ABSTRACT

A WIRELESS EARLY PREDICTION SYSTEM OF CARDIAC ARREST THROUGH IoT

by Yosuf Amr ElSaadany

The increase in popularity for wearable technologies has opened the door for an Internet of Things (IoT) solution to healthcare. One of the most prevalent healthcare problems today is the poor survival rate of out-of-hospital sudden cardiac arrests. The objective of this study is to present a multisensory system using IoT that can collect physical activity heart rates and body temperatures. For this study, we implemented an embedded sensory system with a Low Energy Bluetooth communication module to discreetly collect electrocardiogram and body temperature data using a smartphone in a common environment. This study introduces the use of signal processing and machine learning techniques for sensor data analytics for sudden cardiac arrest and or heart attack prediction.

A WIRELESS EARLY PREDICTION SYSTEM OF CARDIAC ARREST THROUGH IoT

A 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

Yosuf Amr ElSaadany

Miami University

Oxford, Ohio

2017

Advisor: Dr. Donald Ucci

Co-Advisor: Dr. AKM Jahangir Alam Majumder

Reader: Dr. Yamuna Rajasekhar

This Thesis titled

A WIRELESS EARLY PREDICTION SYSTEM OF
CARDIAC ARREST THROUGH IoT

by

Yosuf Amr 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. Yamuna Rajasekhar

Table of Contents

List of Tables

List of Figures

Dedication

I dedicate this thesis to my hardworking parents who have been very supportive throughout this two-year program. I am very grateful to have such parents, my mother for her constant support and prayers, and my father for his financial support and for sharing his experiences with me which helped me pursue this field of research.

Acknowledgments

I would like to express my gratitude to my advisor, Prof. Donald Ucci, for giving me the support and encouragement to complete my Master’s degree. His time and efforts throughout this two-year program is something I will cherish for the rest of my life. I have been blessed to work as his teaching assistant for some courses and he put me on the right path according to my research interest.

A special thanks to Dr. Jahangir for his support as my co-advisor for my research work. Dr. Jahangir helped me choose a very interesting research topic and I am very proud to have two published papers with him. All my achievements in my research would not have been possible without his continued guidance.

Many thanks to Dr. Rajasekhar for taking the time out of her busy schedule to serve as a committee member. Her constructive criticism during my proposal phase allowed me to improve the quality of the content in my research.

Lastly, I would like to thank all my friends in the Electrical and Computer Engineering Department and at Miami University. Amlaan Shakeel and Mohamed ElSaadany have been very supportive and helpful friends. Together, we were able to help each other by studying together and working as a team on many projects for some courses. I would also like to thank my friends Maddie Bieber and Olivia Lewis for helping me collect data for my research work

Chapter 1

Introduction

Health issues are of utmost importance to human beings. The widespread use of new technologies is paving the way for advanced healthcare systems. Much progress has been made in the medical sciences but, additionally, computing and communication technologies are creating a variety of services that are much closer to actual patients.The availability of wearable devices helps patients monitor and control their health metrics. For instance, a patient can be made aware of the up-to the minute status of his or her condition because of the development and deployment of such devices. Additionally, information can be available to the treating physician or other healthcare provider, which can help to deliver prompt treatment for a condition or even save the life of the user.

1.1 Background

Connected health is becoming a major application of newly developing technologies. The concept of connected health care systems and smart medical devices have enormous potential for companies and for the well-being of people in general. As such, we investigated new technologies that can enable the creation, enhancement, and expansion of connected health systems with the objective of developing a system that can help certain type of patients obtain better awareness about their health status through early medical warning signs.

Before going deeper into the core of the main discussion, we must first shed some light on the technologies that can be used in developing connected health systems. According to ITU-T Global Standards, IoT is a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) “things” based on existing and evolving interoperable information and communication technologies [1].

The goal of the IoT is to enable things to be connected anytime, anyplace, with anything and anyone ideally using any path/network and any service [3]. This goal requires more development in many areas including communications and applications. Many research and development entities are involved in development activities. Cisco defines the Internet of Everything (IoE) as connectivity of people, data, things, and processes in networks of connections [3]; in other words, IoE is a network of computers and devices of all types and sizes, all communicating and sharing information. Cisco [4] also predicts there will be 50 billion devices connected to the Internet by 2020. IoT can be described as a network of networks.

A special dedicated IEEE standard is under development for the architectural framework of the IoT, namely, IEEE P2413 [5]. This standard defines IoT as a system of interconnected people and physical objects along with Information and Communication Technology (ICT) to build, operate, and manage the physical world via smart networking, pervasive data collection, predictive analytics, and optimization [6]. The IoT standard provides a reference model, defines architectural building blocks, and affords development mechanisms for the relevant systems.

As the Internet continues to grow, one of the key enablers is the IPv6 [7] global deployment which supports the ubiquitous addressing of any communicating “smart thing”. It will provide access to billions of smart things allowing new models of IoT interconnection and integration. However, as a result of network expansion, more requirements will be added to network functions, network management, and network composition. IPv6 must enable the interconnection of heterogeneous IoT components together with heterogeneous applications. 6LoWPAN [8] is an optimized version of IPv6 for Low Power Wireless Personal Area Networks. It is basically IPv6 implemented on resource constrained IoT devices.

There are many research areas related to IoT and these are mainly in the areas of networking communication, networking applications, networking security, as well as others. IoT security is one of the main research topics as there is a need to provide security for the growing number of connected devices. For example, there is a need to ensure that IoT devices are only providing information to authorized entities [9]. IoT hardware development has many related research issues as new devices are introduced and many of them are small and have limited battery life. Moreover, the IoT sensor devices must be integrated into the Internet using communication protocols. These protocols must consider the low energy of the sensor battery especially when sensors are deployed in remote locations.

There are many protocols developed and more to be developed that consider the use of Low Energy (LE) for IoT devices. For example, an efficient service announcement and discovery protocol in IP-based ubiquitous sensor networks is proposed [10]. The protocol adopts a fully distributed approach to ensure optimal acquisition times, low energy consumption, and low generated overhead, with timely reaction to topology changes. The protocol is capable of realizing optimal acquisition times with minimal cost in terms of energy and generated overhead, making it suitable for mobile networks.

The Internet Engineering Task Force has done the major standardization work for the Constrained Application Protocol (CoAP) that allow seamless integration of low power devices into the Internet [11]. CoAP can run on most devices that support User Data Protocol and the network architecture that use this protocol is a hot research topic [12-16].

IoT devices use different protocols (Bluetooth, Zigbee, etc.) and different networks (LANs, WANs). Thus, an IoT platform has three building blocks: Things (IoT devices), Gateways (Access Connectivity), and backend networking, along with Cloud Computing as shown in Fig. 1. Cloud Computing is used as an enabling platform that supports IoT based systems to allow connecting the large number of devices and sensors. IoT based healthcare applications can use Cloud Computing platforms to facilitate sensors communication, instead of implementing separate means to have all the sensors communicate directly.

1.2 Problem Description

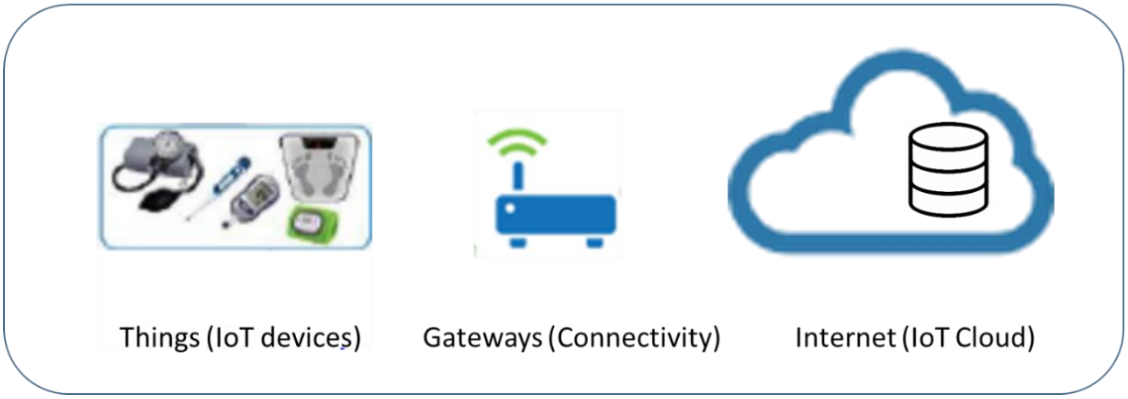


Fig. 1: IoT building blocks

There is a vast amount of research relating to health applications and we have chosen this area as our main topic. “Health is the fundamental capability humans require to perceive, feel, and act effectively, and as such, it represents a primary element in the development of the individual, but also of the environment humans belong to. That is why it is necessary to provide adequate ways and means to ensure the appropriate healthcare delivery based on parameter monitoring and direct providing of medical assistance.” [17]

The increase in popularity for wearable technologies has opened the door for an IoT solution to healthcare. One of the most prevalent healthcare problems today is the poor survival rate of out-of-hospital sudden cardiac arrests [18-21]. All existing systems for predicting cardiac arrest in the elderly mainly consider the various available heart rate parameters. It has been proven that heart rate derived features can give an early indicator of sudden cardiac arrest and providing an early warning has the potential to save many lives [22-24]. IoT wearable devices can provide this warning through the use of embedded sensors. By 2050, it is estimated that more than one in five people will be of age 65 or over [25]. Heart disease in the elderly is a very common occurrence. Approximately one-third to one-half of the elderly population experiences a heart attack or cardiac arrest on a yearly basis [26]. In an aging society, heart attacks have huge consequences since they tend to cause tremendous concerns as related to weakening in quality of life and an increase in the cost of healthcare.

