11]. This is done by comparing the placement between the RR intervals. Each peak on the R wave represents a node in the heart rate array. The time between each R wave peak is the instantaneous heart rate. It is potentially inconvenient to spend the whole day with 12 electrodes attached to the body, and there is technology to track the heart rate without use of an ECG signal. This creates a preference to have a system which relies on more convenient sensors (wearable devices) in comparison to 12 ECG electrodes, despite the electrodes having the potential to provide more information.

2.4 Arrhythmias

A regular ECG signal is called a regular sinus rhythm. There are specific classes of ECG signals representative of what is going on inside the heart. There are a couple key vocabulary words that are used to identify these different classes [11].

1. Arrhythmia - An abnormal heart rhythm associated with the failure of the heart's electrical system.
2. Tachycardia - A class of rhythms that occur when the heart beats too quickly.
3. Bradycardia - A class of rhythms that occur when the heart beats too slowly.
4. Atrium - The top two chambers of the heart.
5. Ventricles - The bottom two chambers of the heart.

Based on these terms, we can determine an arrhythmia such as Ventricular Tachyarrhythmia (VT) is a rapid heart beat that starts in the bottom chambers of the heart [11]. VT is one of the main arrhythmias associated with sudden cardiac arrest [9]. These arrhythmias have certain classifying features. For example, VT is easily identified by the fast oscillatory waves, which represent the rapid twitching of the heart. This is shown in Fig. 2.2.

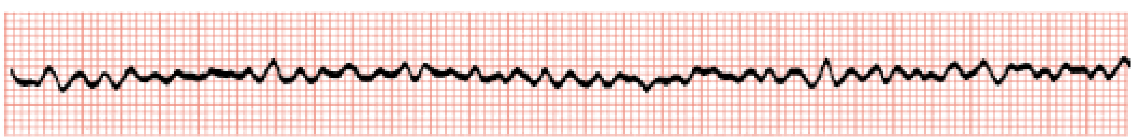

Figure 2.2: **ECG Signal for Ventricular Tachyarrhythmia**:
ECG Signal for Ventricular Tachyarrhythmia [11]. A notable feature is the short period of the waves.

2.5 Cardiac Arrest

Cardiac arrest occurs when the beating of the heart and all electrical activity stops [11]. This means that blood stops pumping to the body, and is especially important because brain damage can occur within ten minutes from blood loss to the brain [9]. This is different from a heart attack, which is a physical failure in comparison to an electrical failure. Heart attacks are caused by the blockage of blood flow to the heart. To use analogies, heart attacks are caused by faulty plumbing while cardiac arrest is caused by a power outage. Sudden cardiac arrest (SCA) is cardiac arrest that occurs unexpectedly and can result in death, called sudden cardiac death (SCD) [26].

There are three main arrhythmia classes that correlate with cardiac arrest: Ventricular Tachyarrhythmia (VT), Ventricular Asystole (VA), and Pulseless Electrical Activity (PEA). VT is described in the previous section, and is the most common predecessory to SCA [11]. VA is represented by a flat line. This is shown in Fig. 2.3. PEA is especially difficult, as the person may have regular rhythms, but not enough mechanical activity to pump blood effectively.

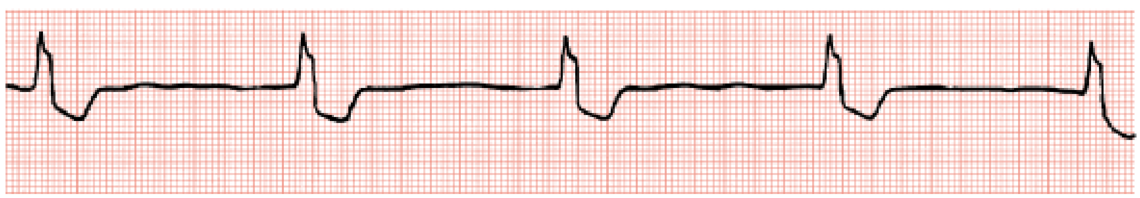

Figure 2.3: **ECG Signal for Ventricular Asystole**:
ECG Signal for Ventricular Asystole [11]. A notable feature is the long duration of a flat reading.

2.6 Machine Learning

In the history of SCA prediction, various machine learning algorithms are in use. This section highlights the five machine learning classification algorithms explored in this paper. In this case, datasets are broken up into a healthy (non-SCA) class and a SCA warning class (five minutes until cardiac arrest) to create a two class classification problem. Kotsiantis et al. [21] provide a strong review of many classification algorithms. The implementations of each algorithm is provided via the 0.16.1 version of the Scikit-Learn Python library [24].

2.6.1 Supervised Learning

The techniques used in this paper are all supervised learning techniques. The data is all labelled either by hand (in the case of the Microsoft Band physical dataset) or automatically (collected via spreadsheet). This provides labelled classes for the algorithms.

2.6.2 Classification

Classification is a machine learning problem where a data sample can belong to one of two or more classes [21]. While classification algorithms can deal with more than two classes, this paper focuses on a binary classification problem where the two possible classes are either normal sinus rhythms or the onset of sudden cardiac arrest. Machine learning classifiers use the feature vectors derived from the data samples to learn how these features help identify what class future data samples belong to.

2.6.3 Support Vector Machine

A Support Vector Machine (SVM) is a machine learning algorithm for classification proposed by Cortes and Vapnik in 1995 [6]. The main idea of SVMs is to learn a non-linear function by a combination of linear mappings in high dimensional feature space. The number of features to the dimensionality of the input space. The desired function would be able to map all training data within a certain margin of error. Support vectors are especially good in high dimensionality spaces because SVMs do not depend upon the dimensionality of the input space.

For example, a SVM could draw a line that separates two classes in 2-D space across features A and B. In Fig. 2.4 this is shown by x and y axes. The SVM classifier attempts to draw the best line that separates the two classes of dots. This is just across two features. The problem is brought into higher dimensionalities as more features are included.

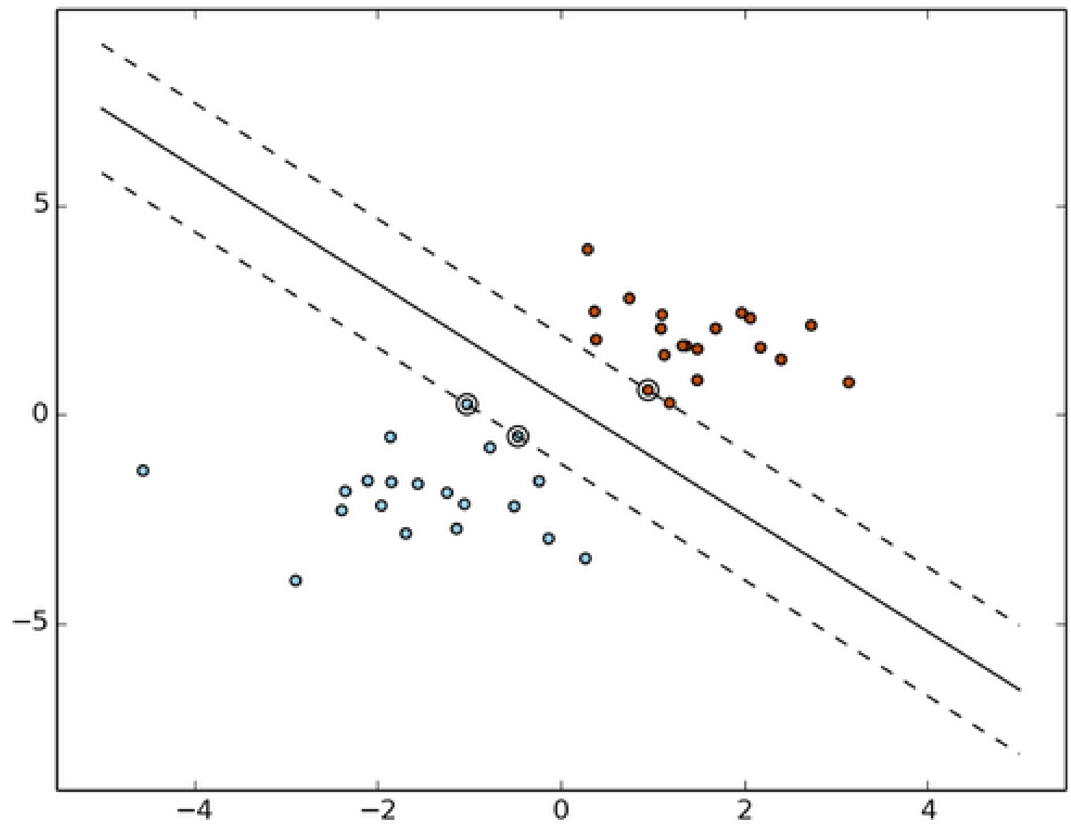

Figure 2.4: **Support Vector Machine Example**:
Support Vector Machine Example [24].

2.6.4 Stochastic Gradient Descent

Stochastic Gradient Descent is a classifier that uses a loss function to improve a set of parameters [24]. The algorithm differs from Gradient Descent in that each iteration only updates for one training sample. In comparison to the batch algorithm, this algorithm is extremely efficient because it only updates its parameters for one sample at a time. The algorithm has a weakness in its sensitive towards feature scaling (normalization).

2.6.5 Decision Trees

Decision Trees build a ruleset to classify input data [24]. Each node in the tree that is built represents a decision. As the input data traverses through this tree, it will eventually arrive to a leaf node which dictates the class of the input data. The trees built have logarithmic run times due to the structure. Unfortunately, the ruleset is prone to overfitting for which there are a number of techniques to avoid.

2.6.6 Naive Bayes

Naive Bayes is a classification algorithm based on the Bayes' theorem, seen in Fig. 2.5.

1. P(Class) and P(Features) are the probabilities of the set of features and the class without regard to each other.
2. P(Class — Features) is the conditional probability of a class given a set of features.
3. P(Features — Class) is the probability of a set of features given the class.

This classifier uses the theorem to derive the probability that a given feature vector is associated with a specific class. The algorithm naively assumes that there is an independence between every pair of features [24, 21]. This assumption creates a weakness in the algorithm, because there will almost never been an indepdendence between every pair of features given enough features. Regardless, the classification algorithm is proven to be strong in smaller training sets.

2.6.7 Boosting

The main idea of the boosting algorithm is to use a set of weak learners to help build a better classifier (called the strong learner) [32]. Each weak learner is assigned a weight during training which indicates how much consideration that weak learner's

$$P(\text{Class} \mid \text{Features}) = \frac{P(\text{Class}) * P(\text{Features} \mid \text{Class})}{P(\text{Features})}$$

Figure 2.5: **Bayes' Theorem**:
Bayes' Theorem.

classification is used during testing. The strength of this algorithm is that each weak learner needs to only be a little bit better than a guess, and the distribution of weights will allow the strong learner to create accurate predictions.

This paper utilizes the LogitBoost algorithm. Fig. 2.6 shows the algorithm demonstrated by Schapire and Fruend [32]. LogitBoost is different from other boosting algorithms in that it uses the log-likelihood loss for binary classification.

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in \mathcal{X}$, $y_i \in \{-1, +1\}$.
Initialize: $F_0 \equiv 0$.
For $t = 1, \ldots, T$:

- For $i = 1, \ldots, m$:

$$p_t(i) = \frac{1}{1 + e^{-F_{t-1}(x_i)}}$$

$$z_t(i) = \begin{cases} \frac{1}{p_t(i)} & \text{if } y_i = +1 \\ -\frac{1}{1-p_t(i)} & \text{if } y_i = -1 \end{cases}$$

$$w_t(i) = p_t(i)(1 - p_t(i)).$$

- Choose $\alpha_t \in \mathbb{R}$ and $h_t \in \mathcal{H}$ to minimize (or approximately minimize if a heuristic search is used)

$$\sum_{i=1}^{m} w_t(i)(\alpha_t h_t(x_i) - z_t(i))^2.$$

- Update: $F_t = F_{t-1} + \alpha_t h_t$.

Output F_T.

