#### ABSTRACT

#### A NOVEL IOT-BASED WIRELESS SYSTEM TO MONITOR HEART RATE

#### by Mohamed Assem ElSaadany

The Internet of Things (IoT) with smartphone technologies has vast applications in solving the problems of heart diseases in patients needing care. With wireless sensors and smart devices, remote monitoring can identify the real-time physical status of heart patients under many normal physical activities. In this research, we designed and developed an IoT system to monitor heart rate. A wearable ECG sensor is used to monitor the pulse patterns and smartphone built-in sensors— accelerometer and GPS—- are used to measure the body acceleration and location information of the user. Data is collected via a Low Energy (LE) Bluetooth communication network between the ECG sensor and smartphone for further processing in the phone. Experimentation and verification is conducted on a number of test subjects with different test scenarios including sitting, walking, jogging, and running. The sensor outputs, with heart rate acquired from the experiment, are presented in this research. An algorithm for activity classification is implemented and discussed in the results.

#### A NOVEL IOT-BASED WIRELESS SYSTEM TO MONITOR HEART RATE

Thesis

Submitted to the

Faculty of Miami University

in partial fulfillment of

the requirements for the degree of

Master of Science in Computational Electrical and Computer Engineering

by

Mohamed Assem ElSaadany

Miami University

Oxford, Ohio

2017

Advisor: Dr. Donald Ucci

Co-Advisor: Dr. AKM Jahangir Alam Majumder

Reader: Dr. Gokhan Sahin

©2017 Mohamed Assem ElSaadany

This thesis titled

#### A NOVEL IOT-BASED WIRELESS SYSTEM TO MONITOR HEART RATE

by

Mohamed Assem ElSaadany

has been approved for publication by

College of Engineering and Computing

and

Department of Electrical and Computer Engineering

Dr. Donald Ucci

Dr. AKM Jahangir Alam Majumder

Dr. Gokhan Sahin

# Contents

1	Intro	oduction	n	1
2	Mot	ivation a	and Related Work	5
	2.1	Problem	m definition	5
	2.2	Related	l work	6
3	Syst	em Arcl	hitecture	9
	3.1	Hardwa	are	10
		3.1.1	Heart Monitor Sensor	10
		3.1.2	Arduino UNO	11
		3.1.3	Bluetooth Shield	11
		3.1.4	Connected Devices	12
	3.2	Softwa	re	13
		3.2.1	Arduino	13
		3.2.2	Android <sup>TM</sup>	14
4	Data	a Collec	tion	19
	4.1	ECG D	Data Collection	19
		4.1.1	Using IoT device	19
		4.1.2	Previous Stored datasets	21
	4.2	Accele	rometer Data Collection	21

5	Resu	lts and Evaluation	23
	5.1	Body Acceleration	23
	5.2	ECG Signals	28
	5.3	Algorithm	32
		5.3.1 Naive Bayes	33
6	Cond	clusion and Future Work	39
	6.1	Conclusion	39
	6.2	Future Work	40
Bil	oliogr	raphy	41

# **List of Tables**

2.1	Comparison of our system and other related work	8
4.1	ECG collection details	20
5.1	Calculated energies from the accelerometer data for different scenarios	27
5.2	ECG features average values	32
5.3	Classes count and percentage	35
5.4	Mean and standard deviation for features in different classes	35
5.5	Confusion matrix	36

# **List of Figures**

1.1	One period of ECG signal	2
1.2	ECG signal for STEMI heart attack	3
3.1	Architecture of proposed system	9
3.2	AD8232 Heart Monitor	10
3.3	Arduino UNO	11
3.4	Adafruit Bluefruit LE Shield	11
3.5	Prototype IoT device	12
3.6	Portable wearable Device	12
3.7	Login/Registration Page	14
3.8	Main Application Page	15
3.9	Bluetooth Connection Page	16
3.10	ECG Page	17
3.11	Accelerometer Page	18
3.12	ECG and Acceleration Graphs Page	18
4.1	ECG Data Collection using IoT	20
4.2	IoT device usage guide	21
5.1	Accelerometer Data	24
5.2	Acceleration for different activities	24
5.3	Acceleration for different activities after smoothing	25
5.4	ECG signal for different activities	28

5.5	Processed ECG signal	29
5.6	Processed ECG signal	30
5.7	Length Transform to find QRS duration	31
5.8	Flow chart for analyzing data	32
5.9	Plot showing the regions of each class using Naive Bayes	36
5.10	Box plot of heart rates vs activities	37
5.11	Box plot of horizontal energy vs activities	38

#### DEDICATION

I dedicate this thesis to my parents and their support throughout my entire life all the way to getting my master degree. I am very grateful to them, my mother for her constant support and prayers, and my father for his continuous encouragement and financial support which helped me achieve my target.

#### ACKNOWLEDGMENTS

I would like to express my deep appreciation to my advisor, Prof. Donald Ucci, for giving me all the support needed to achieve my goals and getting my Master degree. His continued advices during my study program steered me in the right direction and his guidance during my work assignments and responsibilities made me a successful teaching assistant. His early help in selecting my research topic was a main reason for completing the thesis according to my area of interest. I would like to acknowledge Dr. Jahangir for his support as my co-advisor for my Master thesis. His tremendous support during the details of the research topic was the cornerstone of getting me to complete my work requirements. His push for me to attend competitive conference and publishing our work has contributed to my professional career. His continued suggestions and comments was a main reason for taking the time out to serve as a committee members for my thesis. His constructive feedback both during my proposal stage and my exam stage allowed me to improve the quality of my thesis. Lastly, I would like to thank my friends in the Computational Electrical and Computer Department who have been very supportive and helpful. They were there for me during studying, working, and also in collecting data for my thesis work.

## Chapter 1

## Introduction

The Internet of Things (IoT) is a massive and growing virtual universe with a very high demand over the past few years caused by the increasing number of electronic devices and sensors connected to the Internet and to each other. Applications for this technology can be found in manufacturing, industrial systems, medical and healthcare, transportation, as well as others. According to K. A. Hua [1], the Internet is expected to have over 28 billion devices connected by 2020. Even though the number seems massive, it is reasonable taking into account the number of applications and uses considered. It is not just the number of smartphones, computers, and sensors used in well-known industries but, also, sensors used to monitor livestock animals such as cows and sheep. Since animals are very valuable, there is a need to track and monitor their movement patterns and behavior. Using deep learning algorithms, farmers can detect any abnormality in their assets, especially for large numbers of animals.