Historically, seniors living all around the world have been known to be late adapters to the world of technology compared to their younger compatriots but their movement into digital life is continuing to expand. Today, 59% of seniors report that they go online, and 47% say they have a high-speed broadband connection at home. In addition, 77% of them have a phone and among that number, 18% are using smartphone devices [27]. With recent developments, smartphones have increased processing capabilities and are equipped with a number of built-in multimodal sensors, including accelerometers, gyroscopes, and GPS interfaces. As self-contained devices, smartphones present a common commodity and software environment for developing various cardiac arrest or heart attack detection systems. Smartphone-based heart attack detection systems can function almost everywhere since mobile phones are portable. Ideally, integrated sensors along with the ECG can automatically detect a risk of injury due to heart variability. Due to these and several other advancements in mobile technology, the elderly may increase their smartphone use based on such systems.

Existing cardiac arrest and heart attack detection systems can only detect a risk after it has already occurred, following which the system sends an alert to an emergency contact. The ideal way to reduce the number of risks is to alert the users about their abnormal heart conditions prior to potential heart attacks. This may cause false alarms based on how a user’s heart behaves during the day since fluctuations in our heart signals can be normal. These fluctuations can result from many things like sudden scares, emotional endeavors, or sudden movements like jumping, running, and exercising. However, having a few false alarms in this case would definitely be better than the alternative where heart attacks can only be detected but not predicted. Keeping in mind that the number of false alarms can be minimized the more the system is perfected. If abnormal ECG patterns can be accurately identified using automated processes, the elderly may be able to avoid injury from a potential Myocardial Infarction (MI). Therefore, our focus is on heart attack prediction rather than just detection. Though there has been much research on automatic heart attack detection, the area of heart attack prediction is still understudied. The need to identify all the possible patterns that can lead to a heart attack is very challenging.

1.3 Research Challenges

We aim to create a system that is unique and stands out when it comes to eHealth based IoT systems. In doing so we faced many research challenges that are listed below:

1) Developing a low energy communication channel between the IoT device and the smartphone application
2) Collecting, analyzing, and plotting data in real-time
3) Predicting heart abnormalities using a system of integrated sensors
4) Generating an alert message to a caregiver or emergency contact

1.4 Our Approach

Our goal is to create a system that solves those research challenges that have not quite been resolved using integrated IoT systems. Ehealth systems [28] are good examples of problems that can be handled using IoT devices. When successful, such systems will eliminate the need for patients to go to hospitals on every occasion. This becomes more important for older people who need special health care and monitoring throughout many hours of the day. Another concern is the number of deaths caused by strokes that were not timely noticed and then attended. Using ECG signals, along with other sensors, we can design a system that can **predict** heart attacks sooner. To accomplish this, we must study the human body’s action prior to the onset of a heart attack. Not only must we understand how the heart behaves, but also, we need to understand how other symptoms, such as body temperature, can indicate possible problems. By analyzing the data from the sensors, we can create an algorithm that can predict heart attacks.

IoT devices can be used to enable remote health monitoring and emergency notification systems. Health monitoring devices can range from blood pressure and heart rate monitors to advanced devices capable of monitoring and analyzing data. Using a number of sensors and a microcontroller, we can design a device capable of preventing health related disasters. In particular, we used several hardware components such as a pulse sensor, a temperature sensor, an Arduino microcontroller, and a Bluetooth chip (RN-42). This hardware kit (IoT device) basically senses and transmits the sensed data to a computing device for analysis.

Furthermore, we use the relationship between an individual's pulse rate and body temperature to design an algorithm that predicts heart attacks. This required a very good understanding of the human's body behavior in both normal and abnormal states. The normal states are simpler and can be analyzed easily. For example, we all know that a normal body temperature ranges from 36 to 37 (degrees Celsius), so intuitively, any temperature out of that range would be abnormal. Similarly, a person's heart rate in normal cases would range from 60 to 100 beats per minute. Any heart rate out of that scope is something that needs to be diagnosed properly and as soon as possible. Of course, humans have different heart rates in normal states which can vary for different age groups and different health conditions. Athletes tend to have lower heart rates around 50 to 60 beats per minute in resting state, while when exercising their heart rates can go up to 150 beats per minute or even more. People who do not normally exercise can have heart rates ranging from 70 to 90 beats per minute. So, it's very important to consider the different subjects that are under study.

Thus, we propose a smartphone-based heart attack prediction system that can alert users about their abnormal ECG patterns. Since abnormal ECG patterns can lead to a myocardial infarction, the system uses the identification of an abnormal heart-rate to alert the user regarding a potential heart attack. The proposed system is useful, not only among elderly, but also has a scope in identifying heart disease among children, adults, and stroke patients. Keeping track of a user's body temperature is very helpful and can provide additional indications to help predict heart attacks. In general, prior to heart attacks, people's body temperatures tend to go up. Even though there is a correlation between these entities, we do not assume that there is a direct relationship between them; however, this correlation gives a better chance of prediction

Chapter 2

Motivation

In this section, we discuss the impact of our system and the reasons that motivated us to pursue this area of research. We first present some information regarding health services and the reasons for which people need these services nowadays. Then, we describe some scenarios that drove us towards learning more about eHealth and IoT based systems.

2.1 Health Services

A key area of interest in today’s society is health services and improvements that can be made to them. Engineers are continually developing and evolving technologies that are critical to the enhancement of health services available to the public. One such area is remote sensing and monitoring, which allows doctors to get the status of their patients and inform them of critical conditions before they happen. This decreases the health-care cost as it frees up hospital equipment for the treatment of other patients who may need emergency services.

In the USA, electronic health monitoring has been given the go-ahead by the Federal Communications Commission (FCC). The FCC allows the use of allotted frequencies for sensors to control devices wirelessly in the monitoring of health at hospitals and homes. They allocated 40 MHz of spectrum bandwidth for Medical Body Area Network (MBAN) low-power, wide-area radio links at the 2360-2400 MHz band, primarily due to the saturated Wi-Fi spectrum [29].

The FCC forecasted the potential cost savings using MBAN as doctors can intervene before a patient’s condition seriously deteriorates, resulting in less time spent in the intensive care unit. The remote monitoring of patients can decrease infections that can be caused by hospital visits.

Healthcare applications can be used by elderly people to monitor their statuses and by their family members to identify their whereabouts. The application can be connected to a wearable device that is equipped with location-based sensors and attached to the elderly individual. This device can emit signals to alert family members about the condition of their elderly patient. Healthcare applications can also record important body conditions such as heart rates and body temperatures. The application can check and predict if the patient is going to have health problems. In the future, researcher are developing biodegradable materials for sensors that can be implanted on or in patients. The sensor chips can detect internal organ responses to new medication and guide the application of drugs to infected areas for better treatment. Also, new smart pills that are ingestible sensors, are swallowed to record various body measures.

2.2 Scenarios

In terms of motivation, there were many aspects that contributed to direct us to the area of eHealth and the use of IoT devices to enhance health care. One major trigger that made us realize this, is the very important fact that we are dealing with saving lives and trying to increase the average life span of humans. Many heartbreaks and sorrows happen every day when family members pass away due to heart failures. Although this is how life works, there is no reason why anyone should not have the opportunity to live a long healthy life surrounded by his loved ones because of insufficient monitoring of health. In this section, we stress the importance of a couple of scenarios that anyone would much rather avoid.

2.2.1 Scenario 1

Now that we provided a better idea of the aims of our system, we can talk a briefly about the scenarios that lead us to more depth in the eHealth area. One scenario that is quite common is that an person (young or old) starting to feel chest pains in a relatively isolated room. This may happen more often than not with the elderly who spend many hours of the day alone at their homes. Without anyone’s help, this person may have to go through cardiac arrest alone and might not be able to call for help one’s self and may be unable to move during the cardiac arrest event further inhibiting any ability to garner assistance or medical treatment. This scenario can be avoided if that person had a wearable device that predicted a problem and immediately alerted an emergency contact.