Figure 2.6: **LogitBoost Algorithm**:
LogitBoost Algorithm [32].

Chapter 3

Related Work

3.1 University of California, Irvine Dataset

Reiss and Stricker at University of California, Irvine (UCI) host the only existing public dataset for heart rates of physical activity called 'PAMAP2 Dataset: Physical Activity Monitoring' [30]. The dataset is sampled using an HR monitor with a frequency at 9 Hz collected in 2012. The dataset is comprised of the heart rates of nine individuals performing 18 different physical activities. The dataset also holds other biometric values such as body mass index, acceleration data, temperature, gyroscopic data, and orientation. The activities covered are shown in Fig. 3.1.

3.2 Feature Extraction

ECG signals contain a set of features that cardiologists use to identify them by. Commonly used features include heart rate, PR interval, RR interval, QT interval, period of T wave, period of P wave, and period of QRS complex [1, 36]. All durations are usually measured in milliseconds. Many approaches also use age, sex, height, and weight to accomodate these features.

Heart rate can be derived from the ECG signal. The peak of each R wave of an ECG signal respresents a node in the heart rate array. Datasets are commonly composed of the time between these peaks (the length of the RR interval) [12, 19].

1 lying
2 sitting
3 standing
4 walking
5 running
6 cycling
7 Nordic walking
9 watching TV
10 computer work
11 car driving
12 ascending stairs
13 descending stairs
16 vacuum cleaning
17 ironing
18 folding laundry
19 house cleaning
20 playing soccer
24 rope jumping
0 other (transient activities)

Figure 3.1: **Physical Activities covered in the UCI Physical Activity Monitoring Dataset**:
PAMAP2 Dataset (2012): These physical activities are performed by nine individuals whose biometric values are recorded [30].

3.3 Classification of ECG Signals

Some of the earliest work with machine learning on ECGs is the classification of ECG signals [35, 36]. Signals are broken down into different classes of arrhythmias (see section 2.2). The features of ECG signals are used to classify the ECG. Much of the early work in ECG classification is done with neural networks [7, 16].

Jinkwon Kim et al. [20] use the MIT-BIH arrhythmia database to compare classification using an extreme learning machine, back propogation neural network, radial basis function neural network, and support vector machine. They classify into six classes of ECG. The authors find that the extreme learning machine performs faster than the other three approaches, and their system has 98.72% average accuracy.

Soman and Bobbie [33] use OneR, J4.8, and Naive Bayes on a dataset from University of Califonia, Irvine (UCI). The authors classify into 16 classes on 279 attributes and 452 instances. OneR has an accuracy of 61.28%, J48 has an accuracy of 91.81%, Naive Bayes has an accuracy of 76.55%. J48 requires the most training to show maximum success, but Naive Bayes and OneR do not. This gives Naive Bayes as a happy medium of requiring relatively little data and yielding higher results.

Polat and Gunes [28] use principal component analysis and least square support vector machine. Similarly to Soman and Bobbie [33], they use the UCI database. The authors found 96.86% accuracy with a 50-50% training-test split, and 100% accuracy with a 70-30% split.

3.4 Cardiac Arrest Prediction through ECG Features

While classification of ECG signals is related to this paper, closer focus is put upon the prediction of cardiac arrest. By supplying a warning indicator for cardiac arrest, there will be a reduction in sudden deaths outside of the hospital [2].

3.4.1 Support Vector Machine with Logistic Regression Solution

Somanchi et al. [34] use a support vector machine with logistic regression (SVR) to predict cardiac arrest up to four hours in advance. The authors use modified early warning score (MEWS) as a baseline. They were able to get 65% recall with 20% false positive rate at four hours before cardiac arrest. This is significantly better than MEWS, which scored 30% recall. The authors use the NorthShore University HealthSystem dataset, which includes patient-level information for 133,000 in-hospital patients. Of those 133,000 patients, there are only 815 cardiac arrests. This means the distribution is extremely inbalanced. The dataset includes vitals and lab tests, and is not entirely focused on the ECG. Also all of the patients were from the hospital, and cardiac arrest occurs 80% of the time outside of the hospital [9].

3.4.2 Modified Early Warning Score Solution

Churpek et al. [5] researched predicting cardiac arrest through a nested case-control study of 88 patients that experience cardiac arrest in a hopsital. Their dataset included 352 patients who do not experience cardiac arrest, so the ratio (1:4) is better than Somanchi et al. [34]. The authors focused entirely on modified early warning score (MEWS), and found that MEWS showed differences 48 hours before cardiac arrest for cardiac arrest patients compared to the control patients. The authors found that maximum MEWS outperformed maximum respiratory rate, maximum heart rate, maximum pulse pressure index, and minimum disatolic blood pressure as an indicator for cardiac arrest 48 hours in advance.

3.5 Cardiac Arrest Prediction through HRV Features

This section is devoted to papers that focused on HRV features rather than ECG features. In all cases, the HRV is derived from ECG signals [8, 23, 19]. In contrast, this paper includes a dataset that does not have ECG-derived heart rates.

3.5.1 k-Nearest Neighbor Solution

Ebrahimzadeh et al. [8] use k-nearest neighbor (k-NN) and multilayer perceptron neural network (MLP) to classify patients that will experience cardiac arrest. The authors use time-frequency and nonlinear features from heart rate variability (HRV) of ECG signals to improve this prediction task. The authors achieved accuracies of 99.73%, 96.52%, 90.37% and 83.96% for first, second, third, and forth one-minute intervals respectively.

3.5.2 Support Vector Machine Solution

Murukesan et al. [23] achieve 96.36% accuracy through the use of a SVM. The authors use the MIT/BIH Sudden Cardiac Death database (23 subjects) and the MIT/BIH Normal Sinus Rhythm database (18 subjects) [12]. The authors use five minutes of the ECG signal for HRV features which is two minutes prior to the onset of SCA. The authors of this paper focus specifically on the selection of features using the Sequential Feature Selection algorithm. The authors start with 34 features but determined that the selection of seven features provided optimal results. The authors also test with a Probabilistic Neural Network to achieve a lesser 93.54%. The feature selection portion of this paper is advanced, but there are a small amount of samples used. This means that their solution is more likely to be specialized to the dataset and not representative of the real world.

3.5.3 Artificial Neural Networks Solution

Joo et al. [19] use artificial neural networks to achieve a 76.6% accuracy. The authors use the Spontaneous Ventricular Tachyarrhythmia Database [12] which contains 106 pre-VT records and 126 controls sets. The authors examine a five minute window of data that exists ten seconds before the event.

Chapter 4

Evaluation

4.1 Expected Results

The results of this paper are expected to be better than a support vector machine in predicting if a person could experience cardiac arrest in the next five minutes, based on heart rate variability. This is based on the baseline provided by Murukesan et al. [23] and their support vector machine implementation.

4.2 Hypothesis

With the increased interest in wearable technology, there will be a market for HRV monitoring [22, 10, 17]. With the large amount of cardiac arrests that occur outside of the hospital, constant HRV monitoring may be able to prevent a large amount of deaths [9]. This constant HRV monitoring will have less data than typical in-hospital data, and will have to be mostly reliant on heart rate data as well as some patient-level data [22]. There has been success in predicting cardiac arrest with in-hospital data [34], but the potential for success may exist with limited data with physical activity as well. Success in this case is defined as a system that is more accurate, in terms of accuracy and F-score, than the linear SVM classifier which has previously provided the best results on a similar experiment in Murukesan et al. [23].

4.3 Proposed System

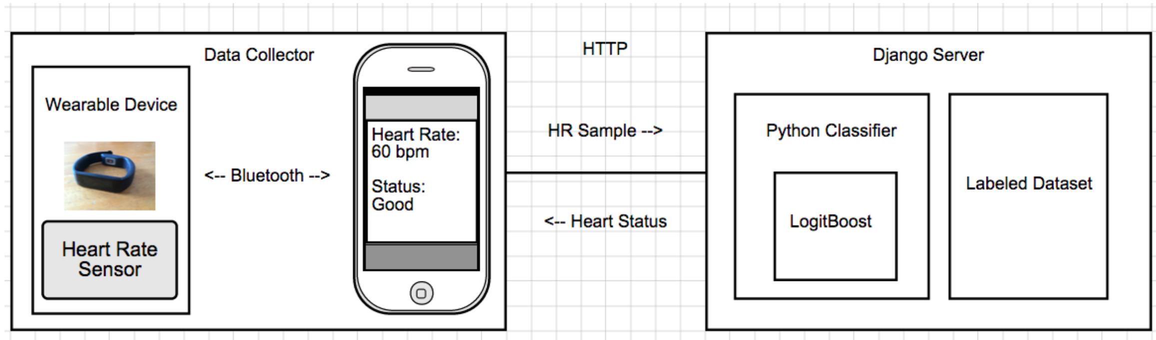

Figure 4.1: **Proposed System**:
The proposed Internet of Things system for use with wearable devices that have a heart rate monitor.

Fig. 4.1 details an outline of the proposed system, similar to the system that builds the Physical Activity Dataset. The wearable device collects heart rate information using its heart rate monitor. This example uses the Microsoft Band, but the system may be built for any wearable device that has a heart rate monitor or electrodes for ECG recording. That data is sent to a smart phone via bluetooth. An application on the phone holds this data in two minute intervals. Data may be sent to the phone on each beat or all at once. At the end of each two minute interval, the sample is sent via HTTP request to a Django server hosting the Python machine learning code. Also on this server is the dataset of labelled heart rate samples, which has the opportunity to grow with each new input sample. The running code will classify the sample and send back a response. If the sample is classified as the onset of sudden cardiac arrest, then the phone and wearable device's notification system is used to notify the user. Otherwise, the phone sends no warning notification.

4.3.1 Microsoft Band

Figure 4.2: **Microsoft Band 1**:
The Microsoft Band. The closest panel is a LED display which shows panels such as heart rate and GPS information. The metallic part in the back is the heart rate monitor.

Figure 4.3: **Microsoft Band 2**:
The Microsoft Band being worn.

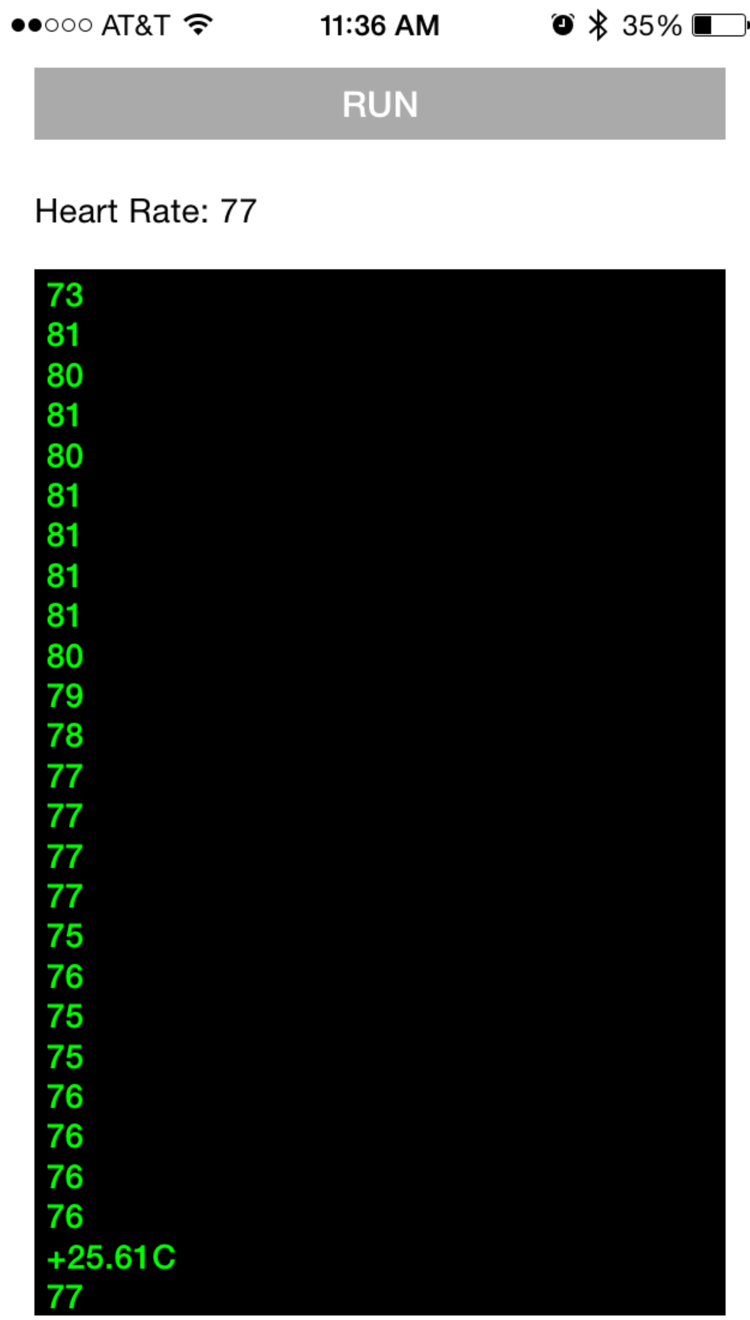

Figure 4.4: **Microsoft Band iOS App**:
Microsoft Band iOS App

The Microsoft Band is a commercial product available to the public. It features a heart rate monitor, skin temperature sensor, GPS tracker, and other sensors. It is visible in Fig. 4.2 and Fig. 4.3. Microsoft has provided a SDK that allows access of the band's sensor readings. The band communicates via Bluetooth to a synchronized smart phone. The sample iOS app is shown in Fig. 4.4. The integer values displayed are the instantaneous beats per minute (bpm) which can be used to derive the heart rate.