Using IoT in the medical and healthcare field is of a great interest and is proving to be efficient and useful for monitoring a patient's health conditions using smart devices and sensors. In recent years the medical sensors and devices are getting smaller in size, more accurate and precise, and it is now easier to create communications modules connecting these sensors and devices. Collecting large sets of data and performing an appropriate analysis on this data is the new ongoing challenge since a person's health is critical, especially in regards to heart related conditions. Heart disease is the leading cause of deaths for men and women worldwide [2] <sup>1</sup>, with over 600,000 deaths a year in the US alone. It is estimated that these deaths cost the US an estimate of \$207 billion including the cost of health-care services, medications, and lost productivity. One of the most common heart diseases is heart attacks which occurs when the heart muscle does not get enough oxygen supply from its intervening blood vessels. This often happens when the coronary arteries, which are the passages that allow the blood flow to the heart muscle, are narrowed or blocked due to the buildup of fats and other substances known as plaque. Heart attack symptoms do not occur in the same severity, however, in all individuals. It is common for people experiencing heart attacks to be found clutching their chest or arm, losing balance, and even falling to the ground. About 15% of people having a heart attack will die from it and half of that happens outside a hospital. This presents a serious problem that needs to be solved.

Anyone is susceptible to a heart attack, however, most heart attacks occur to middle-aged or older individuals with the average age of the first heart attack to be around 70 for men and 73 for women [3]. In order to address this problem, our research focuses more on the elderly age group. We study the symptoms experienced when people get heart attacks and design a system that helps those patients monitor their heart conditions, and predict any abnormality.

The proposed system works with the Electro-Cardio-Gram (ECG or EKG) signal collected from a patient body in real-time. ECG data collection devices can be seen almost at every hospital and physicians' offices. This signal is an electrical representation of the heart's activity and it carries much information that can be used to check for the heart's condition to see if there is any abnormality. An ECG signal contains three waves as seen in Fig. 1.1 P, QRS complex, and T.



Figure 1.1: One period of ECG signal

<sup>&</sup>lt;sup>1</sup>Most of references regarding heart disease facts and statistics come from this site https://www.cdc.gov/heartdisease/

Our research is concerned with the heart rate, as well as the ST segment, these two parameters are affected in the case of ST-segment Elevation Myocardial Infarction (STEMI) Heart Attacks. STEMI is a serious form of heart attack that requires immediate attention and occurs when the coronary artery is completely blocked. In this form of heart attack, the heart rate often increases, blood pressure elevates, respiratory rate increases, and coughing and wheezing [4]. Figure 1.2 is an example showing one period of an ECG signal with an elevated ST.



Figure 1.2: ECG signal for STEMI heart attack

With the recent advances in mobile technologies and the decreasing cost of the smartphones with respect to their growing capabilities and power, smartphones proved to be the best portable and most efficient devices because of their ubiquity and adaptability to a wide variety of applications.

#### For our system to work the following challenges are addressed. We need to

- Build an integrated IoT system
- Collect real-time data of ECG and accelerometer
- Detect abnormality in heart rate as well as body balance
- Develop an algorithm to predict heart attacks
- Create an alert notification system

Our system is based on a smartphone that is connected with IoT devices to gather and send data to and from an Android<sup>TM</sup> application that is used for analysis and display. Our system monitors a patient's real-time ECG signal and uses the smartphone's accelerometer to obtain data regarding the patient's body movement. Our system analyzes and evaluates specific parameters to detect any abnormality in the ECG; subsequently, it checks body movement. After these processes are completed, our algorithm would then make decisions as to when a heart attack might occur. In case of emergencies, our application will make aware the user and inform a physician or caregiver.

## Chapter 2

## **Motivation and Related Work**

### 2.1 **Problem definition**

People can get heart attacks anytime and anywhere whether sitting, walking, driving, eating, or performing any Activities of Daily Living (ADL). This results in many negative consequences for heart attack victims, such as loss of balance as well as reduction of independence since they will have a fear of being alone in case something happens and they need assistance. This will also have the consequence of them being isolated and unsocial due to the decline in the health conditions. Our system aims to collect continuous data to monitor the heart patterns as well as other information gathered by the built-in smartphone sensors, enabling the system's users to live their lives more normally without having to constantly worry about their health.

Many existing systems monitor health conditions or detect problems and alert users. Our research and system is unique because it examines a wide variety of health attributes. First, given sensor's data and ECG signals, our system uses an algorithm we developed to predict heart attacks rather than just detect them. This is done by studying the symptoms and patterns prior to heart attacks of actual patients that were diagnosed with the condition. We also challenge ourselves to develop and implement a Low Energy (LE) real-time abnormality detection system using an LE Bluetooth device for power consumption and by decreasing computation complexity, in addition to having a sleep mode option on the application.

### 2.2 Related work

This section examines the different work and research that have been done on related systems. The goal of our system aims to outperform these systems by using acquired data, analyzing the gathered information, then, from the results develop a novel and unique system that yields better results.

The system prototype, proposed by R. S. B. Rosli aims to detect heart problems for patients in an attempt to provide more mobile means to monitor heart conditions [5]. A pulse sensor is used to obtain the ECG signal, which is then processed by an Arduino <sup>TM</sup> to determine the heart rate. If it is out of the normal range, the user is alerted via a text message. Global System for Mobile (GSM) module was used for communication between the phone and Arduino<sup>TM</sup>. First, the user specifies the minimum and maximum heart rate providing a "normal" range. Then, it transmits this range via text to Arduino<sup>TM</sup> to set a threshold. A sensor keeps reading data from the user until the text message "Stop" is sent. The user will be alerted with a text message only when the heart rate falls out of range.

A smartphone based system was developed by G. Wolgast to monitor and detect heart failures in real-time [6]. A single lead heart monitor sensor was used with an Arduino<sup>TM</sup> microcontroller, LE Bluetooth device and antenna to send and receive data to and from the Android<sup>TM</sup> application created on the smartphone. This research focuses on the communications aspects to exchange data in the most effective way with minimum noise to obtain a clear signal for analysis and display on the application. The Android<sup>TM</sup> application has two modes: one for displaying the ECG signal and one to show the signal's strength. The authors claim that the system was able to transmit a noise free signal to the smartphone with high accuracy and that only further software development is needed for heart failures predictions by looking at the ST-segment.

A. Dewan and M. Sharma proposed a hybrid algorithm that would be able to extract unknown patterns and other information related to heart disease based on previous heart disease database records [7]. The main focus of this proposed system is to use data mining techniques and utilize the already stored at the hospital to make decisions regarding predictions of heart diseases.