2.2.2 Scenario 2

Another scenario that occurs frequently is related to athletes or marathon runners. Marathon runners with family histories of heart diseases should not consider themselves immune to potential heart failure. A recent study [30] has stressed that physicians should not assume that physically fit marathon runners cannot have serious, life-threatening cardiac problems. A person preparing for a run or in a running state would be much safer if s/he had a wearable device attached to him/her, knowing that if any problem were to occur, the device would alert that individual or a caring person depending on how serious is the condition.

Chapter 3

Related Work

Given that the area under study is a very hot research area that is growing everyday as technology is improving, there has been a plethora of work done to which we can relate. Many individuals currently perform research in eHealth and many companies have taken advantage of this work by designing systems that connect patients with doctors around the world. We examine two different categories of related systems: comprehensive health care systems and connected eHealth mobile applications. Our work is more related to connected eHealth mobile applications since we are developing a mobile application that connects with an IoT device while huge corporations focus on comprehensive health care systems that allow patients to interact with one another and benefit from health articles and resources. However, we discuss both categories since what we are doing is essentially related to health systems.

3.1 Comprehensive Health Care Systems

There are many platforms that are shared by patients, doctors and research institutes to help all parties with their needs. One of the very well-known companies is called “PatientsLikeMe” with a main goal of listening to patients to identify their symptoms and treatments [31]. It focused on helping patients answer the question: “Given my status, what is the best outcome I can hope to achieve, and how do I get there?” They answered patient questions in several forms like having patients with similar conditions connect to each other and share their experiences.

Another related system is called “DailyStrength”. It i’s a social network centered on support groups, where users provide one another with emotional support by discussing their struggles and successes with each other. The site contains online communities that deal with different medical conditions or life challenges [32]. It is very similar to “PatientsLikeMe” in the sense that both of them are free platforms that involve patients and doctors interacting. Two major differences between them is that “DailyStrength” does not involve research institutes and does not have a mobile application. However, they are both global platforms that are very helpful for patients with health problems. “PatientsLikeMe” is more of a platform where patients share experiences of same symptoms while “DailyStrength” gives patients the option to talk to doctors. Another comprehensive health care system is called "Omnio” which is an all-in-one application for Medical Resources [33]. It provides, among its services, clinical resources, diagnostic resources, disease guides, drug information. Everyday Health [34] is a company which owns websites and produces content relating to health and wellness. It has higher ratings and publishes many health articles than can be very helpful for patients. In addition, it has a smart search that provides users with easy access their materials. A comparison of the comprehensive health care systems described above is shown in Table 1.

Table 1 Comparison of comprehensive health care systems

Attribute	**PatientsLikeMe**	**DailyStrength**	**Omnio**	**Everyday Health**
Mobile app	Yes	No	Yes	Yes
Diseases guide information	Yes	Yes	No	Yes
Drug guide information	Yes	Yes	Yes	Yes
Support Groups	Yes	Yes	No	No
Doctor-Doctor connection	No	No	No	No
Patient-Doctor connection	No	Yes	No	No
Patient-Patient connection	Yes	Yes	No	No
Free or Paid Subscription	Free	Free	Mix	Free

3.2 Connected eHealth Mobile Applications

Even though all the system mentioned above provide health services, they do not provide devices that can be used by patients to monitor their everyday activities and alert them when needed. On the other hand, there are many heart monitors that provide users with their ECG signals so they can keep track of their condition but none of which who alert the users upon emergencies. A Smart Elderly Home Monitoring System named SEHMS has been designed and developed on an Android™-based smartphone with an accelerometer; it could detect a fall of the user [35]. It provides a Graphical User Interface (GUI) to display health information gathered from the system.

The main advantage of SEHMS is that it provides remote monitoring for elderly and chronically ill patients.

Remote Mobile Health Monitoring (RMHM) is a system that provides monitoring of a user's health parameters such as his or her heart rate, which is measured by wearable sensors [36]. RMHM has a GUI that shows the wearer's health status. It allows doctors (and family members) to monitor the patient's condition to facilitate remote diagnosis.

Other researchers have developed relevant examples [37]. The Advanced Health and Disaster Aid Network (AID-N) system is an electronic method to handle emergency responses to increase quality and quantity of patient care. The Personal Health Monitor (PHM) system is designed for patients who have a suspected cardiovascular disease and need to be monitored around the clock. The system uses sensors for each patient to provide personalized monitoring and treatment. A comparison of the related systems is shown in Table 2.

Table 2 Comparison of connected eHealth mobile applications

Attribute	**SEHMS**	**RMHM**	**PHM**	**DTSDSA**
Uses Temperature Sensor	N	Y	N	**Y**
Uses Pulse Sensor	N	Y	N	**Y**
Shows Heart ECG	N	N	Y	**Y**
GUI	Y	Y	Y	**Y**
Real-time Plotting of Data	N	N	N	**Y**
Low-energy system	N	N	N	**Y**
Mobile System	Y	Y	Y	**Y**
Emergency Detection	Y	Y	N	**Y**

Many IoT systems rely on using wearable devices. Qardiocore is a well-known and high quality heart monitor that tracks a user's complete heart health and displays it on smartphones [38]. The device yields very accurate results and is one of the best products on the market in terms of showing real time graphs of ECG. However, it only detects cardiac conditions and allows users share data with their doctors. It does not give the patient/user the option of alerting the medical person in real time when their heart is at a serious condition and it certainly does not predict heart attacks. In fact, neither of the systems described above have the option of heart attack predictions.

Other related systems have been implemented and can successfully detect heart attacks. The idea of predicting heart attacks remains a challenge and that is the focus of our research. Every research group specifies its own approach on how they plan to achieve its objective. We decided to use a combination of body temperature and heart data to predict heart abnormalities. Other systems have different approaches with different hardware implementations. The attributes in Table 2 are all binary which means that the systems either have the attribute or not. While some systems have similar features to our system, none of them were concerned about energy consumption as much as we made it a priority. Our system uses a low energy Bluetooth module (BLE) which gives our IoT device a very long lifetime. The details of the device's performance are discussed in Chapter 4.

Recently published papers present a comparison between different data mining techniques for heart attack prediction [23,24]. Those papers present just prediction algorithms rather than a complete system with a data collection device and a computing platform. As shown in those papers, the best techniques that are most commonly used for predicting heart problems are: Decision Tree, Naïve Bayes, Neural Network, and K-mean. Our research not only includes a complete system with an IoT device and a computing platform, but also uses one of those data mining techniques (Decision Tree) to predict heart problems. This makes our system unique in the sense that we created a low energy IoT device and we applied a data mining technique in our prediction algorithm. Upon testing our prediction algorithm, we obtained results that were almost perfect for all our healthy and unhealthy test subjects. The algorithm's results are discussed in more detail in the Chapter 7.

Chapter 4

System Architecture

To integrate the sensors, we used the output of the embedded sensors to perform an extensive set of experiments for evaluating and discriminating between normal and abnormal heart rate patterns. Subjects wear the embedded sensors, and carry their smartphone in their pocket or hold it in their hands. The embedded ECG and temperature sensors constantly collect the heart parameters while the subject is living a normal life. After receiving the data through a LE Bluetooth communication channel, the smartphone will process the data to classify whether the user's condition is normal or abnormal. A quantitative heart-rate analysis is performed in the Android platform which gives the user the option of viewing his/her real-time plots of the ECG signal and body temperature.

To determine abnormal heart patterns, we first establish a criterion for normal heart-rate. Quantitative analysis of heart rate stability and pulse symmetry will yield a series of parameters, like heart rate, RR intervals[1], and ST segments[2]. We then design an early warning system to monitor those parameters for signs of cardiac arrest during any activity. Although the system continuously monitors ECG patterns, the planned design only triggers a warning if the ECG patterns and body temperature of the user reaches a certain point, wherein the user might face a potential heart attack. At that moment, the system transmits a warning to the subject in the form of a message or a vibration alert.

Fig. 2 shows the flow of the system's architecture. We discuss all the components in detail in the next section but, basically, the IoT device constantly collects data from the user and sends

[1] RR Interval: is the duration between two consecutive R peaks in an ECG signal.

[2] ST Segment: is the flat section of the ECG signal between the end of the S wave and the beginning of the T wave. It represents the interval between ventricular depolarization and repolarization.

it via Bluetooth to the application as shown in the Fig. 2. All the processing and data analysis take place in the application where the user has the option to view his/her real-time plots. These plots provide the user a basic idea of his/her body’s status. The user does not have maintain a record of his/her data to ensure that s/he is in a healthy or unhealthy state since the application’s job is to alert the user upon an emergency. Finally, when the algorithm senses an abnormality it immediately alerts the user.