4.4 Variables

4.4.1 Datasets

The datasets used are divided into control and arrhythmia datasets. Included in the control datasets is a prospective physical activity dataset collected via the Microsoft Band. Prior to this work, there is only one dataset that provides the heart rates of individuals performing physical activities [30]. Each dataset is composed of text files which hold the RR interval lengths (heart rates).

Murukesan et al. [23] use the MIT-BIH Sudden Cardiac Death database and the MIT-BIH Normal Sinus Rhythm database [12]. Table 4.1 details the comparison of each dataset in terms of number of subjects, year of the collection, and location of the collection. Every dataset uses ECG recordings, and the heart rates are dervied via a Physionet algorithm [12].

Dataset Name	**Subjects**	**Year**
MIT-BIH Normal Sinus Rhythm [4]	18	2000
Normal Sinus Rhythm RR Interval [12]	54	1992 and 1995
Sudden Cardiac Death (SCD) Holter [13]	18	1989
MIT-BIH Malignant Ventricular Arrhythmia [12]	22	2000

Table 4.1: **Dataset Comparisons**:
Dataset Comparisons.

Activity	Samples Count
Sitting	4
Computer Work	13
Walking	10
Stretching	3
Car Driving	9
Video Games: Super Smash Bros.	22
Lying	6
Eating	8
Post-Workout	12
Video Games: Call of Duty	18
Total	105

Table 4.2: **Physical Heart Rate Activity Dataset Distribution**: The distribution of activities collected by the Microsoft Band.

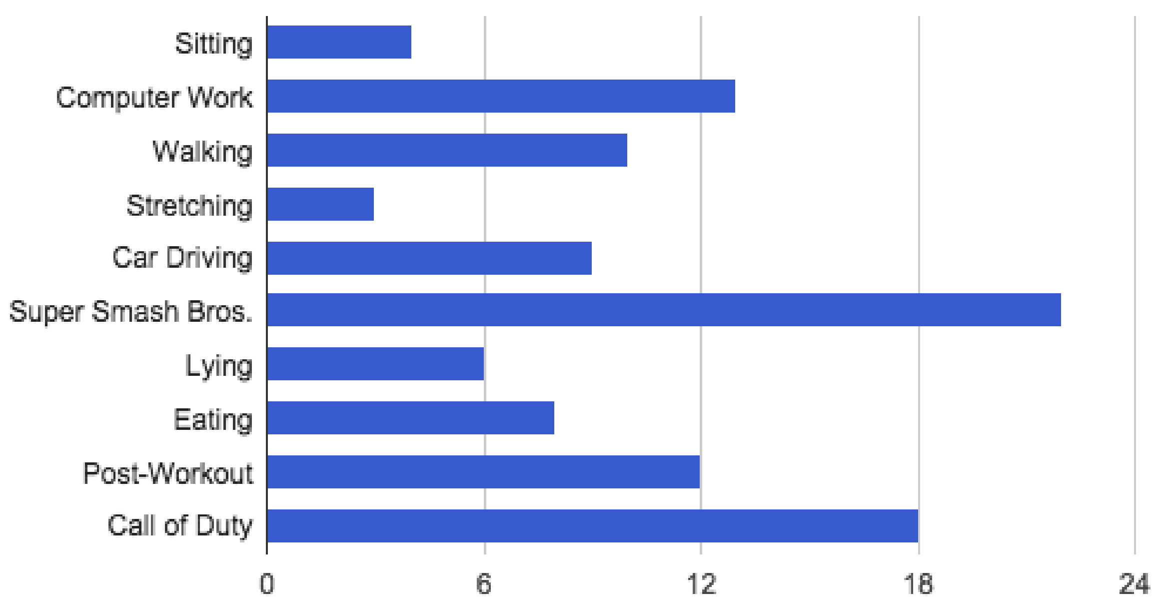

Figure 4.5: **Physical Heart Rate Activity Dataset Distribution**: The distribution of activities collected by the Microsoft Band.

This paper introduces the use of a prospective study that may be representative of future heart rates of people using a commercially available wearable device. The prospective physical activity dataset in this paper is collected from a Microsoft Band. This dataset features the two minute samples of the heart rates of five individuals performing ten unique activities. The distribution of these activities is detailed in Fig. 4.5. A table is provided with the counts as well in Table 4.2. There are a total 105 samples in this dataset. The physical features of these product evaluators is available in Table 4.3. All product evaluators are current Cal Poly San Luis Obispo students. Each evaluator represents a range of physical fitness, as the resting heart rates range from 54 beats per minute (bpm) to 70 bpm. The derived features of these samples is shown in Fig. 4.6.

For data collection purposes, the product evaluators are told to breathe and move normally. The recording is a two minute sample that is started and ending during the activity, and not before or after. Each sample is from a normal workday hours (9 am to 5 pm) and not during any exhausting timeslot (after or before waking hours). Each activity is sampled with at least a two minute resting period between each activity so that there is no bleeding between activities. This bleeding may have an effect on the heart rates and is avoided [11]. This connection of the strap to the evaluator's wrist is checked before and after each sample to make sure that the connectivity of the sensor to the wrist is strong. Each beat is sent to an iPhone which hosts an application that stores the two minute recording before shipping it off via HTTP PUT request to a Google Cloud server. During the collection process, no product evaluators had heart or physical conditions with no prescriptions that would affect results. The product evaluators also sign a form agreeing to volunteer for the evaluation.

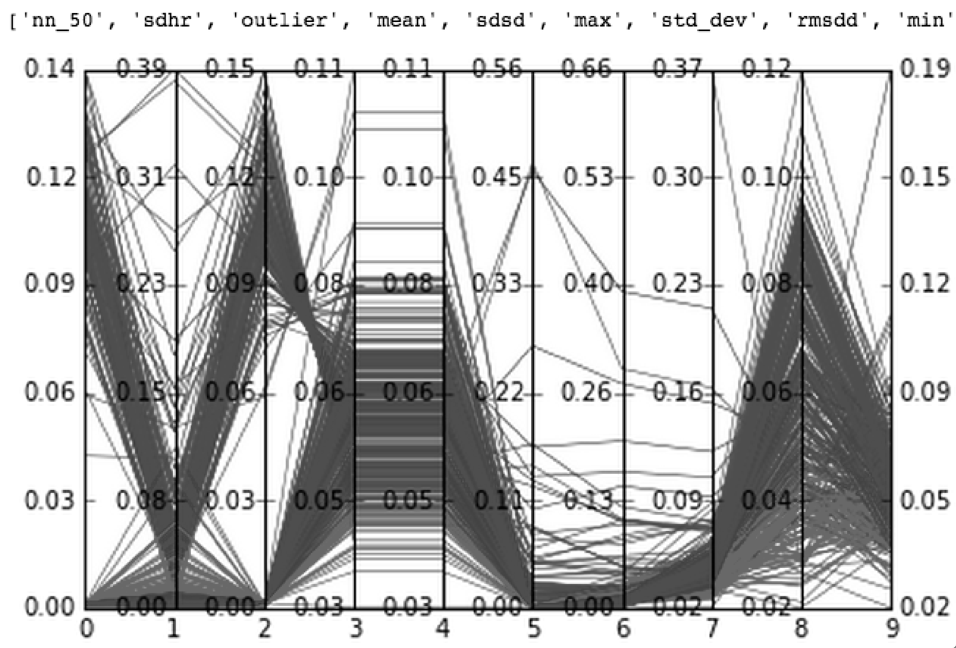

Figure 4.6: **Physical Heart Rate Activity Dataset Features**:
Parallel coordinates graph of the features of activities collected by the Microsoft Band.

Evaluator ID	Gender	Age	Height	Weight	Weekly Exercise Amounts
0	Male	23	5 Feet 8 Inches	150 lbs.	1 times per week
1	Male	23	5 Feet 8 Inches	160 lbs.	6 times per week
2	Male	23	6 Feet 2 Inches	200 lbs.	6 times per week
3	Male	21	6 Feet 0 Inches	180 lbs.	4 times per week
4	Male	21	6 Feet 2 Inches	220 lbs.	2 times per week

Table 4.3: **Physical Heart Rate Activity Dataset Evaluators**:
Physical Heart Rate Activity Dataset Evaluators. Each evaluator is a Cal Poly San Luis Obispo Student.

The control datasets are the Physiobank Normal Sinus Rhythm Dataset [4] and the MIT-BIH Normal Sinus Rhythm Database [12]. These datasets contain the normal sinus rhythm classs of ECG signals and show no problems or warnings for cardiac arrest. The heart rates derived from this dataset are collected using the RR interval. The PTB database contains 18 long term recordings. The MIT-BIH database contains 290 subjects with mixed rhythms, but all are normal sinus rhythms. Two minutes of heart rate data are randomly sampled for each healthy recording. This is to properly align with the unhealthy recordings which sample two minutes of heart rate data as well.

The datasets which contain other classes of arrhythmias and a potential for cardiac arrest are the MIT-BIH Sudden Cardiac Death (SDC) Holter Database [13] and the Malignant Ventricular Arrhythmia Database [13, 12]. The Holter database contains 23 half-hour episodes of cardiac arrest. The Malignant Ventricular Arrhythmia Database contains 83 half-hour episodes. The heart rates derived from this dataset are collected using the RR interval. Not all samples are used, due to the restriction of requiring a five minute warning time and taking samples of the two minutes before that. There are two samples in this dataset that are discarded, because the sudden cardiac arrest event occurs earlier than the required seven minutes of sampling. Thus, a total of seven minutes of sampling before the event is required.

4.4.2 Features

The independent variables for this paper are based on common features that act as indicators for cardiac arrest and arrhythmia classes [1, 19] . This includes common HRV features such as mean HR, minimum HR, maximum HR, standard deviation of the HRs, standard deviation of the beats per minute, median of the HR, root mean square of the standard deviation of the HR, and the number of outliers. These features are shown in Table 4.4. Another change to the experiment is the percentage split of training to testing data. By controlling the amount of data allowed for learning and identification, specific approaches come out ahead.

Number	Features
1	Mean HR
2	Minimum Instantaneous HR
3	Maximum Instantaneous HR
4	Standard Deviation of the HRs
5	Standard Deviation of the Beats per Minute
6	Median of the HRs
7	Root Mean Square of the Standard Deviation of the HRs
8	The Number of Outliers (greater than 50 ms difference)

Table 4.4: **Features Extracted by the System**:
These HRV features are typical of other experiments [23, 8, 19].