One proposed system by M. Koshti uses IoT based devices to monitor heart condition [8]. Using body worn sensors to collect data, the proposed system then sends these data using ZigBee<sup>TM</sup>

modules to a remote location and the received data is monitored and graphed on a computer using serial connection. Such system makes it difficult for patients to track real-time data because, unlike smartphones that are highly portable, computers are not so portable. Our proposed system has a unique architecture as it embeds wireless IoT devices for heart monitoring and prediction of heart attacks and links them to a smartphone.

The work done by D. S. Medhekar focuses on the use of Bayesian classification and uses the Naive Bayes algorithm to predict heart attacks given an unknown data sample as input. This works on two phases. First the training phase in which a database from the Cleveland clinic was used to create models and classes. The datasets contains information about heart disease diagnosis and 303 observations on 14 parameters such age, gender, chest pain, blood pressure, and other ECG parameters. Next, there is the testing phase where a prediction probabilistic algorithm classifies the input as high, average, or low risk, after a training sequence is used [9].

P. Jambhulkar and V. Baporikar proposed a system that uses Data Mining (DM) and Wireless Sensor Network (WSN) to predict heart disease, by collecting a large set of data from the Cleveland heart disease databases and, comparing different algorithms and classification techniques, that are run on the collected datasets [10]. This research found Naive Bayes to be the most accurate technique for this process. It outperforms more sophisticated classifications in predicting outcomes based on large set of data using probability theory. The authors then proposed WSN to collect multiple patients in real-time and then transmit that data to a central system. Then MATLAB <sup>TM</sup> will be used to extract necessary features from the datasets for heart disease predictions using a DM technique via MATALB<sup>TM</sup>. The proposed system does not utilize the smartphone and, thus, it is not a system for all users. Rather it is a system for a hospital or a Cardiologist to use in monitoring patients.

A smartphone based system by M. Raihan was developed to predict heart attacks using clinical data and data mining techniques. An Android<sup>TM</sup> application was created on the smartphone to collect the user's information. No sensors or real time data are used as an input to the application. A user is asked to enter information such as age, previous heart conditions, or other health related data in the form of answering questions. These entered data are then evaluated and interpreted

by the application to calculate the risk of having a heart. A score is calculated for every datum provided by the user. Then using a Chi-Square correlation, Fisher's exact test, and probabilistic techniques a risk score is calculated and displayed for the user [11].

Another proposed work concerning the prediction of a heart attack is conducted by the University of Turku, Finland [12]. Their system would predict heart attacks solely using the gyroscope sensor on smartphone. The smartphone is placed on the patient's chest and, using a sensitive embedded sensor, they are able to detect the micro-movements when the blood flow to the heart muscle is disturbed. The proposed system is claimed to have an accuracy of 70%-90% if the patient's baseline is recorded before the heart attack.

Table 2.1 provides a simple comparison between our system and other proposed related work previously discussed which shows that our system is unique.

	IoT	Has monitoring	Has detection	Prediction	Low Energy	Real-time
	system	capability	capability	system	system	system
Rosli' 16	N	N	Y	N	Ν	Y
Wolgast' 16	Y	Y	Y	N	Y	Y
Dewan' 15	N	N	N	Y	Ν	N
Koshti' 15	Y	Y	N	N	N	Y
Medhekar' 13	N	N	N	Y	N	N
Jambhulkar' 15	Y	Y	Y	Y	N	Y
Raihan' 16	N	N	N	Y	N	N
Our System	Y	Y	Y	Y	Y	Y

Table 2.1: Comparison of our system and other related work

## Chapter 3

## **System Architecture**

The architecture is a key factor differentiating our proposed system from other previously proposed systems that use body-worn sensors to monitor the heart rate and, provide ECG data. In our system, we have an ECG sensor connected to the body as shown in Fig. 3.1 ECG data is sent via Bluetooth to our smartphone application. Accelerometer data is also collected internally from the smartphone and both sensors' data are stored in a database. After analysis, if there is any abnormality in heart pattern, a voice alert is set to alert the user and inform an attending doctor or caregiver, while using the smartphone's GPS to show the location information of the user. The following sections describe the system implementation.



Figure 3.1: Architecture of proposed system

### 3.1 Hardware

Our hardware architecture consists of two IoT devices a smartphone and ECG data acquisition device with the communication module. The sensor is soldered to the microcontroller and the LE Bluetooth module. A 9V battery is used to power the device, making it portable and easy to use.

#### 3.1.1 Heart Monitor Sensor

The AD8232 is a 4mm x 4mm single lead heart rate monitor that is operational at 3.3 volts, between -40 and 85 degrees Celsius and with low current supply of about 170  $\mu$ A. The circuit is designed to extract, amplify, and filter small bio-potential signals in the presence of noisy conditions, such as those created by motion or remote electrode placement. It has an integrated filter coupled with an amplifier that produces a high gain signal (G=100) and a high pass filter in a single stage, thus allowing the AD8232 to recover quickly so that it takes valid measurements after connecting the electrodes to the subject, thus saving cost and space. Three electrodes are connected to the body as shown in Fig. 3.2, with a 3.5mm jack for the Biomedical Pad Connection on the other end connected to the heart rate monitor through which analog readings are read continuously and sent to the output pin. On the board, we also have an LED indicator that will simulate the rhythm of heart beat [13].



Figure 3.2: AD8232 Heart Monitor

#### 3.1.2 Arduino UNO

An Arduino UNO is a powerful micro-controller with 6 analog input pins and 14 digital pins that can be used for either input or output. It works with a High Performance, Low Power Atmel®AVR® 8-Bit Microcontroller chip that is operational at 1.8 to 5.5V and between -40 and 85 degrees Celsius. The power consumption for the Arduino UNO at 1MHz, 1.8V, 25 Celsius is 0.2 mA on Active Mode, 0.1  $\mu$ A on Power-down Mode and 0.75  $\mu$ A on Power-save Mode. The microcontroller can be powered by a battery, USB cable, or AC-to-DC adapter [14].



Figure 3.3: Arduino UNO

### 3.1.3 Bluetooth Shield

An Adafruit Bluefruit LE Shield<sup>TM</sup> is a LE Bluetooth Shield for the Arduino UNO that operates on 5V. It is interfaced with the Arduino using hardware SPI pins (MISO, MOSI, SCK) that can be soldered with the micro-controller, in addition to a chip select line (default D8), interrupt line (default D7), and reset (default D4) [15].