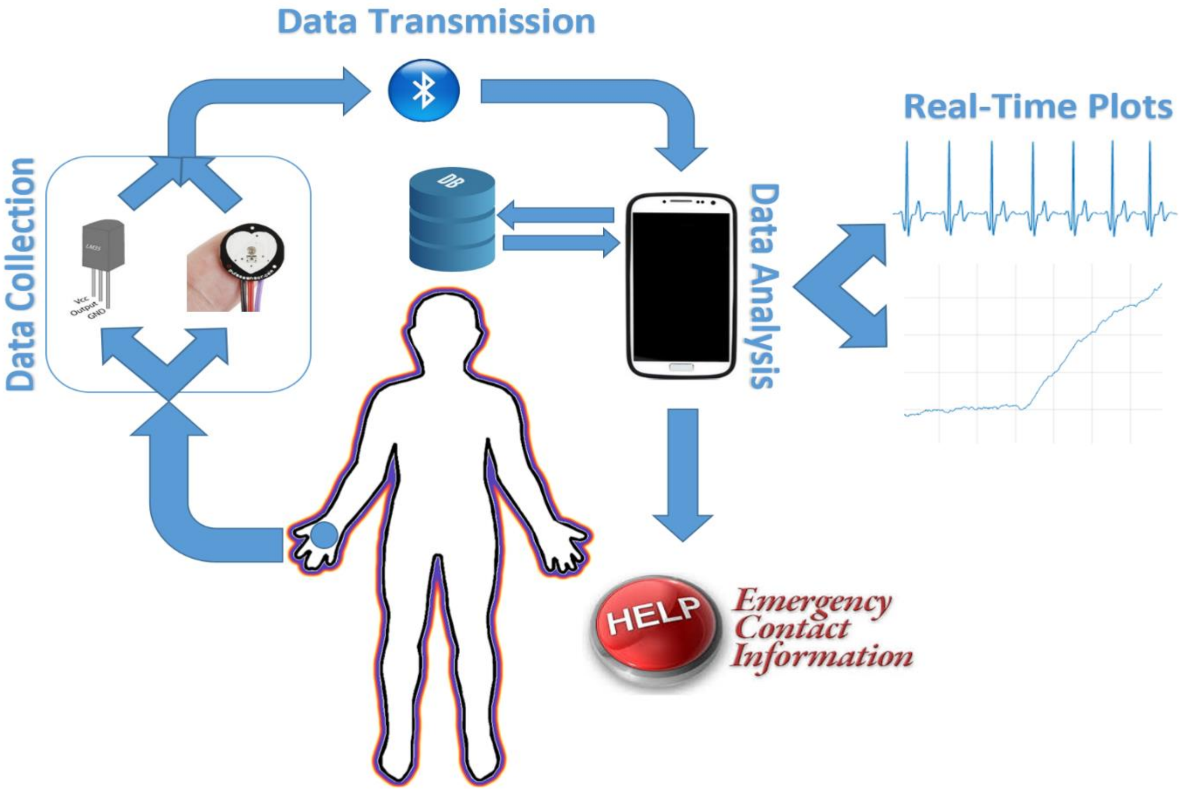
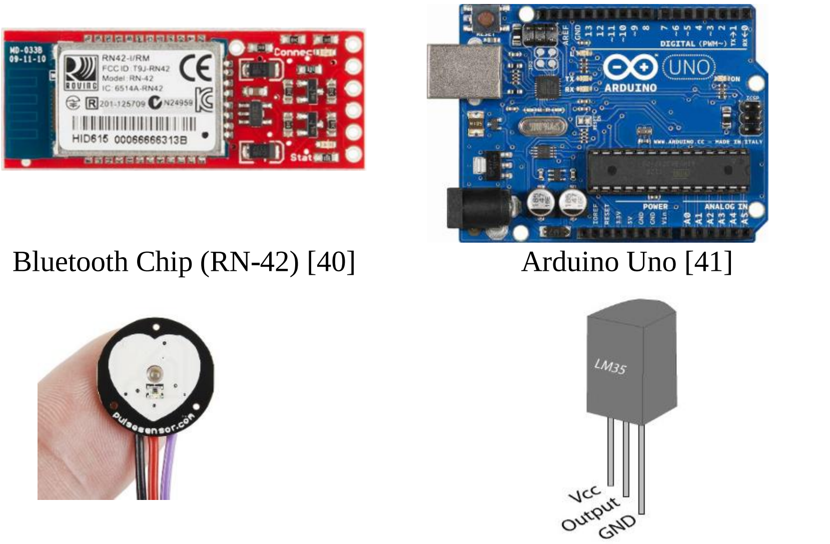


Fig. 2: System architecture

4.1 Hardware

The developed IoT device consists of four main components which are listed in Fig. 3 below. A LE Bluetooth chip, an Arduino Uno™, a pulse sensor, and a temperature sensor were all used to build the IoT device. The other components are the power supply unit along with a smartphone that has the application downloaded on it.

The Arduino simply serves as an Analog to Digital Converter (ADC) [39]. An Arduino is an open-source physical computing platform based on a simple I/O board and a developmental environment that implements the processing/wiring language. The Arduino is programmed to read analog signals from the pulse and temperature sensors and create a data packet to convert the signals into digital form. Subsequently, it sends those packets to the phone as a response to the data sending request. It also manages the Bluetooth communication by coordinating with the RN42 Bluetooth chip. The Bluetooth chip basically equips the Arduino with the ability to connect to the smartphone application.



Bluetooth Chip (RN-42) [40]

Arduino Uno [41]

Pulse Sensor [42]

Temperature Sensor (LM-35) [43]

Fig. 3: Hardware components

The data read from the sensors is always an analog value between 0 and 5 volts since that is the operating voltage of this microcontroller. The Arduino then maps those voltage values to digital values ranging from 0 to 1023. To convert those values into numbers that make sense we use the equations that correspond to each sensor. Since the y axis for ECG signals is also a voltage, all we had do is scale the digital values to back voltage. The following lines of Arduino code in Eq. (4.1- 1) and Eq. (4.1- 2) show how we achieve that.

int sensorValue = analogRead(A0); (4.1- 1)

float voltage= sensorValue * (5.0 / 1023.0); (4.1- 2)

Basically, we read the sensor value from the Arduino through analog pin 0 and then multiply it by 5 and divide it by 1023 to get the correct voltage value. This only applies to the pulse sensor since the expected output from the temperature sensor is in degrees Celsius. The following lines of Arduino code in Eq. (4.1- 3) and Eq. (4.1- 4) show how to get Celsius values from the temperature sensor.

int sensorValue = analogRead(A5); (4.1- 3)

float temperature = (5.0 * sensorValue * 100.0) / 1024; (4.1- 4)

Similarly, we read the sensor value through analog pin 5 and then use the equation shown above to get the temperature in Celsius. The manufacturers of the sensor (LM-35) and Arduino specified this equation and all we need is to get the correct temperature value.

Reading simultaneous readings from multiple analog pins in an Arduino results in inaccurate values. To avoid that problem, we not only need a delay between each reading but also need to read from the same analog pin twice. So basically, we read the temperature data from the sensor twice and send the second reading, then do the same for the pulse sensor. Of course, we need to send different symbols before the sensor readings to be able to parse the data at the receiving end (android application). Before sending a temperature reading we send a '/' and before sending a pulse reading we send a '- ', which makes data parsing simple. Finally, the battery power supply gives the IoT device the mobility. This concludes the description of the functionalities of all the hardware components.

4.1.1 Hardware Modifications

After proposing our research, we worked on modifying the hardware to develop a better IoT device that can later on be used as a user friendly wearable device. In this section, we will discuss the new hardware components used, the design of the wearable device, and the performance of the device (power consumption /current draw).

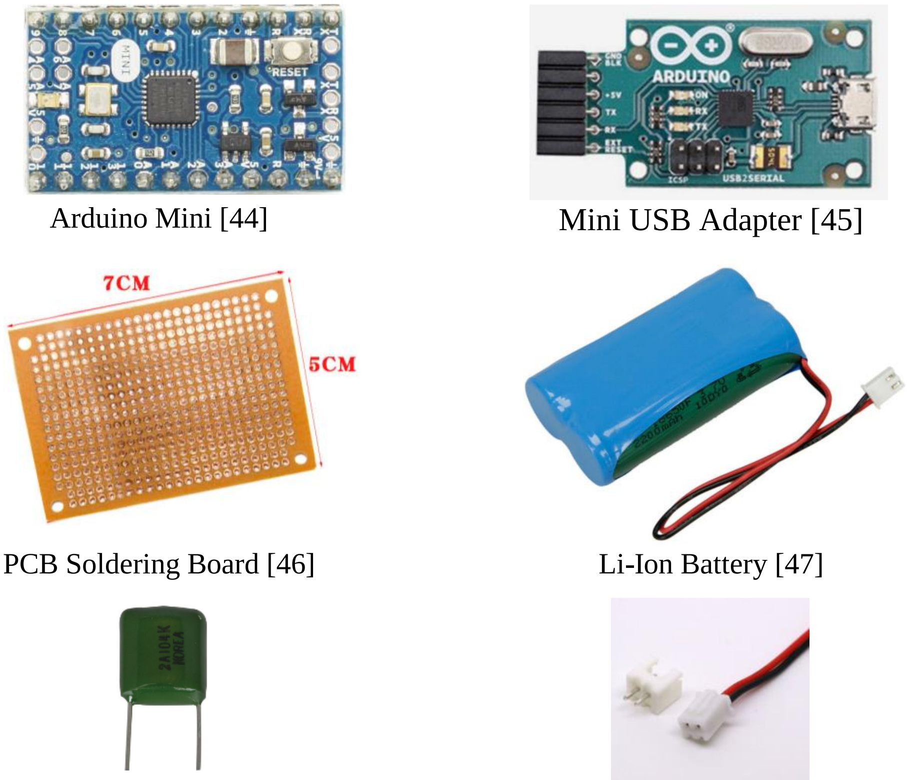
4.1.1.1 New Hardware Components

Rather than using the Arduino Uno, we decided to use the **Arduino Mini** instead for many reasons that are explained below. Table 3 shows a comparison between both computing platforms.