4.4.3 Labels

The dependent variable in this experiment is the class of arrhythmia. This experiment tests using binary classification. The person may have a heart rate that is normal or a heart rate that is exhibiting an onset of sudden cardiac arrest. Being able to classify the rhythm is important so that individuals may take certain precautions or be able to detect and treat specific symptoms.

4.5 Measures

4.5.1 Accuracy

All data comes labelled regarding the class of the heart rates as well as the time until cardiac arrest [12, 4, 1]. The output of the machine learning algorithms is directly compared to these classes, and is judged as wrong or right on a binary scale. Accuracy (the number of correct predictions divided by the total number of test sets) is used to determine the proficiency of the algorithm. The accuracy of these classifiers are compared against baselines such as the one provided by Murukesan et al. [23]. In all cases, at least five runs were performed to collect an average accuracy. There are many cases where more than five runs were performed due to high variance in the results, and it is noted as such.

4.5.2 Precision, Recall, and F-score

Oftentimes, it has been determined that unbalanced datasets such as this one cannot have reliable classification results with accuracy alone [24]. In this case, a classifier has the potential to nearly always guess that the sample is a normal sinus rhythm, because the number of normal sinus rhythms is twice that of the sudden cardiac arrest onsets. Because of this, the LogitBoost algorithm is also evaluated in terms of Precision, Recall, and F-score. These evaluation metrics are based on the ideas of positives and negatives where:

1. Positive = Class of the positive label, in this case the onset of sudden cardiac arrest class of heart rates.
2. Negative = Class of the negative label, in this case the normal sinus rhythm class of heart rates.
3. True Positive = A correctly predicted positive case. The classifier correctly identified the onset of sudden cardiac arrest.
4. False Positive = An incorrectly predicted positive case. The classifier did not correctly identified the onset of sudden cardiac arrest.
5. True Negative = A correctly predicted negative case. The classifier correctly identified the normal sinus rhythm.
6. False Negative = An incorrectly predicted negative case. The classifier did not correctly identify the normal sinus rhythm.

Precision is the number of true positives predicted divided by the sum of the number of true positives and false positives, seen in Fig. 4.7. Recall is the number of true positives divided by the sum of the number of true positives and false negatives, seen in Fig. 4.8. F-score uses both precision and recall to evaluate classification, seen in Fig. 4.9. F-score is determined by the product of twice the precision and recall divided by the sum of the precision and recall.

$$\text{Precision} = \frac{\text{Num. of True Positives}}{\text{Num. of True Positives + Num. of False Positives}}$$

Figure 4.7: **Precision**:
Precision.

$$\text{Recall} = \frac{\text{Num. of True Positives}}{\text{Num. of True Positives + Num. of False Negatives}}$$

Figure 4.8: **Recall**:
Recall.

$$\text{F-score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision + Recall}}$$

Figure 4.9: **F-score**:
F-score.

4.6 Experimental Protocol

The empirical study is broken down into multiple parts. First, data is acquired from the various datasets [13, 4, 12]. Second, multiple machine learning classifiers are randomly trained upon a percentage of the total dataset. Third, the rest of the dataset is used to test against these classifiers and provide an accuracy to demonstrate the proficiency of the algorithm.

4.6.1 Data Acquisition

The first step is the collection and normalization of all data. The datasets are from different sources and not all features are the same, making feature extraction and normalization necessary. This creates 379 two minute samples of heart rates. 101 samples are derived from the ECG signals in the MIT-BIH Sudden Cardiac Death (SDC) Holter Database [13] and the Malignant Ventricular Arrhythmia Database [13]. 278 samples are healthy rhythms. 105 samples are provided from the physical database collected via Microsoft Band. 173 are collected from the Physiobank Normal Sinus Rhythm Dataset [4] and the MIT-BIH Normal Sinus Rhythm Database [12].

4.6.2 Scikit-Learn LogitBoost Class

To support this paper, a publicly available Python class is developed to support the use of multiple, heterogeneous classifiers [25]. This class is to be used with the 0.16.1 version of the Scikit-Learn Python library [24]. The base implementation is built upon the AdaBoost implementation already available in the Scikit-learn library. This allows the inclusion of the four weak learners: Naive Bayes, SVM, SGD, and Decision Trees. Publicly available via GitHub [25]. Fig. 4.10 shows the implementation detailed by Schapire, R. and Freund, Y. [32]. Fig. 4.11 details the additional function which allows for a custom array of already instantiated classes to be used as weak learners instead of a single classifier. The rest of the implementation is slightly modified to accomodate for this.

```
# Error fraction.
estimator_error = numpy.mean(
  numpy.average(incorrect, weights=sample_weight, axis=0))

# Boost weight using multi-class AdaBoost SAMME alg.
estimator_weight = self.learning_rate * (
  numpy.log((1. - estimator_error) / estimator_error) +
  numpy.log(self.n_classes_ - 1.))

# Only boost the weights if there is another iteration of fitting.
if not iboost == self.n_estimators - 1:
  # Only boost positive weights (logistic loss).
  sample_weight *= numpy.log(1 + numpy.exp(estimator_weight * incorrect *
                              ((sample_weight > 0) |
                               (estimator_weight < 0))))

self.estimator_weights_[iboost] = estimator_weight
self.estimator_errors_[iboost] = estimator_error
```

Figure 4.10: **LogitBoost Class Implementation**:
LogitBoost Class Implementation. This is the inner content of the loop that iterates over weak learner in the estimators array. This follows the implementation outlined by [32]. Publicly available via GitHub [25].

```
def set_estimators(self, estimators):
  self.n_estimators = len(estimators)
  self.estimators_ = numpy.array(estimators)
```

Figure 4.11: **LogitBoost Class Array Implementation**:
LogitBoost Class Array Implementation. The main contribution of this modified class is the ability to perform boosting on a set of weak learners that are not the same type of classifier. This allows the easy inclusion of the four weak learners: Naive Bayes, SVM, SGD, and Decision Trees. Publicly available via GitHub [25].

4.6.3 Experimentation

For each experiment, the data set is randomly divided into a training and testing set, and contains an even distribution of HRV classes. The training and testing splits are 10%-90%, 20-80%, 30-70%, 50-50%, and 70-30%. This will demonstrate the strengths of different machine learning algorithms based on how much data they require to train on before showing success.

The machine learning algorithms that are compared include Support Vector Machine with a linear kernel, Decision Trees, Naive Bayes, Stochastic Gradient Descent, and LogitBoost. Comparisons are done in terms of training rate as well as accuracy. The algorithms are derived form the 0.16.1 version of the Scikit-Learn Python library [24]. The LogitBoost implementation uses Support Vector Machine with a linear kernel, Decision Trees, Naive Bayes, and Stochastic Gradient Descent as weak learners. Unless stated otherwise, the LogitBoost classifier uses five of each weak learner (a total of 20). The implementations is available via GitHub [25].

The weak learners and comparative classifiers in this chapter use the default settings from the Scikit-Learn Python library [24] unless stated otherwise. The SVM classifier uses a linear kernel with the C parameter set to 0.1 and gamma set to 0.0 (defaulting to 1 / number of features). The SGD has an alpha parameter of 0.0001, no penalty parameter set, and a hinge loss function. The Naive Bayes implementation has an alpha parameter of 1.0 with uniform prior probabilities. The Decision Tree implementation has no max features set, no max depth set, and the minimum number of samples for an internal node split set to two. The implementations are available via GitHub [25].

4.7 Accuracy Results

4.7.1 Accuracy of LogitBoost vs. SVM

The first two tables of this section compare the LogitBoost classifier accuracy to the SVM classifier accuracy. Table 4.5 details the 70-30% training vs. test split of a SVM vs. LogitBoost accuracy. The results in this table are collected using all of the datasets. Table 4.6 details the 70-30% training vs. test split of a SVM vs. LogitBoost accuracy. The results in this table are collected without using the physical activity dataset collected via Microsoft Band.

Run 70%-30% (186/92)	SVM Accuracy	LogitBoost Accuracy
0	0.9194	0.9731
1	0.9355	0.9570
2	0.9463	0.9463
3	0.9516	0.9731
4	0.9462	0.9677
5	0.9354	0.9624
Average	**0.9391**	**0.9633**
Std. Dev.	**0.0116**	**0.0104**

Table 4.5: **SVM vs. LogitBoost Classification Results With Physical Activities**:
SVM vs. LogitBoost Classification Results With Physical Activities. The classification accuracies across 278 samples. This uses a 70-30 training vs. testing split, training on 186 samples and testing on 92.

Run 70%-30% (121/52)	SVM Accuracy	LogitBoost Accuracy
0	0.8620	0.9310
1	0.8965	0.9137
2	0.9051	0.9137
3	0.9137	0.9482
4	0.8879	0.9224
Average	**0.8931**	**0.9259**
Std. Dev.	**0.0198**	**0.0144**

Table 4.6: **SVM vs. LogitBoost Classification Results Without Physical Activities**:
SVM vs. LogitBoost Classification Results Without Physical Activities. The classification accuracies across 173 samples. This uses a 70%-30% training vs. testing split, training on 121 samples and testing on 52.

4.7.2 Accuracy vs. Training-Testing Split

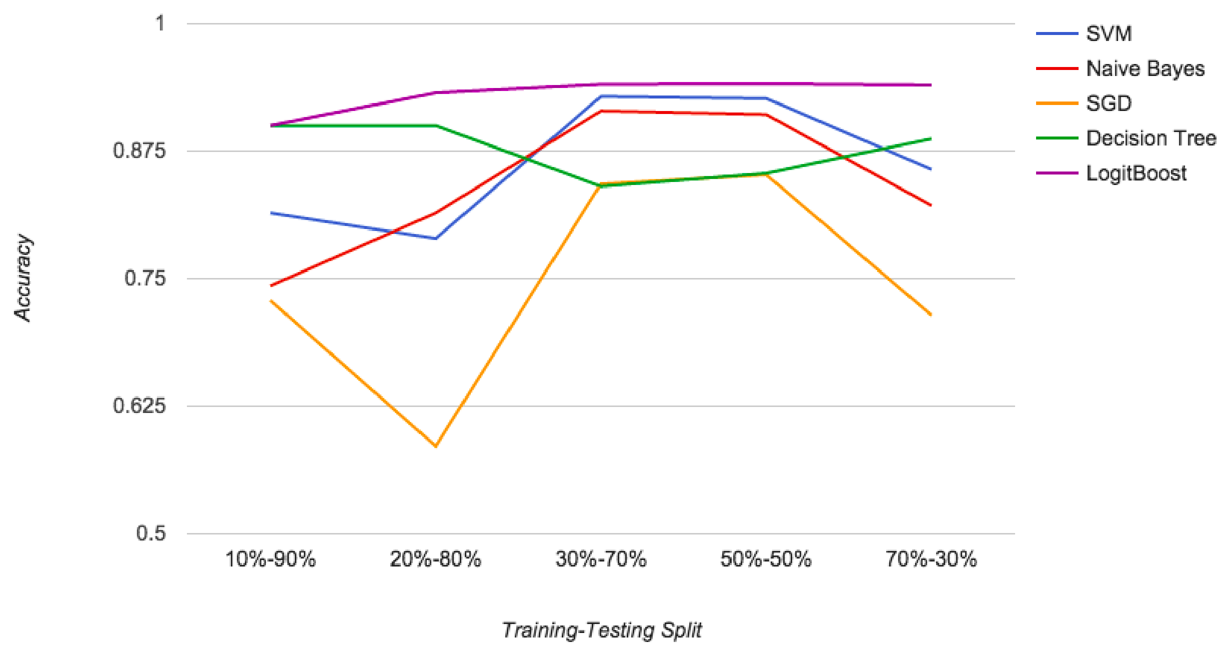

Figure 4.12: **Learning Rates of Each Classifier**:
Learning Rates of Each Classifier. These were collected from the average of five runs for each classifier at each training-testing percentages. The LogitBoost algorithm provides a consitently higher accuracy score than all other classifiers.