	DA God Dig	tal 1/0	Digit	al 1/0
• 💥 🔘 👥			$\bullet \bullet \bullet \bullet$	0000
hluofruit				3 2 1 0
S S S S S S S			RD RS	•
	00000	0000	Šaaš	00
	00000	2000		
	00000	00000		
CONN	00000	0000		
2	00000	0000		
	00000	0000	0000	
	00000	0000	0000	00 00
CO Distance	00000	0000	0000	
	00000	0000	0000	00 5 <u>2</u> 0
	00000	0000	0000	00 EH.
	00000	0000		•• <u>H</u> a
5	00000	0000	0000	00 24
	0000	0000	0000	000
RST	0000	30 B GND	<b>2000</b>	006
and the second second		0000	0 0 0	0000
	10	3v 5v Gnd	Vin 0 1	23150
	000	0000	• • •	0000

Figure 3.4: Adafruit Bluefruit LE Shield

### **3.1.4** Connected Devices

Reliable connection between the different components is achieved by soldering the devices together instead of having connections using wires that can be loosened by body movement, impact with other objects. Soldering also creates a neat, single device that can be easily used as in Fig. 3.5.



Figure 3.5: Prototype IoT device

In order to have a more user friendly device, a leather pouch was handmade to contain the unit. Straps were also added, as seen in Fig. 3.6, that enable a user to attach the device around the arm.



(a) Envelope in leather pouch



(b) Adding straps to wrap around the arm

Figure 3.6: Portable wearable Device

### 3.2 Software

To receive and analyze data from the hardware in our system, an Android<sup>TM</sup> heart rate collector interface in the smartphone is used. We developed a Bluetooth communication system that is capable of transmitting data from the pulse to the smartphone. On receiving inputs from the accelerometer (in the directions of the x-, y-, and z-axes) and ECG sensors, the system processes the data to identify an abnormality in the heart rate or pulse. To process the embedded sensors' data, the communication module performs two software tasks: one for the Arduino and the other for the smartphone. The Arduino, is programmed to read an analog signal from the ECG sensor and creates a data packet converting the signal into a digital form. Subsequently, the Arduino will transmit those packets to the phone as a response to the data sending request. It will also manage the Bluetooth communication by coordinating with the LE Bluetooth device. The smartphone application can communicate with a Bluetooth device to collect sensor data by a polling request. The sensor data will be saved to the database and analyzed by parsing a packet to calculate the real ECG signals' value from the sensor and the threshold value of an individual subject. To install our software on the smartphone, code file must be run on an Android Studio, Eclipse, or any IDE that supports Java, while connecting the phone serially with the personal computer.

#### 3.2.1 Arduino

The software component on the Arduino device does two tasks by interfacing with two hardware components connected to the microcontroller. These are the Bluetooth shield and the heart monitor sensor. Code is written to read the analog input from the heart rate monitor chip and convert it to a discrete floating point number between 0 and 1 that represents the voltage; thus, it acts as Analog-to-Digital Converter (ADC). Another code identifies and setups the pins used to interface the LE Bluetooth Shield with the Arduino allowing it to send data through the Bluetooth device.

### 3.2.2 Android<sup>TM</sup>

Android<sup>TM</sup> application software was written on Android Studio using Java and XML. This is the front end of the system that acts as the user interface that allows for interaction with our IoT device. The following subsections describe our application.

#### Login/Sign up

Figure 3.7 (a) shows the first page of our Android<sup>TM</sup> application. This procedure allows the user to either log into our system or create an account. For storing information, we use the smartphone's database along with a Cloud database such as Microsoft Azure Cloud. In the future, this will allow for privacy and confidentiality of user information. When a user enters information and presses the "LOGIN" button, information is checked in the database and, if valid, then application takes the user to the main page. When the user presses the "SIGN UP" button, s/he is prompted with the window shown in Fig. 3.7 (b) to enter the required information. Upon pressing the "SUBMIT" button at the end of that page, the information is sent to the database for storage.



Figure 3.7: Login/Registration Page

#### Main page

On logging-in, the user will be directed to the home page shown in Fig. 3.8. The "SEARCH" tab takes the user to another page that lists the available Bluetooth devices nearby, as well as any devices paired with the smartphone. Pressing "USER INFORMATION" retrieves the information for that particular user from our database for display. The "ECG DATA" tab allows the user to access the real-time graphs of the ECG and shows the heart rate. The "ACCELEROMETER DATA" tab allows the user to access the real-time accelerometer data showing both the numerical values and the real-time graphs. The "COMBINED GRAPHS" displays, for the user, the real-time graphs for both sensors' data. Finally, the "DECISION MAKING" tab includes the status of the user's health conditions and whether or not everything looks normal, or if a critical condition exists and the user needs immediate attention.



Figure 3.8: Main Application Page

#### **Connect to IoT device**

Figure 3.9 shows the available nearby Bluetooth devices as well as the paired devices. Our application searches for our device by name and, if not found, after 120 seconds that is a default setting, then the search process can start again by simply pulling down the page to refresh. If the device was found, then we can initiate pairing by clicking the device name on the nearby devices list. If the device is discovered and paired, then clicking on the device name in the paired list will open a connection with the device to start receiving data and the application will redirect the user back to the main page once connected successfully. An important feature of our system is that the application is designed to work only with our specific device and will not connect to any other Bluetooth device.



Figure 3.9: Bluetooth Connection Page

#### **Display ECG**

Figure 3.10 allows the user to monitor a real-time graph of the ECG signal with the y-axis as voltage and x-axis time in seconds. The numerical value of the beats/min will be displayed for the user with an alert message, if the heart rate is too high or too low.



Figure 3.10: ECG Page

#### **Display Acceleration**

Figure 3.11 shows the two pages that appear for the user when the user presses on the "AC-CELEROMETER DATA" tab. First, window (a) shows the real-time numerical values for the three axes x, y and z. When the user presses on "DISPLAY GRAPHS", window (b) will appear to display the real-time graphs of the three axes, with the y-axis as the acceleration and x-axis the time in seconds. With the phone in the user's pocket, those graphs tell us the direction of the body movement.



Figure 3.11: Accelerometer Page

#### **Display Combined Graphs**

Figure 3.12 shows the real-time graphs of both sensors' used. The top graph is showing the ECG signal while the bottom one is a real-time plot of the average of the three axes of the accelerometer and also displays the numerical value of the average. The average can be used instead of looking at each axis separately in an attempt to reduce the computation complexity. Note that both pages containing the ECG signal on the application cannot be accessed unless connected to the IoT device.