Table 3 Comparison between Arduino Uno and Arduino Mini

	Length (cm)	Width (cm)	Area (cm^2)	Clock (MHz)	Input Voltage (Volts)	Operating Voltage (Volts)	Microcontroller
Arduino Uno	6.86	5.34	36.63	16	7-12	5	ATmega328P
Arduino Mini	3	1.8	5.4	16	7-9	5	ATmega328P

They both have the same microcontroller, clock speed, operating voltage, and range of input voltage. The main reason why we used the Arduino Mini instead of the Arduino Uno was the size difference. The Arduino Uno has an area of 36.63 cm^2 which is almost 7 times larger than the Arduino Mini. When developing a user friendly wearable device, it is crucial to have smaller components to be able to design a compact device. One challenge that we faced was that the Arduino Mini does not have a serial interface that allowed us to upload code. Therefore, we had to buy a **Mini USB Adapter** that gave us that capability. To be able to upload code to the device, we also needed an **0.1 µF (micro-farad) capacitor** connected in series between the reset pin of the Arduino Mini and the reset pin of the Mini USB Adapter. We used a **PCB soldering board** to solder all the hardware components together. The board, which has dimensions of 5 cm x 7 cm (almost the same size of the Arduino Uno), has all the hardware components soldered to it. To power the device, we used a **7.4 Volt Lithium Ion battery** with a current supply of 2200 mAH (milli-amperes per hour). This battery has an outlet plug that gives it the ability to recharge. So, we also bought a **Pin Battery Connector Plug** to insert the battery in. This allows us to solder the pin plug to the board without soldering the battery itself, allowing the user to remove the battery when it needs to be recharged. All the components that we added (shown in **Bold** in this section) are shown in Fig. 4 below.



Arduino Mini [44]

Mini USB Adapter [45]

PCB Soldering Board [46]

Li-Ion Battery [47]

Capacitor [48]

Pin Battery Connector Plug [49]

Fig. 4: New hardware components

4.1.1.2 Design of the Wearable Device

After soldering all the hardware components on the PCB board, we had to design a method that allows the user to wear the device. We decided to make the device wearable on the wrist of the user using Velcro strips. The device is designed such that the Mini USB Adapter can be connected only when we need to modify code on the Arduino. The final design of the device is shown in Figs. 5 and 6, where Fig. 5 shows the device with the Mini USB Adapter attached and Fig. 6 shows the device without the Mini USB Adapter.

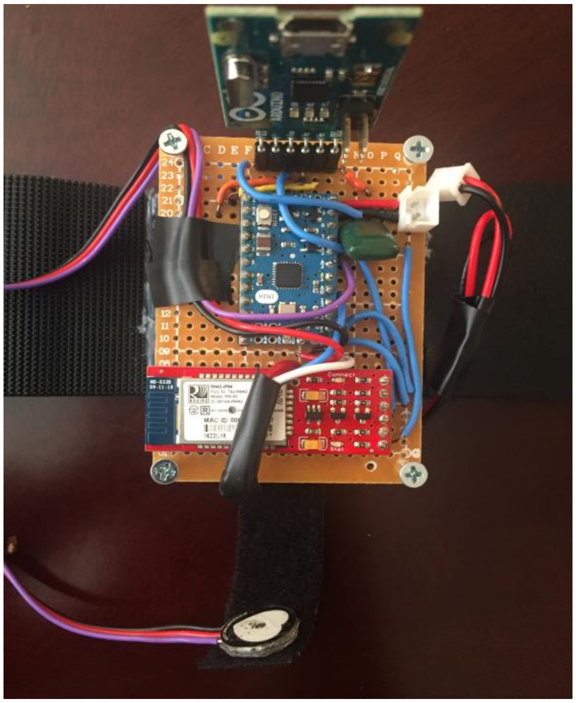


Fig. 5: IoT device with Mini USB Adapter

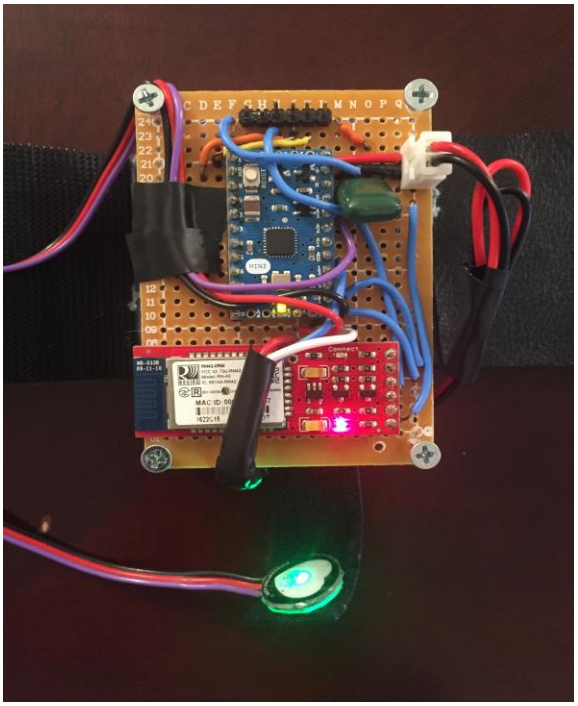


Fig. 6: IoT device without Mini USB Adapter

Figure 6 shows the device when the battery is active; hence, the LEDs of the Arduino Mini, Bluetooth, and pulse sensor are all on. The wires connected to the battery can be easily plugged in and out of the IoT device to allow the user to power the device on and off. The battery is placed between two PCB soldering boards that are supported on top of each other using the four screws at each corner. The temperature sensor’s connection mounts over the Bluetooth chip and under the lower PCB board, where it will be in contact with the user’s skin when the device is worn. The pulse sensor extends to the palm where it should be wrapped around the user’s index finger. Finally, the Velcro is glued to the bottom of the lower PCB board and covered in black leather to give the device a better appearance. Figure 7 shows how the device should be worn. Note that this device is only a prototype that was built to accomplish our goal. However, if this device was to be introduced to the market as a feasible selling product, it would have to be miniaturized into one chip that makes it much more usable and friendly.