The LogitBoost algorithm is able to use the strength of the Decision Tree classifier at low training samples and the strength of the SVM classifier at high training samples to provide a consistent classification accuracy at all training percentages. Fig. 4.12 demonstrates the LogitBoost algorithm's ability to maintain a high classification accuracy, despite the varying accuracies of the weak learners that it uses. The same comparisons are made without the use of the physical activity dataset in Fig. 4.13

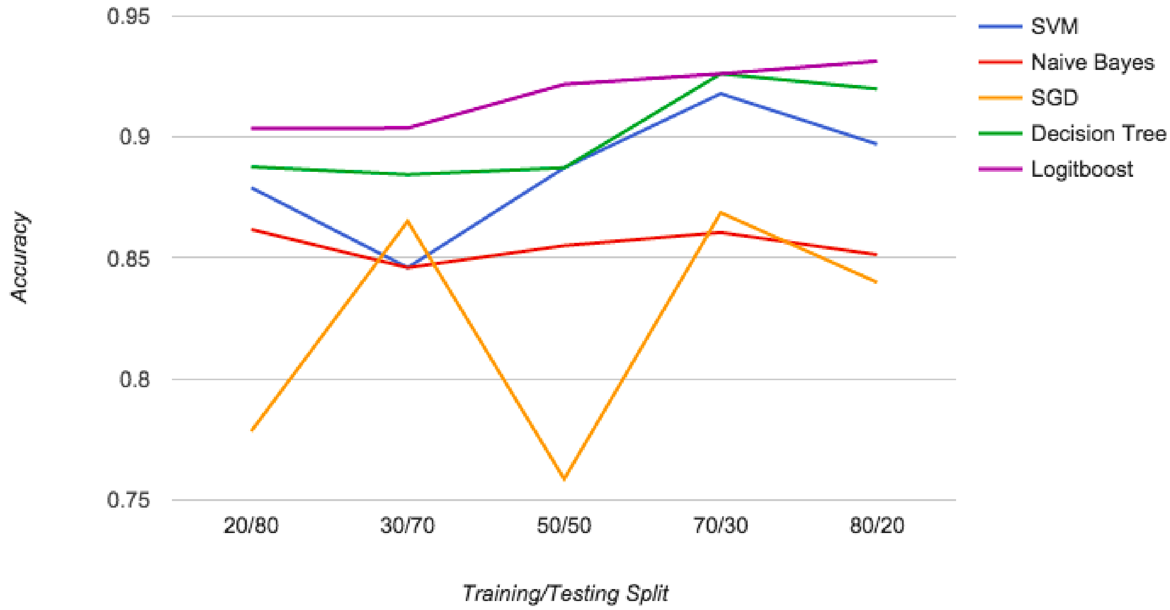

Figure 4.13: **Learning Rates of Each Classifier Without Physical Activities**: Learning Rates of Each Classifier Without Physical Activities. These were collected from the average of five runs for each classifier at each training-testing percentages. The LogitBoost algorithm provides a consitently higher accuracy score than all other classifiers. This figure specifically does not use the physical activity dataset, so there are only a total of 173 samples.

The rest of the section is devoted to the individual runs of each classifier at different training-testing percentages. Table 4.12 and Table 4.13 feature the accuracies without the physical activity dataset. Table 4.7, Table 4.8, Table 4.9, Table 4.10, and Table 4.11 show the accuracies with the physical activity dataset.

70%-30% (186-92)	SVM	Naive Bayes	SGD	Decision Tree	LogitBoost
0	0.8709	0.8064	0.8494	0.8817	0.9408
1	0.8118	0.8279	0.8064	0.8709	0.9462
2	0.8548	0.8279	0.3763	0.8978	0.9408
3	0.8978	0.8172	0.6935	0.8924	0.9301
4	0.8494	0.8279	0.8440	0.8924	0.9408
Average	**0.8569**	**0.8215**	**0.7139**	**0.8870**	**0.9397**
Std. Dev.	**0.0314**	**0.0096**	**0.1988**	**0.0107**	**0.0058**

Table 4.7: **70%-30% Classification Results With Physical Activities**: 70%-30% Classification Results With Physical Activities. The classification accuracies across 278 samples. This uses a 70-30 training vs. testing split, training on 186 samples and testing on 92.

50%-50% (139-139)	SVM	Naive Bayes	SGD	Decision Tree	LogitBoost
0	0.9568	0.8992	0.9136	0.7769	0.9208
1	0.9424	0.8920	0.8920	0.7985	0.9352
2	0.8920	0.9424	0.9064	0.8992	0.9568
3	0.9280	0.9064	0.6330	0.8848	0.9280
4	0.9136	0.9136	0.9136	0.9064	0.9640
Average	**0.9266**	**0.9107**	**0.8517**	**0.8532**	**0.9410**
Std. Dev.	**0.0251**	**0.0194**	**0.1225**	**0.0607**	**0.0186**

Table 4.8: **50%-50% Classification Results With Physical Activities**: 50%-50% Classification Results With Physical Activities. The classification accuracies across 278 samples. This uses a 50-50 training vs. testing split, training on 139 samples and testing on 139.

30%-70% (84-194)	SVM	Naive Bayes	SGD	Decision Tree	LogitBoost
0	0.9166	0.9047	0.9166	0.8333	0.9166
1	0.9642	0.9047	0.8809	0.8095	0.9523
2	0.9047	0.9285	0.8333	0.8690	0.9166
3	0.9285	0.8928	0.6309	0.8690	0.9404
4	0.9285	0.9404	0.9523	0.8214	0.9761
Average	**0.9285**	**0.9142**	**0.8428**	**0.8404**	**0.9404**
Std. Dev.	**0.0222**	**0.0195**	**0.1263**	**0.0273**	**0.0252**

Table 4.9: **30%-70% Classification Results With Physical Activities**: 30%-70% Classification Results With Physical Activities. The classification accuracies across 278 samples. This uses a 30-70 training vs. testing split, training on 84 samples and testing on 194.

20%-80% (56-222)	SVM	Naive Bayes	SGD	Decision Tree	LogitBoost
0	0.7321	0.7857	0.3928	0.8928	0.9107
1	0.8392	0.7857	0.8571	0.9642	0.9464
2	0.8928	0.8750	0.6607	0.8571	0.9642
3	0.8035	0.7500	0.4642	0.8750	0.9107
4	0.6785	0.8750	0.5535	0.9107	0.9285
Average	**0.7892**	**0.8142**	**0.5857**	**0.9000**	**0.9321**
Std. Dev.	**0.0850**	**0.0573**	**0.1818**	**0.0410**	**0.0232**

Table 4.10: **20%-80% Classification Results With Physical Activities**: 20%-80% Classification Results With Physical Activities. The classification accuracies across 278 samples. This uses a 20-80 training vs. testing split, training on 56 samples and testing on 222.

10%-90% (28-250)	SVM	Naive Bayes	SGD	Decision Tree	LogitBoost
0	0.7857	0.6785	0.8214	0.8571	0.8571
1	0.8571	0.7500	0.7142	0.9642	0.9642
2	0.7857	0.8214	0.8214	0.8571	0.8928
3	0.8571	0.8571	0.6785	0.9285	0.8928
4	0.7857	0.6071	0.6071	0.8928	0.8928
Average	**0.8142**	**0.7428**	**0.7285**	**0.9000**	**0.9000**
Std. Dev.	**0.0391**	**0.1022**	**0.0931**	**0.0465**	**0.0391**

Table 4.11: **10%-90% Classification Results With Physical Activities**: 10%-90% Classification Results With Physical Activities. The classification accuracies across 278 samples. This uses a 10-90 training vs. testing split, training on 28 samples and testing on 250.

30%-70% (52-121)	SVM	Naive Bayes	SGD	Decision Tree	LogitBoost
0	0.9423	0.8846	0.8846	0.9423	0.9423
1	0.8846	0.8653	0.7692	0.8846	0.9423
2	0.8653	0.7884	0.8076	0.7307	0.9038
3	0.8653	0.8846	0.8846	0.8653	0.9230
4	0.8461	0.8461	0.8653	0.8846	0.9038
Average	**0.8807**	**0.8538**	**0.8423**	**0.8615**	**0.9230**
Std. Dev.	**0.0370**	**0.0398**	**0.0516**	**0.0786**	**0.0192**

Table 4.12: **30%-70% Classification Results Without Physical Activities**: 30%-70% Classification Results Without Physical Activities. The classification accuracies across 173 samples. This uses a 30-70 training vs. testing split, training on 52 samples and testing on 121.

70%-30% (121-52)	SVM	Naive Bayes	SGD	Decision Tree	LogitBoost
0	0.9426	0.8442	0.8442	0.9016	0.9344
1	0.9344	0.8524	0.7786	0.9180	0.9590
2	0.9180	0.8606	0.8688	0.9016	0.9016
3	0.8934	0.8524	0.8114	0.9016	0.9344
4	0.9180	0.8606	0.8688	0.9262	0.9262
Average	**0.9213**	**0.8540**	**0.8344**	**0.9098**	**0.9311**
Std. Dev.	**00.0188**	**0.0068**	**0.0390**	**0.0115**	**0.0205**

Table 4.13: **70%-30% Classification Results Without Physical Activities**: 70%-30% Classification Results Without Physical Activities. The classification accuracies across 173 samples. This uses a 70-30 training vs. testing split, training on 121 samples and testing on 52.

4.8 Precision, Recall, and F-score Evaluation

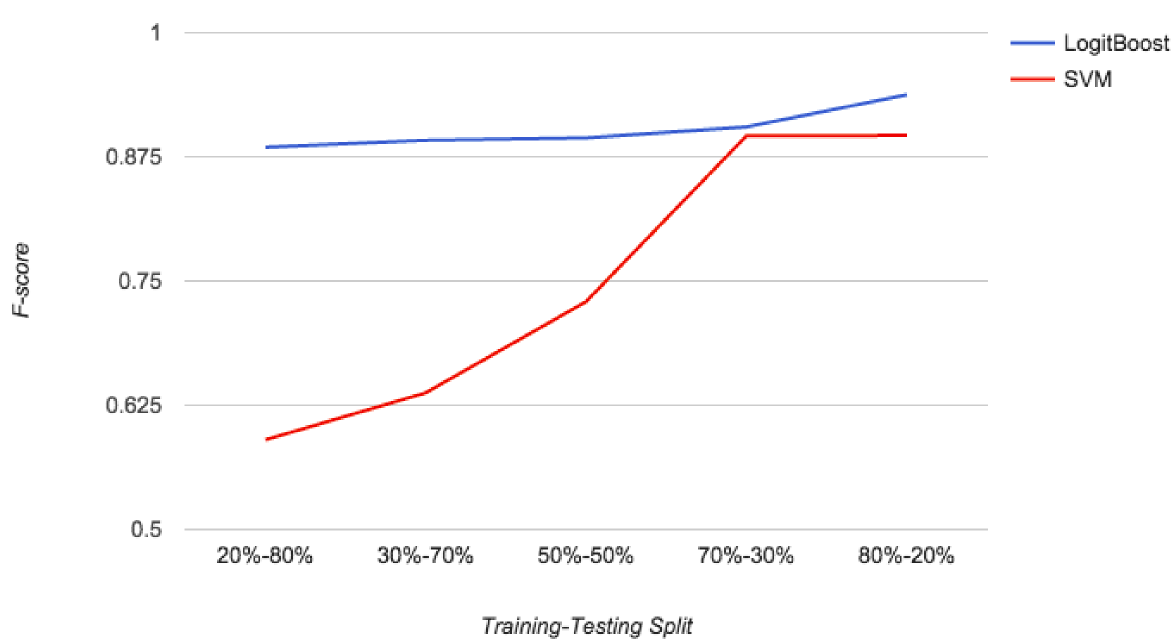

Figure 4.14: **LogitBoost vs. SVM F-score**:
LogitBoost vs. SVM F-score. This is across each training-testing split with a total of 278 samples.