Figure 3.12: ECG and Acceleration Graphs Page

## **Chapter 4**

## **Data Collection**

ECG and Accelerometer data are to be collected for a number of patients, different age groups, and genders. Data is also collected for different activities for healthy test subjects. This allows us to look at the different patterns of the ECG signals and body acceleration data. The following subsections show the methods by which these data are collected.

### 4.1 ECG Data Collection

#### 4.1.1 Using IoT device

Figure 4.1 shows the flow for ECG data is collected using the IoT device. First, when the electrodes that are attached to the body read the analog signal, this signal passes to the ECG acquisition device that has a built in amplifier and a filter. Furthermore, a software based amplifier and filter could be implemented to enhance the signal even more. The signal is passed to the Arduino used as an ADC, only then, sent to the Android <sup>TM</sup> mobile application using a LE Bluetooth device to be later graphed and analyzed on the smartphone.



Figure 4.1: ECG Data Collection using IoT

The above architecture was initially used to collect data for a number of test subjects as seen in Table 4.1, none of which were diagnosed with heart problems.

Table 4.1:	ECG	collection	details

	Height	Weight	Age	Testing scenarios
3 Females	5'5-5'7	120-180	20-34	Sitting ,Walking
5 Males	5'5-6'1	128-185	23-26	Sitting, Walking, Jogging/Exercising

Figure 4.2 shows how a user can use our IoT device. First, the user powers the device using the battery cable; then s/he wraps the straps around his or her arm, attaches the red lead to the right part of the stomach, the black lead to the right part of the chest, and the blue lead to the left part of the chest.



Figure 4.2: IoT device usage guide

#### 4.1.2 Previous Stored datasets

In the future, ECG data can be obtained from hospital databases after taking the necessary permissions since the hospital has large sets of data for healthy and non healthy test subjects diagnosed with heart disease. Large datasets like the ones at the hospital are helpful in testing our prediction algorithms offline. In addition, datasets can be obtained from databases of Cleveland, Hungary, Switzerland, and the VA Long Beach that are provided by UCI Machine Learning Repository [16]. Many different DM techniques and machine learning algorithms can be used to help create decisions based on archived datasets.

### 4.2 Accelerometer Data Collection

An accelerometer is available on any smartphone with data for the three axes. We assume that the smartphone is in the user's pocket. Data is collected internally, then graphed in real-time to give a good visual representation for the user. The numerical values can be easily displayed. Data obtained for each axis include the gravity component. So, to obtain the actual body acceleration, the gravity component is first removed prior to data representation. Acceleration data will only be considered and used when using our IoT device to collect ECG data, since it is unlikely that the stored ECG datasets at the hospital would have accelerometer data with them as well. We note that y-axis points upward, the x-axis to the side, and the z-axis is perpendicular to the smartphone surface. From this, we are able to determine the direction of the body movement.

## Chapter 5

## **Results and Evaluation**

Data was collected multiple times for different test subjects with a duration of about 1 minute for each time. An analysis, using MATLAB<sup>TM</sup>, is performed for easier signal processing. Once acceptable results are obtained and the algorithm is complete, it is then implemented for the mobile application to be used on real-time data received from the sensors. Note that all data presented and analyzed in this chapter are collected using our devices and not previously stored data.

During analysis a window of five seconds is used at a time for signal processing of both ECG and Accelerometer data. After the first five seconds the window moves by one second which was determined by trial and error to be the smallest possible window to achieve reaching a conclusion in a reasonable amount of time without losing important information. In our system, that means in analyzing a 1 minute data we would have 56 windows each with a five second range.

### 5.1 Body Acceleration

Figure 5.1 shows sample data of the accelerometer that was collected in real-time. Data is graphed for the three axes as well as the resultant value. Visually, the graph shows abrupt fluctuations when someone is moving suddenly or is a steady periodic signal when someone is walking.

To better understand of how accelerometer data might change and how those changes can relate to the body movement, we collected accelerometer data over a period of 1 minute for a healthy test



Figure 5.1: Accelerometer Data

subject while s/he was performing different activities. For clear representation Fig. 5.2 shows the graphs of the data for only 25 seconds time frame with sampling rate of 10 points/second.



Figure 5.2: Acceleration for different activities

Average acceleration for the three axes x,y and z was used in the above graph. In the first graph, while sitting, the acceleration is almost 0. Any fluctuations are due to the sensor being extremely sensitive to movement.

In the second graph, when walking, the acceleration is higher, since, two out of the three axes change significantly (x and z). We note that the graph is not consistent, since the person walking may slow down, speed up, change direction, or pause for sometime.

The third set of data was taken while jogging and, as observed, the data appear consistent and periodic; that means that the jogger was maintaining a steady pace the whole time. The average acceleration here appears to be highest due to the fact that all three axes may be changing in the process. With steady consistent graphs like these, we can determine the number of steps taken in that activity.

The last set of data was taken while exercising and the nature of the exercise here was jumping while squatting so the only axis that would be changing here is the y-axis. Since the exercise is periodic, and the movement is only in the vertical direction the graph appears periodic. The frequency is smaller in this instance because the jogging pace is higher than the jumping pace.

Figure 5.3 shows the same data sets but after applying a smoothing filter to remove the high frequency components, which allows for a cleaner graph.



Figure 5.3: Acceleration for different activities after smoothing

The signal's energy is also useful in determining the direction of the body movements, hence we calculate the horizontal and vertical energy. So, for example, in the jumping scenario the vertical energy will be greater than the horizontal energy while, when walking or falling to the ground, the horizontal energy will be greater than the vertical energy.

To obtain the energies, we:

• First, assume that we have the raw accelerometer data stored in the vector shown in Eq. (1)

$$\vec{A}_0(t) = [\vec{A}_{0x}(t), \vec{A}_{0y}(t), \vec{A}_{0z}(t)]'$$
(1)

• Obtain the gravity vector by calculating the bias of the acceleration vector. Bias is found by taking the average of N acceleration vectors where N is the total number of data instances as shown in Eq. (2)

$$\vec{B}(t) = \frac{1}{N} \sum_{t=1}^{N} \vec{A}_0(t)$$
(2)

• Calculate the two tilt angles for the bias vector as follows in Eqs. (3) and (4)

r

$$\theta_1 = \arctan(\frac{B_y}{B_z}) \tag{3}$$

•

and

$$\theta_2 = \arctan(\frac{B_x}{B_y \sin(\theta_1) + B_z \cos(\theta_1)}) \tag{4}$$