4.1.1.3 IoT Device Performance

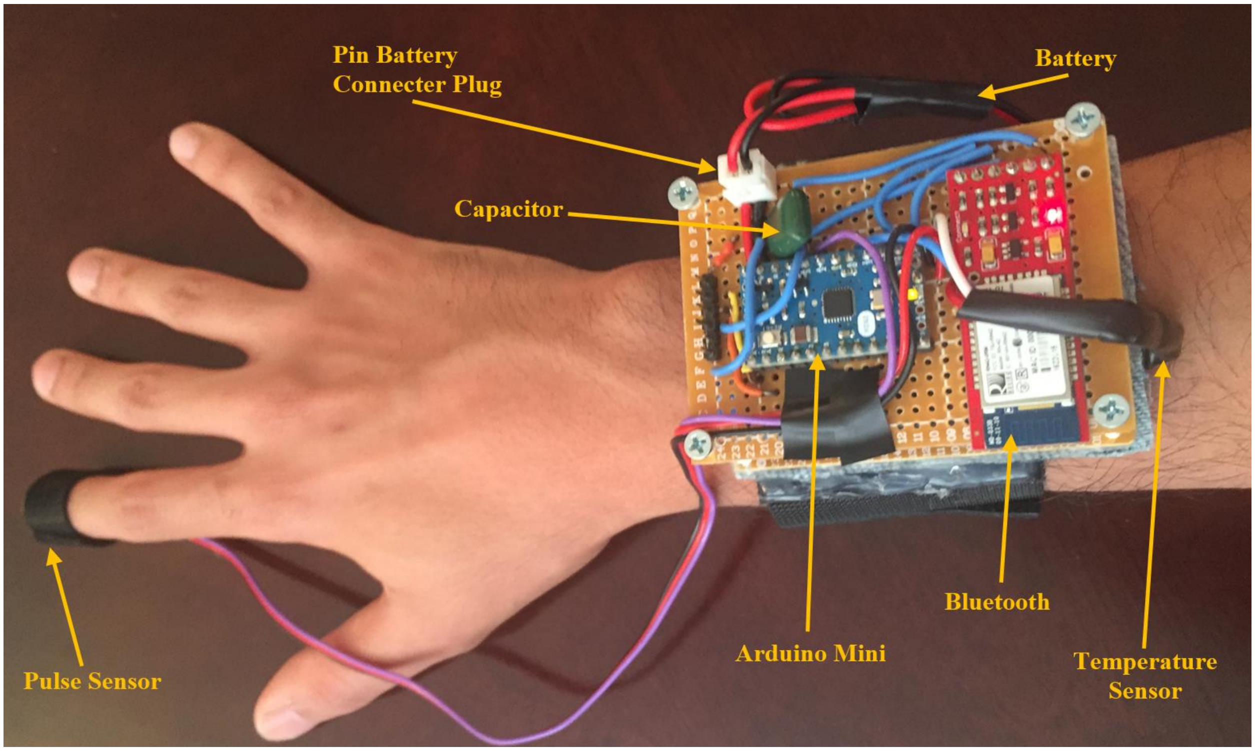


Fig. 7: Wearable IoT device

The next part we elaborate on the power consumption of the IoT device in different modes. When the IoT device is powered, the Bluetooth enters the idle mode where it blinks on and off waiting for a connection request. When the Android device connects to the IoT device through the application, the Bluetooth's LED stops blinking and is set to green indicating a succesful connection.

The performance of the device can be determined by measuring the current consumption which tells us how long the device can be powered. For LE IoT devices, the main objective is to have a device powered and working with minimal voltage consumption and current draw. The voltage supplied from the battery is constant since the Arduino Mini takes the voltage it needs and supplies the devices connected to it. The typical way to determine the performance of the device is by checking the amount of current that is drawn from the battery in the different modes. The two modes in which we need to test the device are: the idle mode and the connected/transmitting mode. The measuring unit of the battery is in millamp hour (mAH) which is an energy measure.

A battery with 2200 mAH will work for an hour if the current drawn from it is always 2200 mA. Similarily, if the current draw is 1100 mA, the battery would last two hours. Therefore, to measure how long the device can be powered in the on state without the battery draining, we need to calculate the average current draw of the IoT device. Table 4 shows the current draws, the device's lifetime, and the power consumtion during the two modes for the IoT device.

Table 4 Performance metrics

	Current Draw (mA)	Lifetime (Hours)	Power Consumption (mW)
Idle mode	26	84	192.2
Connected/Transmitting	60	36	444

The device shows that it has a very long lifetime in both modes which makes it very useful for users. When the battery is too low on power to operate the device, it can be recharged by simply plugging the battery's wires to a charger. This validates that our device is of the LE type.

4.2 Software

To receive and analyze data from the IoT device, we use a heart rate and body temperature collector interface in the smartphone. As described in the hardware section, we developed a Bluetooth communication channel that is capable of transmitting data from the pulse and temperature sensors to the smartphone. On receiving data from the sensors, the system processes the data to identify any abnormality.

To have a fully functional system, we had to start with a wireless communication channel to transmit data from the IoT device to the application. To achieve that we opened a socket from the Android application that connected to the transmitting signals of the Bluetooth module. To communicate with the Arduino, we created a software serial object and specified the transmitting and receiving pins. When the Bluetooth is supplied with power, it immediately enters the pairing mode, where it waits for any device to connect to it. At this point, the, mobile phone's Bluetooth adapter is opened through the application and it starts searching for devices around the phone. When the Bluetooth chip's name appears on the screen, simply clicking on it opens a connection between the two devices. When a successful connection can be confirmed from both sides, the application will produce a message on the screen informing the user that the connection was successful, and the Bluetooth chip's LED will turn from red (pairing mode) to green (connected mode).

After successfully connecting to the IoT device, the application will automatically start receiving the sensors' data. At this point, the application parses the temperature and pulse data into separate arrays that are then sent to different pages where they are plotted in real time. The user has the option of either viewing the separate plots for each sensor data or viewing a page that has both plots in real time. While data is being plotted, the algorithm is constantly examining the ECG data waiting for any abnormality.

The next part shows the pages of the android application starting with the login and sign up pages. When starting the application, the user will have the option of either signing up or logging in depending on whether the user has an account or not. If the user has an account s/he can simply enter the username and password to login. If not, clicking on the sign-up button will take the user to another page where s/he will be asked to enter some information to create an account. Figs. 8 and 9 show the login page and sign up pages, respectively.

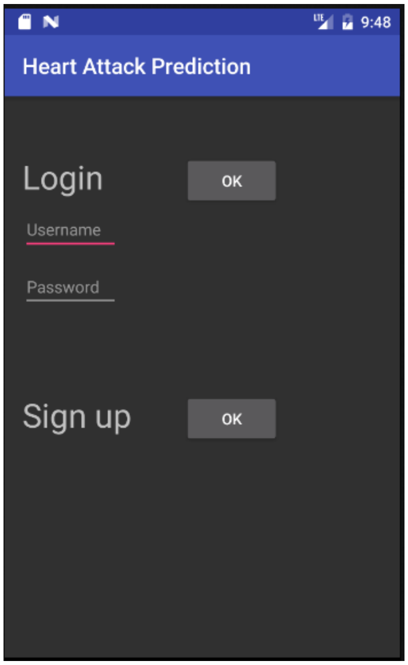


Fig. 8: Login page

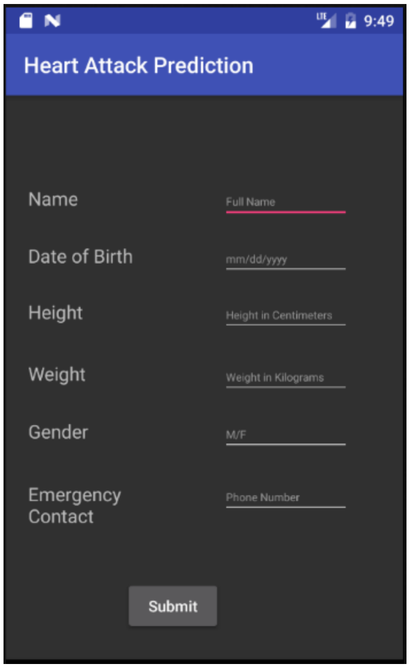


Fig. 9: Sign up page

The user will then be directed to the home page (shown in Fig. 10) of the application where s/he will have different options. The user will need to connect to the IoT device before s/he can start viewing his/her data. This can be done by pressing the connect button which will take the user to another page where s/he can find the device.

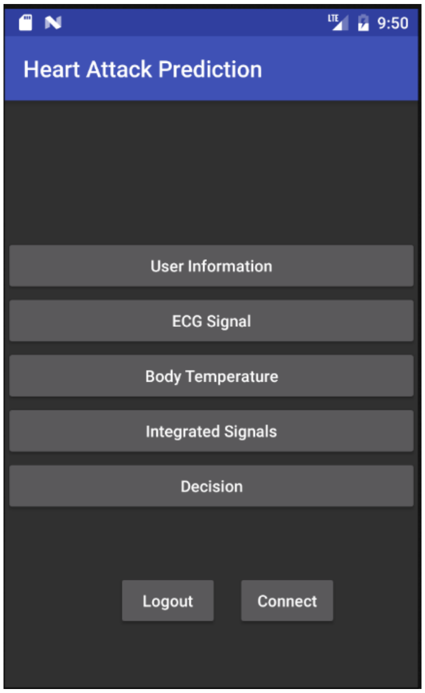


Fig. 10: Home page

The pages shown in Fig. 11 are screenshots from the ‘Connect’ page. The one on the right is the first thing the user sees when s/he enters the page. First the user needs to turn on the Bluetooth of the Android device. By pressing the “TURN ON” button, the Android device will respond to the application’s request, asking the user if the application can open the Bluetooth and by hitting yes, the Bluetooth turns on. The figure on the left shows what happens when the user hits “LIST DEVICES”. Since we are working on a prototype, we named the Bluetooth device “YOSUF” and only allow the user to connect to this Bluetooth device for simplicity. Simply pressing on “YOSUF” will connect to the Bluetooth chip which opens the communication channel between the user and the IoT device. The user can then go to the home page where s/he will have several options between viewing his/her real-time plots of the sensed data or going to the decision page. The decision page will basically have information that describes the user’s current health status.