Fig. 4.15 and Fig. 4.16 show the comparative precision and recall respectively between the SVM and LogitBoost classifier. The LogitBoost classifier has much higher precision, but comparable recall to the SVM classifier. Table 4.14 details the Precision, Recall and F-score of the LogitBoost classifier on multiple training-testing dataset splits accompanied by the accuracy of those splits. Table 4.15 details the Precision, Recall and F-score of the SVM classifier on multiple training-testing dataset splits accompanied by the accuracy of those splits. These values are the average of five runs each. The F-score for the LogitBoost classifier is consist despite the varying number of training samples. The SVM classifier becomes weaker as the number of training samples decreases.

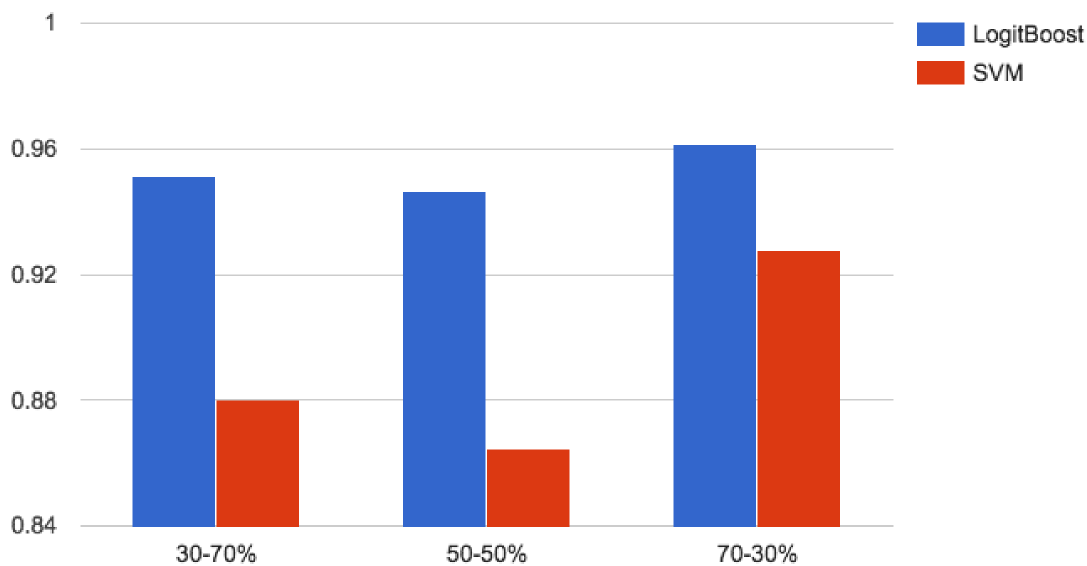

Figure 4.15: **Precision vs. Dataset Split**:
Precision vs. Dataset Split. This chart shows the Precision for each classifier at different training-testing splits. The positive label used in the F-score is the Sudden Cardiac Death class.

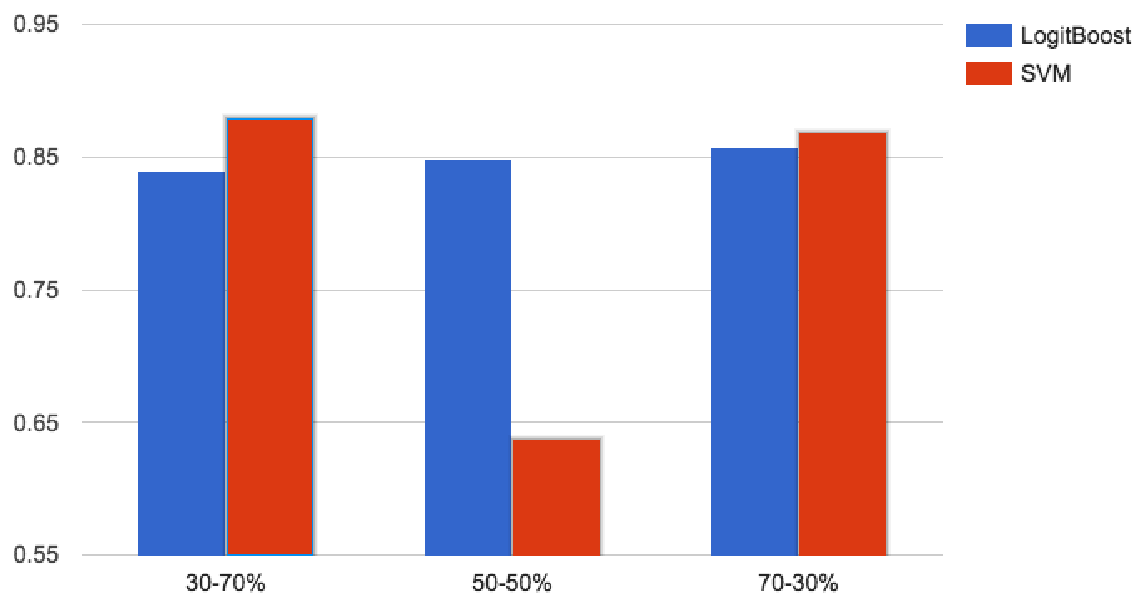

Figure 4.16: **Recall vs. Dataset Split**:
Recall vs. Dataset Split. This chart shows the Recall for each classifier at different training-testing splits. The positive label used in the F-score is the Sudden Cardiac Death class.

Training-Testing Split	Accuracy	Precision	Recall	F-score
30%-70% (84-194)	0.7324	0.9512	0.8402	0.8916
50%-50% (139-139)	0.9280	0.9466	0.8480	0.8941
70%-30% (186-92)	0.9357	0.9617	0.8575	0.9050

Table 4.14: **LogitBoost Accuracy, Precision, Recall, and F-score Results**:
LogitBoost Precision, Recall, and F-score Results

Training-Testing Split	Accuracy	Precision	Recall	F-score
30%-70% (84-194)	0.8307	0.8804	0.8804	0.6374
50%-50% (139-139)	0.8316	0.8647	0.6373	0.7292
70%-30% (186-92)	0.9261	0.9279	0.8686	0.8961

Table 4.15: **SVM Precision, Recall, and F-score Results**:
SVM Precision, Recall, and F-score Results

Fig. 4.14 shows the F-scores for all classifiers. These values are detailed in Table 4.16. Similar to the accuracies, the results for LogitBoost are higher than any other classifier at every training-testing split. The same comparisons are made in Fig. 4.18, but without the use of the physical activity dataset.

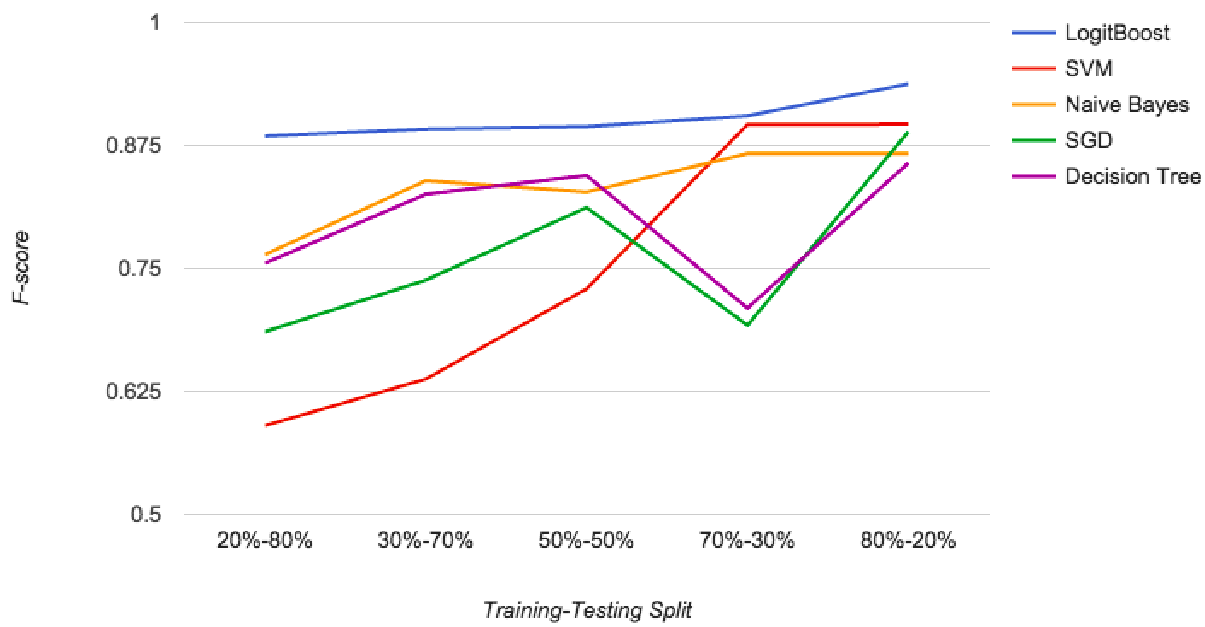

Figure 4.17: **F-Score vs. Dataset Split**:
F-Score vs. Dataset Split. This chart shows the F-score for each classifier at different training-testing splits. The positive label used in the F-score is the Sudden Cardiac Death class.

Training-Testing Split	LogitBoost	SVM	Naive Bayes	SGD	Decision Tree
20%-80%	0.8846	0.5901	0.7640	0.6857	0.7552
30%-70%	0.8916	0.6374	0.8391	0.7380	0.8253
50%-50%	0.8941	0.7292	0.8275	0.8118	0.8444
70%-30%	0.9050	0.8961	0.8666	0.6923	0.7096
80%-20%	0.9375	0.8965	0.8666	0.8888	0.8571

Table 4.16: **F-score Results**:
F-score Results This tables shows the F-score for each classifier at different training-testing splits. The positive label used in the F-score is the Sudden Cardiac Death class.

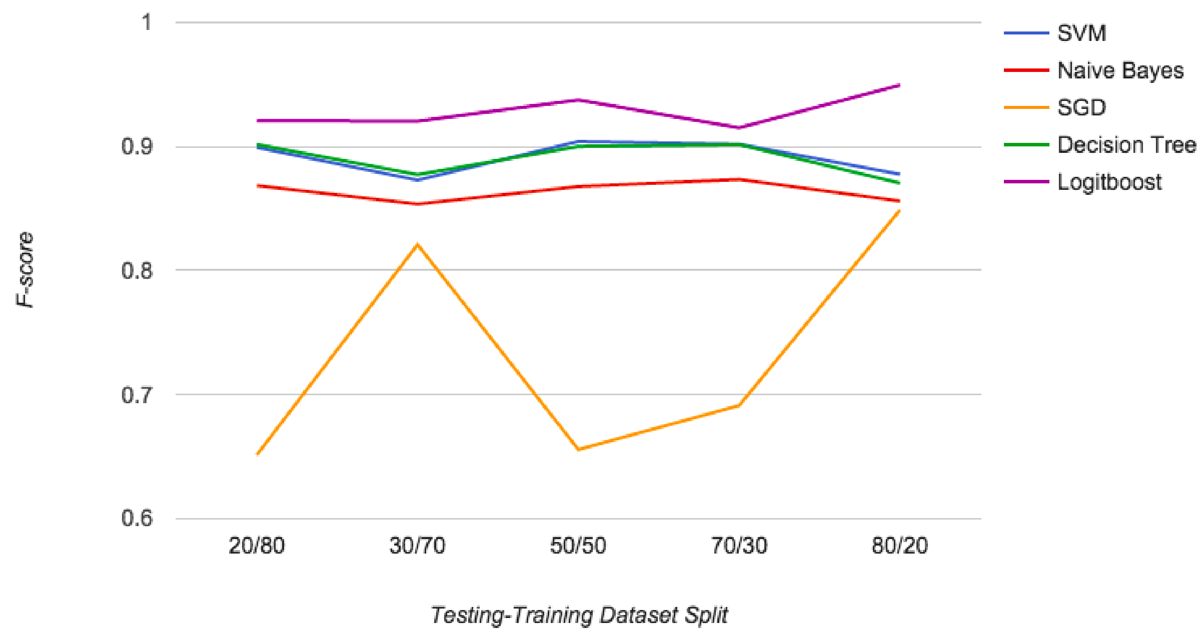

Figure 4.18: **F-score of Each Classifier Without Physical Activities**:
F-score of Each Classifier Without Physical Activities. These were collected from the average of five runs for each classifier at each training-testing percentages. The LogitBoost algorithm provides a consitently higher F-score than all other classifiers. This figure specifically does not use the physical activity dataset, so there are only a total of 173 samples.