• Calculate the tilt compensated acceleration vector by multiplying the raw vector with a rotation matrix as follows in Eq. (5)

$$\vec{A}_{1}(t) = \begin{bmatrix} \cos(\theta_{2}) & -\sin(\theta_{1})\sin(\theta_{2}) & -\cos(\theta_{1})\sin(\theta_{2}) \\ 0 & \cos(\theta_{1}) & -\sin(\theta_{1}) \\ \sin(\theta_{2}) & \sin(\theta_{1})\cos(\theta_{2}) & \cos(\theta_{1})\cos(\theta_{2}) \end{bmatrix} \times \vec{A}_{0}(t)$$
(5)

• Remove the gravity vector from the accelerometer data by calculating the bias of the vertical component of the acceleration and removing it as in Eq. (6)

$$\vec{A}_{2}(t) = [\vec{A}_{1x}(t), \vec{A}_{1y}(t), (\vec{A}_{1z}(t) - \frac{1}{N} \sum_{t=1}^{N} \vec{A}_{1z}(t))]'$$
(6)

• Calculate the magnitude of the horizontal and vertical acceleration as follows in Eqs. (7) and (8). The reason for this calculation is that there is no other vector bias that can be used to correct another axis, so we cannot distinguish between lateral and forward acceleration or between pitch and roll.

$$A_h(t) = \sqrt{(A_{2x}(t))^2 + (A_{2y}(t))^2}$$
(7)

and

$$A_{\nu}(t) = A_{1z}(t) - \frac{1}{N} \sum_{t=1}^{N} \vec{A}_{1z}(t)$$
(8)

• Finally, calculate the accelerometer energy using Eqs. (9) and (10)

$$e_{va} = \int_{t=t_0}^{t=T+t_0} [A_v] dt$$
(9)

and

$$e_{ha} = \int_{t=t_0}^{t=T+t_0} [A_h] dt$$
 (10)

To see how the signal energy can vary depending on the activity, Eqs. (1) - (10) were used on four different activities multiple times. Table 5.1 shows the average energy calculated for each case.

Table 5.1: Calculated energies from the accelerometer data for different scenarios

	Sitting	Walking	Jogging	Running
Horizontal Energy	0.04172	2.46236	6.9583	10.6153
Vertical Energy	0.00722	-0.03356	0.02574	-0.0664

As shown in table 5.1 the horizontal energy is almost zero when a person is sitting and increases

as the user moves with faster speed in the horizontal direction. The vertical energy is close to zero in all cases; however, it also increases from one scenario to the next as, when running, the body is moving more in the vertical direction than in the walking scenario.

## 5.2 ECG Signals

The collected ECG signal is graphed in real time while extracting the following features for analysis: heart rate, QRS voltages, T wave voltages, RR interval and the QRS wave durations. The extracted features are then used as input to the prediction algorithm to produce an output. The sampling rate used for the ECG signals is 80 points/seconds.

Figure 5.4 shows different sets of data taken for a healthy test subject showing only a time frame of 20 seconds for clear representation; however, the code was run for 1 minute of every set of data.



Figure 5.4: ECG signal for different activities

The first data set show that the signal is steady and consistent with no sudden alterations and that was taken in a sitting resting position. Next, we have the signal when walking and the only major change here is that the heart rate is higher, which is reasonable since the heart speeds up to pump more oxygen to the muscles. The third set was taken while jogging: Here, we can see many features change, namely, the heart rate, voltage peaks and wave durations. The last set of data is for exercising and, again, we can see all the features are different than in the previous signals.

To extract features from the signal dynamically without having to set a threshold manually, we perform operations on the signal such as differencing, filtering, and conversion to the frequency domain, as well as finding peaks and sorting.

Starting with the heart rate Figs. 5.5 and 5.6 show the operations performed to extract this feature. First, we use differencing to flatten out parts of the signal that do not vary greatly in magnitude. The result will then contain only the parts that are significantly higher in magnitude than the other parts of the signal, which typically leaves the QRS peaks more visible.



Figure 5.5: Processed ECG signal

A smoothing operation is then performed on the absolute value of the differenced signal with a window of 20 points. This technique removes the high frequency component in the signal that can be due to noise, a high sampling rate, or the sensitivity of the sensor. Figure 5.5 shows a five second window plot of the original signal and after processing. To get the heart rate, we take the Fourier transform of the processed signal as shown in Fig. 5.6. Then, we find the location of the maximum peak, which is the frequency of occurrence for that peak, indicating the QRS.



Figure 5.6: Processed ECG signal

As shown in Fig. 5.6, the frequency of the maximum peak is seven. This is consistent with Fig. 5.5. Since we had a five second window we calculate the heart rate by multiplying 7 by 12 to get 84 beats/min. This technique was tested on all scenarios and for different test subjects. It proved to be accurate without the need to set a numerical threshold. The RR interval can be obtained from the information we have from that window by dividing the window size, in this case, 5 seconds, by

the number of peaks which is 7 to get 0.714 seconds. Both features extracted fall within the normal range, 60 to 100 beats/min for a heart rate.

To get the QRS and T-wave voltages, first we observe all the peaks in the signal window available, then sort them. Using the number of beats, N, calculated earlier we take the largest N numbers as the QRS values and the second N numbers as the T wave values.

Length Transform [17] used to get the QRS duration. First, we use differencing, then we apply smoothing with a window size of 5. Next, we find all the peaks in the processed signal, sort them, take the largest N numbers where N is the number of beats in that window, and average those N peaks. We create an array of zeros and add ones in the places where the value in the processed signal is higher than the calculated average value of the peaks. To get the QRS duration for a certain beat, we count the consecutive ones and divide by the sampling frequency. Figure 5.7 shows the original signal at the top, the processed signal at the bottom and both bounded by the array of zeros and ones created to highlight the QRS duration.



Figure 5.7: Length Transform to find QRS duration

ECG data was collected for different activities. We did this so that, when writing an algorithm, we could see all the patterns that an ECG signal can have and under what conditions some of the characteristics would be missing or changing. Table 5.2 shows the different ECG features and the average values calculated from the all the data collected using our device.

	Sitting	Walking	Jogging	Running	Units
Avg. Heart rate	70.624	79.27	91.027	121.945	beats/min
Avg. RR interval	0.88	0.775	0.663	0.496	seconds
Avg. QRS peak	0.982	0.943	0.852	0.82	volts
Avg. T-wave peak	0.801	0.799	0.789	0.734	volts
Avg. QRS duration	0.082	0.071	0.085	0.07	seconds

Table 5.2: ECG features average values

### 5.3 Algorithm

Figure 5.8 shows the flow chart analyzing data using our system. The system takes the input, signal processes it, analyzes the data, and then makes decision. The process continues, normally unless



Figure 5.8: Flow chart for analyzing data

the decision notes an abnormality, in which case, it produces an alert message as its output.