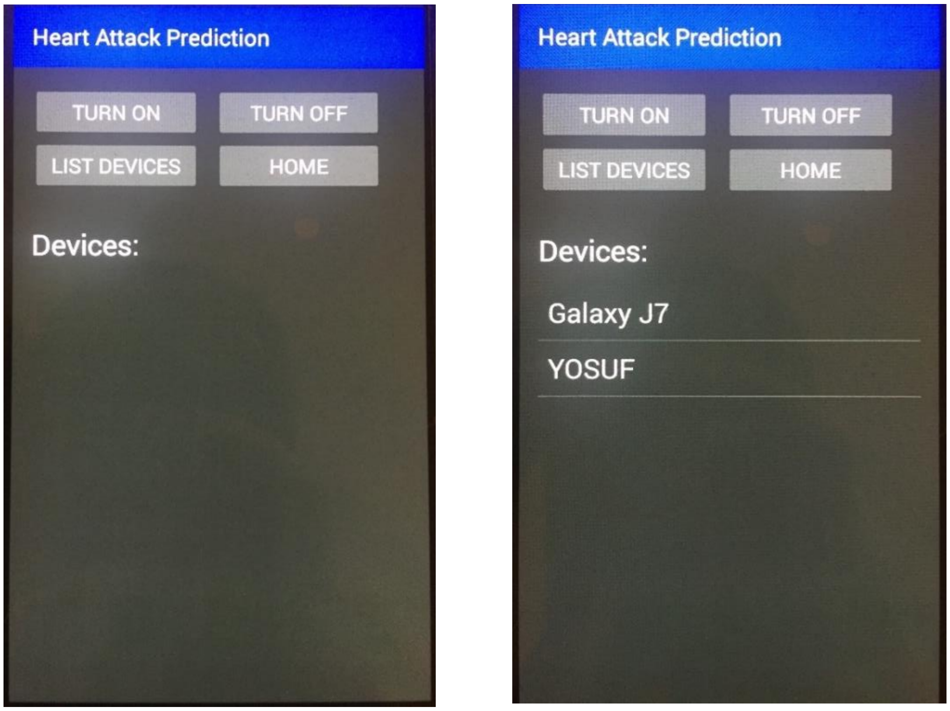


Fig. 11: Connect page

Screenshots in Fig. 12 show the real-time plots for each sensor individually and for both sensors on one page. The time axis shows that the graph retrieves the current time from the Android device and displays it in real-time as the axis moves with incoming data points.

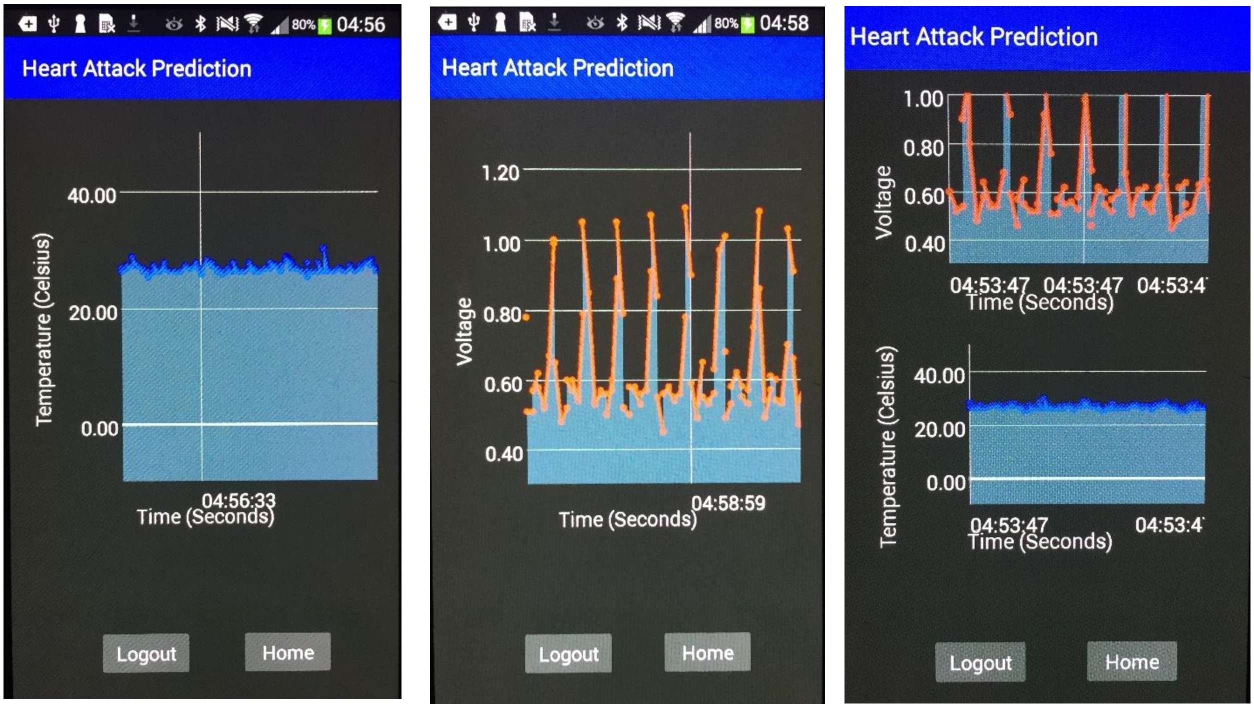


Fig. 12: Real-time plots of sensors' data

We conclude our software discussion by combining all the pages in the application in one diagram to give an overview of how the user can access each page. Figure 13 shows the GUI pages.

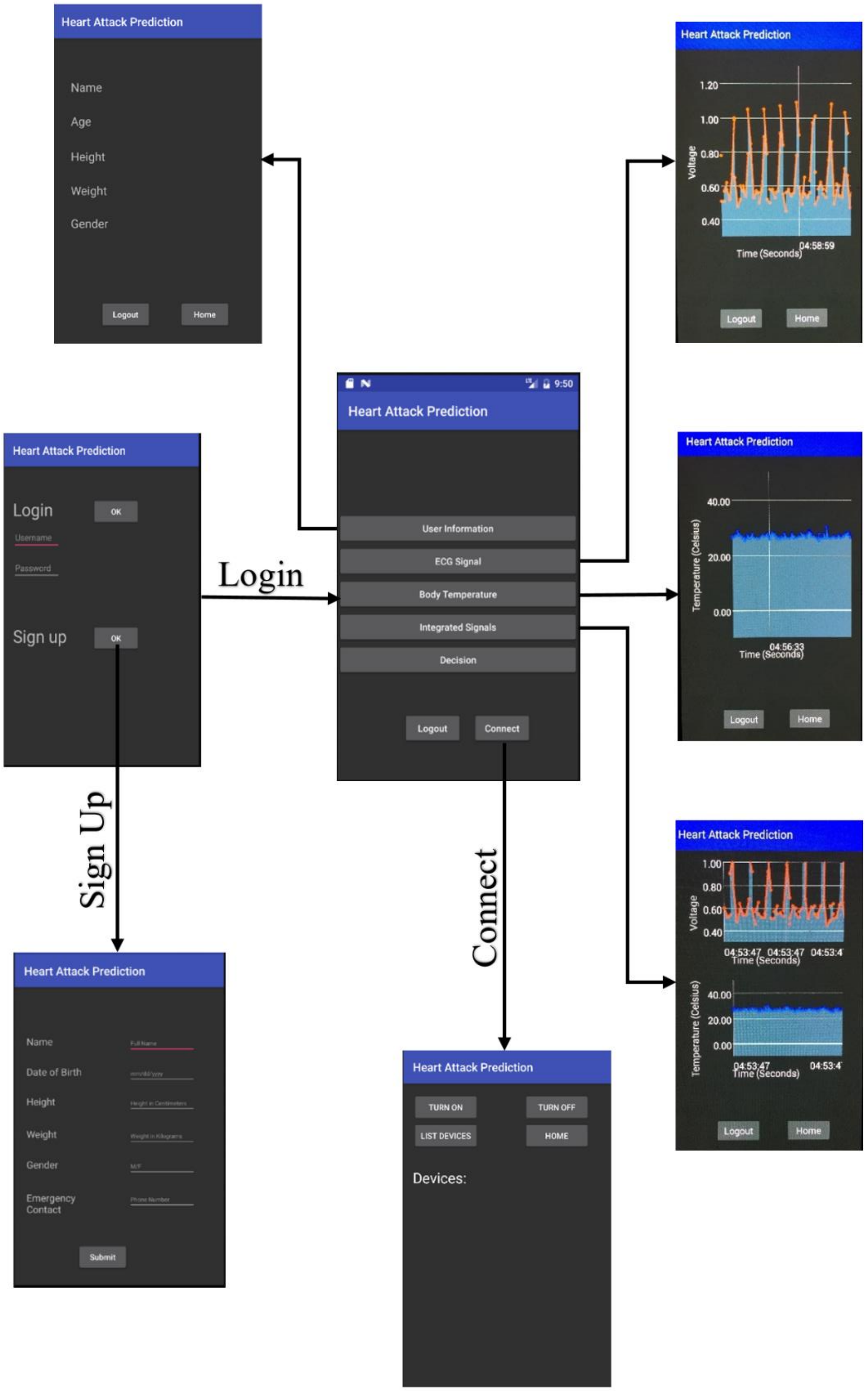


Fig. 13: Graphical user interface of the application

4.3 Summary of System Components

In this section, we summarize all the components that were used to build the system in Table 5.