Training-Testing Split	LogitBoost	SVM	Naive Bayes	SGD	Decision Tree
20%-80%	0.9208	0.8993	0.8685	0.6512	0.9017
30%-70%	0.9205	0.8729	0.8536	0.8209	0.8774
50%-50%	0.9375	0.9041	0.8677	0.6557	0.9001
70%-30%	0.9151	0.9019	0.8734	0.6912	0.9016
80%-20%	0.9496	0.8776	0.8561	0.8489	0.8705

Table 4.17: **F-score Results without Physical Activity**:
F-score Results without Physical Activity. This tables shows the F-score for each classifier at different training-testing splits. The positive label used in the F-score is the Sudden Cardiac Death class. This table specifically does not use the physical activity dataset, so there are only a total of 173 samples.

4.9 Comparing Different Weak Learner Setups

In the previous sections, it is evident that the SVM and Decision Tree classifiers may be strong weak learners in this case than the Naive Bayes and SGD classifiers. The results of the previously defined LogitBoost classifier (four weak classifiers in multiples of five each) is compared to a LogitBoost classifier with only SVM and Decision Tree weak learners (two weak classifiers with ten instances of each). This new classifier is named the pruned classifier. The weak learners are trained in round robin order.

4.9.1 Accuracy

Figure 4.19: **F-Score of All vs. Pruned LogitBoost Classifier**:
F-Score of All vs. Pruned LogitBoost Classifier.

The accuracy of the pruned classifier is slightly higher in this case, but there is a dip in the 50-50 training-testing split. The all weak learners classifier has accuracies of 0.9241, 0.9280, and 0.9357 for the 30-70, 50-50, and 70-30 splits respectively. The pruned weak learners classifier has accuracies of 0.9366, 0.9425, and 0.9476. There is an increase in accuracy for the pruned classifier, but not a significant increase.

Figure 4.20: **Accuracy of All LogitBoost vs. Single Algorithm LogitBoost**: Accuracy of All LogitBoost vs. Single Algorithm LogitBoost.

Fig. 4.20 compares the new implementation of LogitBoost with multiple weak learner algorithms against the default LogitBoost implementation with a single algorithm (SVM) weak learner.

4.9.2 F-score

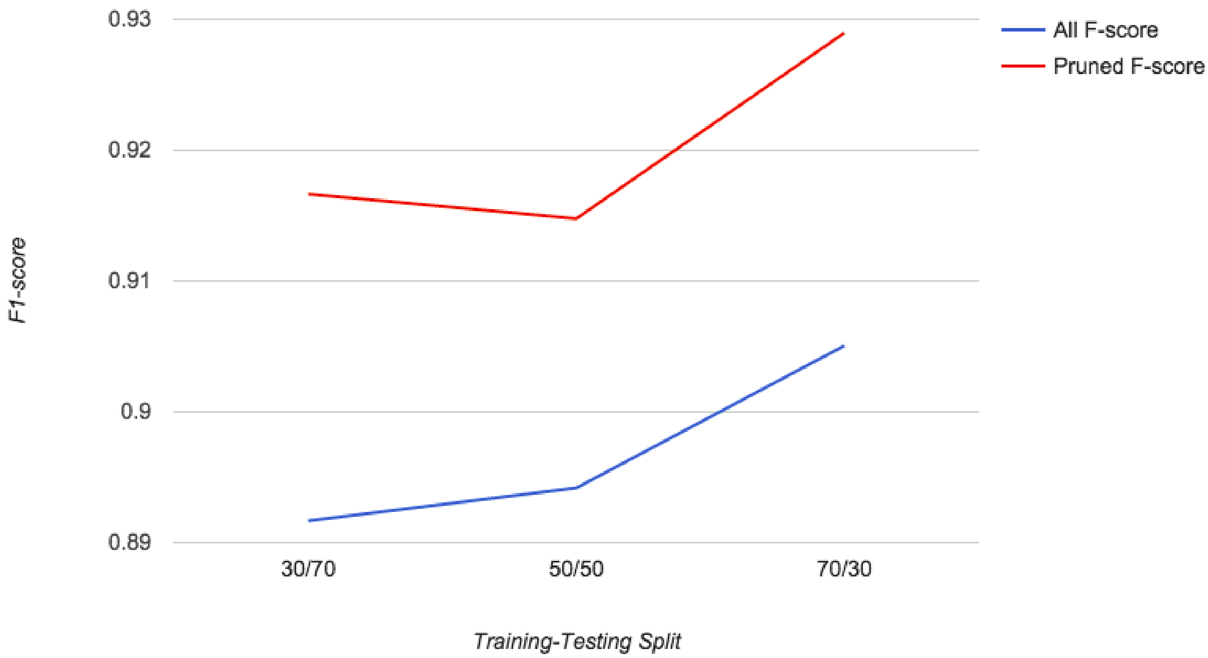

Figure 4.21: **F-Score of All vs. Pruned LogitBoost Classifier**:
F-Score of All vs. Pruned LogitBoost Classifier.

The f-score of the pruned classifier is slightly higher in this case, but there is a dip in the 50-50 training-testing split. The all weak learners classifier has accuracies of 0.9050, 0.8941, and 0.8916 for the 30-70, 50-50, and 70-30 splits respectively. The pruned weak learners classifier has accuracies of 0.9289, 0.9147, and 0.9166. There is an increase in f-score for the pruned classifier, but not a significant increase.

Figure 4.22: **F-Score of All LogitBoost vs. Single Algorithm LogitBoost**: F-Score of All LogitBoost vs. Single Algorithm LogitBoost.

Fig. 4.22 compares the new implementation of LogitBoost with multiple weak learner algorithms against the default LogitBoost implementation with a single algorithm (SVM) weak learner.

4.10 Comparison to Related Work

In Murukesan et al. [23], the MIT-BIH Sudden Cardiac Death database and the MIT-BIH Normal Sinus Rhythm database [12] is the dataset the SVM trains and tests upon. This section makes an apples to apples comparison of the two approaches using the smaller dataset. Fig. 4.23 shows the accuracies for this small dataset. The maximum accuracy for the LogitBoost classifier is 82.50% in these runs and the average accuracy for the SVM classifier is 73.68%. Fig. 4.24 shows the F-scores for this small dataset. The maximum F-score for the LogitBoost classifier is 85.20% in these runs and the average F-score for the SVM classifier is 76.13%.

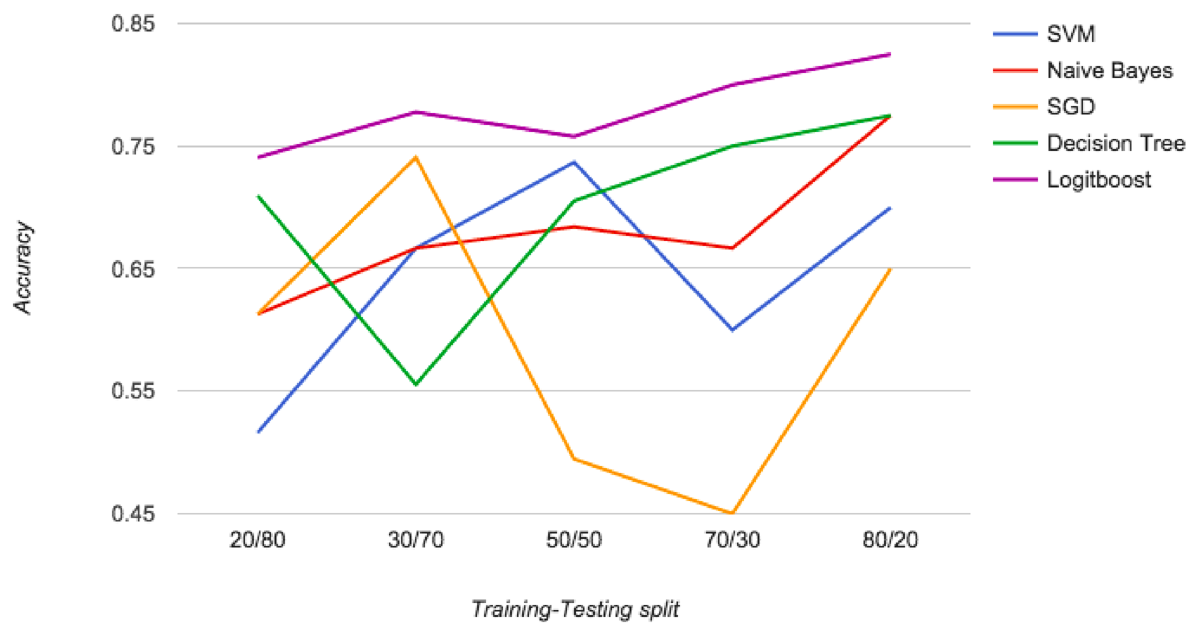

Figure 4.23: **Accuracy vs. Training-Testing Split for NSR and SCD Dataset**:
Accuracy vs. Training-Testing Split for NSR and SCD Dataset. This is the dataset in use by Murukesan et al. [23].

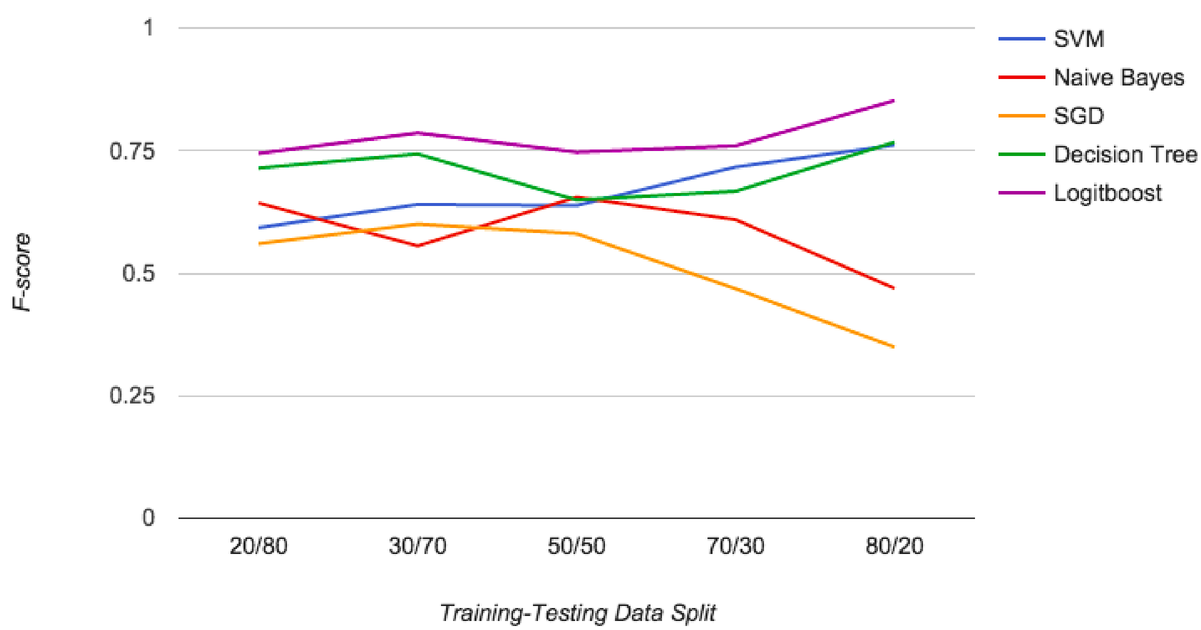

Figure 4.24: **F-Score vs. Training-Testing Split for NSR and SCD Dataset**:
F-Score vs. Training-Testing Split for NSR and SCD Dataset. This is the dataset in use by Murukesan et al. [23].