We now provide details about the process. First, ECG and accelerometer data were collected multiple times for different users and different activities. The collected data was saved to train our algorithm. After that, the system is ready for testing in real time.

Using our system, we start collecting ECG as well as accelerometer data and then performed some functions to extract the different characteristics and features for both signals. Extracted characteristics and features are then provided as an input to our algorithm that is trained by previously stored datasets. The algorithm can be a single classification technique or a hybrid of multiple techniques.

Based on the output of the algorithm, we make a decision. If the probability of getting a heart attack is high and an abnormality is detected, then we alert the user as well as caregiver; if the data looks normal, then the system keeps reading data from both sensors in real-time.

#### 5.3.1 Naive Bayes

After some research, we decided to use Naive Bayes classification because it produces highly accurate results, it is simple, and easy to implement [18]. Naive Bayes is a probabilistic classifier that is popular in real time prediction applications. It is based on Bayes' Theorem which states that if we have prior knowledge of conditions that might be related to an event, we can describe the probability of that event. In Naive Bayes, we have a set of features that are used as predictors to determine the probability of a new data belonging to certain class, as described by Eq. (11).

$$P(Class_c | Features_x) = \frac{P(Class_c)P(Features_x | Class_c)}{P(Features_x)} .$$
(11)

The algorithm assumes that predictors are independent, which means that the presence of a feature in a class is unrelated to the presence of any other feature. Features usually follow a Guassian distribution, which has two parameters: the mean and the variance. Assuming we have two features to model, which are the heart rate (h) and horizontal energy (e) we calculate the mean and variance for class  $C_1$  as follows in Eqs. (12) and (13):

$$\mu_{(h,c_1)} = \frac{1}{N} \sum_{i=1}^{N} h_i \quad , \qquad \qquad \sigma_{(h,c_1)}^2 = \frac{1}{N} \sum_{i=1}^{N} (h_i - \mu_{(h,c_1)})^2 \quad , \qquad (12)$$

and

$$\mu_{(e,c_1)} = \frac{1}{N} \sum_{i=1}^{N} e_i \quad , \qquad \qquad \sigma_{(e,c_1)}^2 = \frac{1}{N} \sum_{i=1}^{N} (e_i - \mu_{(e,c_1)})^2 \quad . \tag{13}$$

We apply Eqs. (12) and (13) for all classes  $C_1...C_n$  so that, for every class, we have the mean and variance. For a new set of features, h1 and e1, that we want to check to see which class they belong to, we apply the probability density functions in Eqs. (14) and (15):

$$p(h_1|c_1) = \frac{1}{\sqrt{2\pi\sigma_{(h,c_1)}^2}} e^{-\frac{1}{2}} \left(\frac{(h_1 - \mu_{(h,c_1)})}{\sigma_{(h,c_1)}^2}\right) \quad , \tag{14}$$

and

$$p(e_1|c_1) = \frac{1}{\sqrt{2\pi\sigma_{(e,c_1)}^2}} e^{-\frac{1}{2}} \left(\frac{(e_1 - \mu_{(e,c_1)})}{\sigma_{(e,c_1)}^2}\right) .$$
(15)

After calculating the density functions for all the features, we can combine them as the features are independent of each other according to Eq. (16)

$$P(Features_{x}|c_{1}) = p(h_{1}|c_{1})p(e_{1}|c_{1}) .$$
(16)

To satisfy Eq. (11) we need to find the probability of having a certain set of features and the probability of class  $C_1$ , using Eqs. (17) and (18), in that order, where N is the number of classes.

$$P(Features_x) = \sum_{i=1}^{N} P(Features_x | c_i) P(c_i) \quad , \tag{17}$$

and

$$P(c_1) = \frac{Count(c_1)}{N} \quad . \tag{18}$$

Finally, we can calculate the probability of belonging to class  $C_1$  given a set of features as follows using Eqs. (16), (17) and (18) in Eq. (19).

$$P(c_1|Features_x) = \frac{P(c_1)P(Features_x|c_1)}{P(Features_x)} .$$
<sup>(19)</sup>

Now, we repeat the above calculations for all the classes and the class with the highest probability is the output of the algorithm.

To implement a Naive Bayes classifier for our system, we used only two features (heart rate from the ECG and the horizontal energy from the acceleration) and four classes (Sitting, Walking, Jogging, Running). First, we created two tables: one called TrainingSet that contains the data for both features used to train the algorithm and the other ClassNames that contains the names for the classification classes. Table 5.3 shows details regarding the four classes used as input to our algorithm.

Table 5.3: Classes count and percentage

Class name	Count	Percentage
Sitting	560	50%
Walking	336	30%
Jogging	112	10%
Running	112	10%

TrainingSet and ClassNames are then used to train our algorithm. For each class, the algorithm now has a mean and a standard deviation for each of the features shown in Table 5.4.

		Sitting	Walking	Jogging	Running
Heart rate	Mean	70.6714	80.2857	90.4286	126.4286
	Std	11.1329	13.3048	7.7130	8.8121
Horizontal Energy	Mean	0.0393	2.4439	6.2831	10.9118
	Std	0.0829	0.6034	0.8313	1.6845

Table 5.4: Mean and standard deviation for features in different classes

The algorithm is then tested on other collected data for validation. When features are extracted every second, values for the heart rate and the horizontal energy are provided as input to the algorithm which outputs "1" if they belong to a class and "0" if they are not part of a class. Table 5.5 shows the confusion matrix that describes the performance of the Naive Bayes classifier after it

is trained. The confusion matrix shows the number of data for which test case values are known to be true versus other cases, where they can be misclassified. From Table 5.5 we note that we used a total of 1120 data points to train our algorithm, half of those are sitting data as Table 5.3 shows. For sitting class, out of the 560 data points only 8 could be misclassified which presents about 1.4% error. For running class out of 109 data points we have 3 can be misclassified accounting for 2.67% error. As for the other two classes we have no misclassification, this is due to the use of small number of data points, using 10,000 data points to train the algorithm would result in more robust algorithm with error rate higher than zero in all classes.

Table 5.5: Confusion matrix

	Sitting	Walking	Jogging	Running
Sitting	552	8	0	0
Walking	0	336	0	0
Jogging	0	0	112	0
Running	0	0	3	109

The algorithm proved to be highly accurate, as shown in Fig. 5.9. The use of both heart rate and the horizontal energy clearly separates the four different classes with little overlap.