4.11 Comparison Between Datasets

A comparison between the datasets is made in Fig. 4.25 by accuracy and Fig. 4.26 by F-score. The classifier is slightly better when trained and tested on the larger dataset (278 samples).

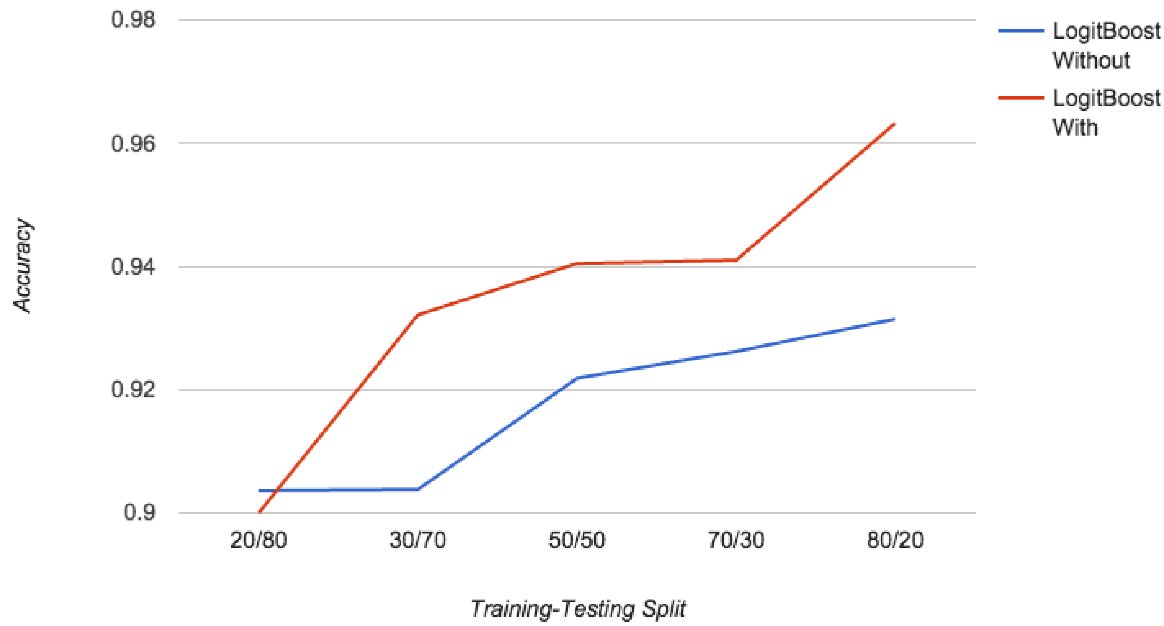

Figure 4.25: **Accuracy of All vs. Without Physical Dataset**:
Accuracy of All vs. Without Physical Dataset. This figure compares the accuracies of the LogitBoost algorithm between the two datasets (278 samples and 173 samples).

Figure 4.26: **F-Score of All vs. Without Physical Dataset**:
F-Score of All vs. Without Physical Dataset. This figure compares the F-scores of the LogitBoost algorithm between the two datasets (278 samples and 173 samples).

Chapter 5

Discussion

This paper presents a new sudden cardiac arrest prediction technique, a LogitBoost classifier implementation for multiple weak learners, a prospective physical activity heart rate dataset, and an Internet of Things solution towards heart rate monitoring and sudden cardiac arrest warning. Using a 70%-30% training-testing split, the work in this paper is able to achieve a 96.33% accuracy with a 0.9375 F-score for the classification of sudden cardiac arrest prediction. Comparably, Murukesan et al. [23] achieve 96.36% accuracy, but their approach uses a smaller dataset and does not include physical activity heart rates. This paper's approach uses HRV derived features on two minute samples of heart rate data. This paper proves that it is possible to classify resting as well as active heart rates against the heart rates of people about to go into sudden cardiac arrest. The inclusion of heart rate data from a wearable device also proves that there is a future for sudden cardiac arrest prediction through wearable devices.

One of the major difficulties with this paper is the collection of data samples. The samples in this paper are a collection of publicly available datasets made available through Physionet [12]. More datasets exist, but are not currently publicly available. More datasets may be made available in the future by hospitals performing the same kind of studies, and the release of these would provide the confidence of a more robust model. Until then, researchers not directly connected with specific institutions have to make due with what is available.

Chapter 6

Future Work

6.1 A More Robust Physical Activity Dataset

The prospective dataset presented in this paper is a small, pilot study to provide a representation of what commercial wearable devices would provide in terms of heart rate data. A more advanced study would have to be done with human subject testing procedures.

6.2 Sudden Cardiac Arrest Data via Wearable Device

The prospective data collected from the Microsoft Band only contains normal rhythms of healthy individuals. It would be beneficial to collect heart rates using the same or a similar wearable device of the onset of sudden cardiac arrest.

6.3 Broader Physical Activity Dataset

The physical activities demonstrated in the prospective physical activity dataset only provide a limited number of activities and on a limited number of people. The Microsoft Band has poor performance in heart rate monitoring on people in intense motion, making the band unreliable in activities such as running, jump-roping, and similar activities. Future work should try to incorporate those activities with more people to create a broader dataset.

6.4 Governmental Approval of such a System

The system proposed in this paper would require some sort of governmental approval in order to see public use. It would be extremely disastrous to have an individual incorrectly assume that this system is the only form of necessary protection from sudden cardiac arrest. This system has the potential to attract many lawsuits and would need governmental and legal protection.

6.5 Outlier Detection System

An outlier detection system could be implemented which would learn the cluster of normal heart rate behaviors for an individual. Anytime the behavior is outside of this cluster, a warning is generated. This would not handle sudden cardiac arrest warning specifically, but would potentially handle a large number of cases related to similar problems.

BIBLIOGRAPHY

[1] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[2] M. Beck. If you're stricken, minutes matter, yet many ignore signs, delay treatment. 2012.

[3] K. P. Birman. Rule-based learning for more accurate ecg analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pages 369–380, 1982.

[4] R. Bousseljot, D. Kreiseler, and A. Schnabel. Nutzung der ekg-signaldatenbank cardiodat der ptb ber das internet, 1995.

[5] M. M. Churpek, T. C. Yuen, M. T. Huber, S. Y. Park, J. B. Hall, and D. P. Edelson. Predicting cardiac arrest on the wards in-hospital cardiac arrest prediction a nested case-control study. *CHEST Journal*, 141(5):1170–1176, 2012.

[6] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[7] Z. Dokur and T. Ölmez. Ecg beat classification by a novel hybrid neural network. *Computer methods and programs in biomedicine*, 66(2):167–181, 2001.

[8] E. Ebrahimzadeh, M. Pooyan, and A. Bijar. A novel approach to predict sudden cardiac death (scd) using nonlinear and time-frequency analyses from hrv signals. *PloS one*, 9(2):e81896, 2014.

[9] A. S. Go, D. Mozaffarian, V. L. Roger, E. J. Benjamin, J. D. Berry, W. B. Borden, D. M. Bravata, S. Dai, E. S. Ford, C. S. Fox, et al. Heart disease and stroke statistics–2013 update: a report from the american heart association. *Circulation*, 127(1):e6, 2013.

[10] K. Goh, J. Lavanya, Y. Kim, E. Tan, and C. Soh. A pda-based ecg beat detector for home cardiac care. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 375–378. IEEE, 2006.

[11] A. L. Goldberger. *Clinical electrocardiography: a simplified approach*. Elsevier Health Sciences, 2012.

[12] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. Components of a new research resource for complex physiologic signals, 2000.

[13] S. Greenwald. Development and analysis of a ventricular fibrillation detector, 1986.

[14] H. A. Guvenir, S. Acar, G. Demiroz, and A. Cekin. A supervised machine learning algorithm for arrhythmia analysis. In *Computers in Cardiology 1997*, pages 433–436. IEEE, 1997.

[15] N. P. Hughes, L. Tarassenko, and S. J. Roberts. Markov models for automated ecg interval analysis. In *Advances in Neural Information Processing Systems*, page None, 2003.

[16] S. Jadhav, S. Nalbalwar, and A. Ghatol. Artificial neural network based cardiac arrhythmia classification using ecg signal data. In *Electronics and Information Engineering (ICEIE), 2010 International Conference On*, volume 1, pages V1–228. IEEE, 2010.

[17] Z. Jin, J. Oresko, S. Huang, and A. C. Cheng. Hearttogo: a personalized medicine technology for cardiovascular disease prevention and detection. In *Life Science Systems and Applications Workshop, 2009. LiSSA 2009. IEEE/NIH*, pages 80–83. IEEE, 2009.

[18] Z. Jin, Y. Sun, and A. C. Cheng. Predicting cardiovascular disease from real-time electrocardiographic monitoring: An adaptive machine learning approach on a cell phone. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 6889–6892. IEEE, 2009.

[19] S. Joo, K.-J. Choi, and S.-J. Huh. Prediction of ventricular tachycardia by a neural network using parameters of heart rate variability. In *Computing in Cardiology, 2010*, pages 585–588, Sept 2010.

[20] J. Kim, H. S. Shin, K. Shin, and M. Lee. Robust algorithm for arrhythmia classification in ecg using extreme learning machine. *Biomed Eng Online*, 8:31, 2009.

[21] S. Kotsiantis, I. Zaharakis, and P. Pinetlas. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 26:159–190, 2006.

[22] E. Kuronen et al. Epic sensors in electrocardiogram measurement. 2013.

[23] L. Murukesan, M. Murugappan, and M. Iqbal. Sudden cardiac death prediction using ecg signal derivative (heart rate variability): A review. In *Signal Processing and its Applications (CSPA), 2013 IEEE 9th International Colloquium on*, pages 269–274, March 2013.

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

[25] L. Plewa. Zagreus. `https://github.com/luke-plewa/zagreus`, 2015.

[26] M. Plewa. Evidence of repolarization abnormalities in previous electrocardiograms of victims of non-traumatic sudden cardiac death. 2014.

[27] K. Polat, B. Akdemir, and S. Güneş. Computer aided diagnosis of ecg data on the least square support vector machine. *Digital Signal Processing*, 18(1):25–32, 2008.

[28] K. Polat and S. Güneş. Detection of ecg arrhythmia using a differential expert system approach based on principal component analysis and least square support vector machine. *Applied Mathematics and Computation*, 186(1):898–906, 2007.

[29] K. Polat, S. Şahan, and S. Güneş. A new method to medical diagnosis: Artificial immune recognition system (airs) with fuzzy weighted pre-processing

and application to ecg arrhythmia. *Expert Systems with Applications*, 31(2):264–269, 2006.

[30] A. Reiss and D. Stricker. Introducing a new benchmarked dataset for activity monitoring. *The 16th IEEE International Symposium on Wearable Computers (ISWC)*, 16, 2012.

[31] J. Rodriguez, A. Goni, and A. Illarramendi. Real-time classification of ecgs on a pda. *Information Technology in Biomedicine, IEEE Transactions on*, 9(1):23–34, 2005.

[32] R. Schapire and Y. Freund. Boosting: Foundations and algorithms. 2014.

[33] T. Soman and P. O. Bobbie. Classification of arrhythmia using machine learning techniques. *WSEAS Transactions on computers*, 4(6):548–552, 2005.

[34] S. Somanchi, S. Adhikari, A. Lin, E. Eneva, and R. Ghani. Early code blue prediction using patient medical records.

[35] F. Wang, R. Quiniou, G. Carrault, and M.-O. Cordier. Learning structural knowledge from the ecg. In *Medical Data Analysis*, pages 288–294. Springer, 2001.

[36] Z.-D. Zhao and Y.-Q. Chen. A new method for removal of baseline wander and power line interference in ecg signals. In *Machine Learning and Cybernetics, 2006 International Conference on*, pages 4342–4347. IEEE, 2006.