Figure 5.9: Plot showing the regions of each class using Naive Bayes

Figure 5.10 is a box plot for the heart rate and different activities, that shows the minimum, median and maximum values. The x-axis represents the activities, y-axis represents the heart rate in beats per minute. When sitting, the heart rate is between 60 and 85 with a median of about 70 beats/min. As the activity changes with more muscle movement, we note that the minimum and median of the heart rate increases.



Figure 5.10: Box plot of heart rates vs activities

Figure 5.11 shows the box plot of the horizontal energy for all activities. The x-axis represents activities while the y-axis is the energy. First, in sitting scenario, the energy shows to be almost zero, as we get faster movement in the horizontal direction we can see that the minimum, median and the maximum energy values increase.



Figure 5.11: Box plot of horizontal energy vs activities

## Chapter 6

## **Conclusion and Future Work**

### 6.1 Conclusion

Our system is inexpensive, mobile, and validated with real subjects albeit, mostly, healthy subjects. Such a system would be able to save millions of dollars, since medical hospital bills can be very high. So, when a modified smartphone contacts a caregiver or hospital in cases of high risks of heart failures, that eliminates the need to go to the hospital frequently. Saving money for the patient as well as saving hospital resources also allows doctor's time to be spent on more critical events as well as using hospital resources emergency cases.

So far, our system is able to collect data from sensors, send data to the smartphone application using a LE Bluetooth and plot real-time data of all sensors on the smartphone application. For offline analysis on MATLAB<sup>TM</sup> as well as to obtain training data sets, collected data is stored '.txt' files. We use the Fourier transform, differencing and filtering as techniques to extract features. Naive Bayes, which is a probabilistic technique, is used for activity prediction based on the training data sets. We successfully collected data for multiple healthy test subjects doing more than one activity.

#### Challenges completed by our system:

• Built an integrated IoT system

- Collected real-time data of ECG and accelerometer
- Sent data to smartphone via Bluetooth for real-time plotting
- Trained Naive Bayes algorithm using collected data
- Used the trained algorithm for activity prediction

### 6.2 Future Work

In the future, data for real test cases with heart attacks needs to be collected in order to be used in training our algorithm. The algorithm would then be able to predict heart attacks based on the training data sets. We need to create an alert notification system by setting a voice alert to make the user aware of an emergency, while also notifying the nearest hospital, doctor, or caregiver, and use available GPS to provide location information for the patient. Additionally, false alarm rates must be investigated and methods to reduce them need to be developed. under severe noisy conditions.

## **Bibliography**

- K. A. Hua, "Internet of things: Challenges and opportunities for collaborative technologies," in 2016 International Conference on Collaboration Technologies and Systems (CTS), pp. 613 – 614, IEEE, Nov. 2016.
- [2] "Centers for disease control." https://www.cdc.gov/heartdisease/. [Last accessed July 2017].
- [3] E. M. Crimmins, M. D. Hayward, H. Ueda, Y. Saito, and J. K. Kim, "Life with and without heart disease among women and men over 50," *J Women Aging. Author manuscript*, Nov 2010.
- [4] S. Man, C. C. ter Haar, A. C. Maan, M. J. Schalij, and C. A. Swenne, "The dependence of the stemi classification on the position of st-deviation measurement instant relative to the j point," in 2015 Computing in Cardiology Conference (CinC), pp. 837 – 840, IEEE, Sept. 2015.
- [5] R. S. B. Rosli and R. F. Olanrewaju, "Mobile heart rate detection system (moherds) for early warning of potentiallyfatal heart diseases," in 2016 International Conference on Computer and Communication Engineering (ICCCE), pp. 422–427, IEEE, July 2016.
- [6] G. Wolgast, C. Ehrenborg, A. Israelsson, J. Helander, E. Johansson, and H. Månefjord, "Wireless body area network for heart attack detection," in *IEEE Antennas and Propagation Magazine*), vol. 58, pp. 84–92, IEEE, 2016.
- [7] A. Dewan and M. Sharma, "Prediction of heart disease using a hybrid technique in data

mining classification," in 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 704–706, IEEE, Mar. 2015.

- [8] M. Koshti and S. Ganorkar, "Iot based health monitoring system by using raspberrypi and ecg signal," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 5, no. 5, May 2015.
- [9] D. S. Medhekar, M. P. Bote, and S. D. Deshmukh, "Heart disease prediction system using naive bayes," *INTERNATIONAL JOURNAL OF ENHANCED RESEARCH IN SCIENCE TECHNOLOGY & ENGINEERING*, vol. 2, no. 3, Mar 2013.
- [10] P. Jambhulkar and V. Baporikar, "Review on prediction of heart disease using data mining technique with wireless sensor network," *International Journal Of Computer Science And Applications*, vol. 8, no. 1, pp. 55–59, Jan-Mar 2015.
- [11] M. Raihan, S. Mondal, A. More, M. O. F. Sagor, G. Sikder, M. A. Majumder, M. A. A. Manjur, and K. Ghosh, "Smartphone based ischemic heart disease (heart attack) risk prediction using clinical data and data mining approaches, a prototype design," in 2016 19th International Conference on Computer and Information Technology (ICCIT), pp. 299 – 303, IEEE, Dec. 2016.
- [12] "Smartphone app for heart attacks." https://www.dicardiology.com/content/ new-smartphone-app-detects-heart-attacks. [Last accessed July 2017].
- [13] "Heart monitor." http://www.analog.com/media/en/technical-documentation/ data-sheets/AD8232.pdf. [Last accessed July 2017].
- [14] "Microcontroller." https://www.arduino.cc/en/main/arduinoBoardUnof. [Last accessed July 2017].
- [15] "Bluetooth-shield." https://learn.adafruit.com/adafruit-bluefruit-le-shield/ overview. [Last accessed July 2017].

- [16] "Clinical database." https://archive.ics.uci.edu/ml/datasets/Heart+Disease. [Last accessed July 2017].
- [17] E. Pino, L. Ohno–Machado, E. Wiechmann, and D. Curtis, "Real–time ecg algorithms for ambulatory patient monitoring," AMIA Annu Symp Proc, 2005.
- [18] K. A. I. Aboalayon, W. S. Almuhammadi, and M. Faezipour, "A comparison of different machine learning algorithms using single channel eeg signal for classifying human sleep stages," in 2015 Long Island Systems, Applications and Technology, pp. 1–6, IEEE, 2015.