Table 5 System Components

Category	**Item**
Hardware	Arduino Mini
	Bluetooth RN-42
	Pulse Sensor
	Temperature Sensor (LM 35)
	Li-Ion Battery + Pin Battery Connecter Plug
	PCB board + Capacitor + Wires + Male Headers
Software	Android Smartphone Application
	Arduino Mini software
	MATLAB prediction algorithm
Supplemental Materials	Leather + Jeans + Screws + Velcro

This information can be helpful to people who plan on creating such a system. The hardware components listed in the hardware category and the supplemental materials are sufficient to help developers create our IoT device. Of course, the IoT device can always be improved by using more advanced components. For example, one can design a smaller device by using a Nano Arduino instead of the Mini Arduino. There are also many other Bluetooth modules, temperature sensors, pulse sensors, and batteries out that that can be used instead of the ones we chose. However, we believe that the hardware components that we chose were enough to give us a LE IoT device that provides the performance metrics we set at the start of the research. The supplemental materials can also be upgraded to create a more user friendly wearable device. For the software, we have the Android smartphone application that collects the data and plots it in real time, the Arduino software that is responsible for sending the data via Bluetooth, and the prediction algorithm written in MATLAB™.

Chapter 5

Data Collection

Gathering data is the most important task of the IoT device. This process can be divided into two main parts, reading the data from the sensors and sending it to the application. For the first part, one sensor gets the heart's pulse rate and the other one gets the body temperature. As described in the previous chapter, once the required data arrives at the computing device, it is parsed and plotted on the device's screen. Each plotting page contains its corresponding data. The plot gives the user the indication of the status of the patient under investigation.

5.1 Data Collection Interface

One of the most important things in data collection is the sampling frequency or the rate at which we are collecting the data. For our system, we send the data from the two sensors simultaneously, so intuitively, the sampling rate for our system would be less than the sampling rate of a system that reads data from just one sensor. Given that the body temperature does not undergo as many changes as the ECG signal, we increased the ECG's sampling rate at the cost of decreasing the temperature's sampling rate. We fixed the sampling rates for the temperature sensor and the ECG signal at 5 Hz and 160 Hz, respectively. Figure 14 shows the block diagram that describes the data collection interface from both ends, the sensors' side and application's side. The two sensors are connected to different analog pins on the Arduino. The Bluetooth chip is alsoconnected to the Arduino which enables the IoT device to transmit the sensed data to the Android application. This gives the user a data collection interface that is simple to use on both ends.

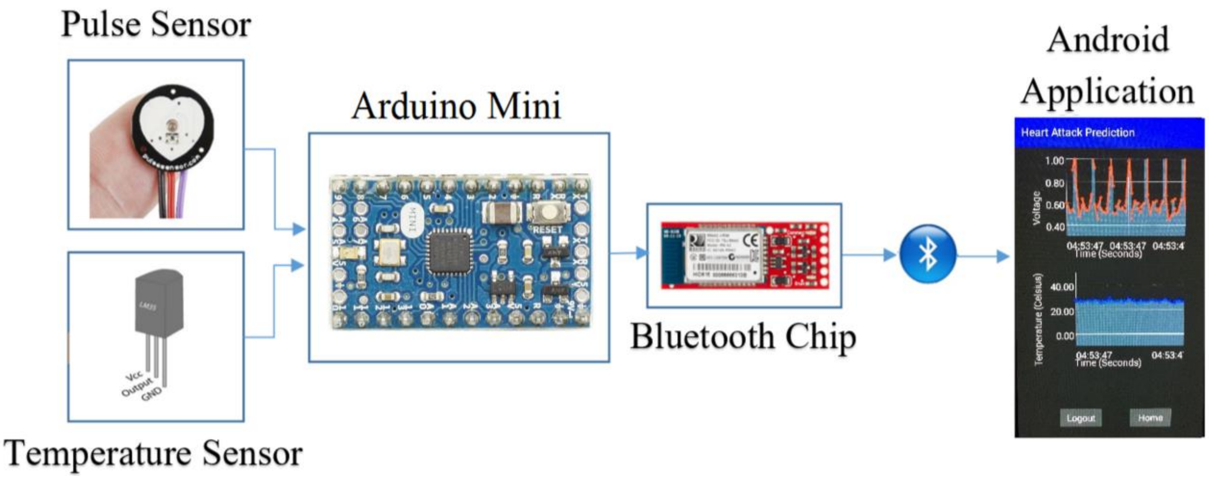


Fig. 14: Block diagram of data collection interface

The user first wears the device as described in the hardware section and then uses the application to connect to the Bluetooth interface as described in the software section. From this point the user only needs to interface with the application where s/he can navigate through the different options.

5.2 Subject Data Collection

This system is fully equipped to collect data from people and store it in the phone's database but, more importantly, it can plot and process the data in a fast manner. To be able to write our algorithm, we had to collect data from many subjects while doing different activities to see how the heart behaves under different circumstances. The three scenarios that we consider for each subject are: sitting, walking, and climbing (upstairs). We believe that those different scenarios can help us understand how everyone's heart behaves during different activities. Table 6 shows the different subject groups from which we collected data. We also used data from other resources to have a better scope of the different heart behaviors for different age groups with different weights and heights and different heart conditions.

Table 6 Test subject groups

	Weight Range (lbs)	**Height Range (cm)**	**Age Range**	**Testing Scenarios**
7 Males	128 - 220	165 - 190	23 - 26	➢ Sitting ➢ Walking ➢ Climbing Upstairs
3 Females	125 - 180	160 - 184	20 - 34	

5.3 Subject Sample Data

The data collected show that the system has a data collection system that is capable of gathering data under any circumstances, such as in the three scenarios described in Section 5.2. In this section, we show the sample ECG data for each of our four test subjects. The temperature sensor sample data are just plots to demonstrate the accuracy of the sensor and to show that it works.

5.3.1 Temperature Data

In this subsection, we present the detailed data for our temperature sensing process. Temperature does not need much analysis except for converting the data points to the time domain and smoothing the signal for better visual representation. Figure 15 shows a typical result; noting the "noisiness" of the signal indicates a need for smoothing.

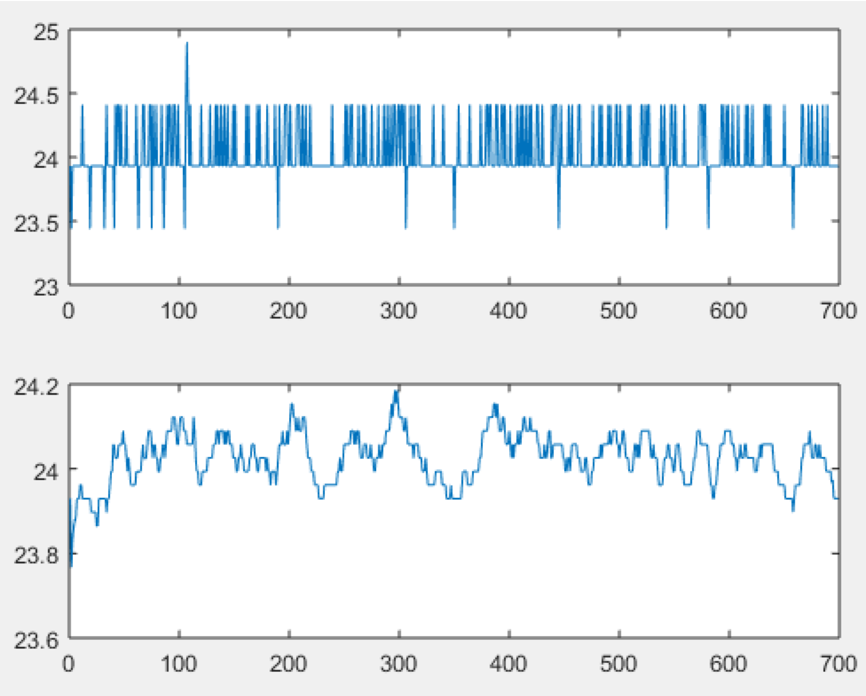


Fig. 15: Sample temperature data

Of course, the smoothing operation will not affect the decision; its only purpose is to show more consistent flow of data. The *y*-axis represents the temperature in Celsius and the *x*-axis shows the number of data points. To convert the data points to time in seconds, we need to use the sampling frequency which for this case was 100 Hz. The sampling rate that was used here was just to demonstrate the plot in an easier way since 700 hundred data points can be easily mapped to 7 seconds using 100 Hz. However, the sampling rates used for our system are still 5 Hz for the temperature data and 160 Hz for the ECG data. Figure 16 shows a set of data when converted from data points to time in seconds.

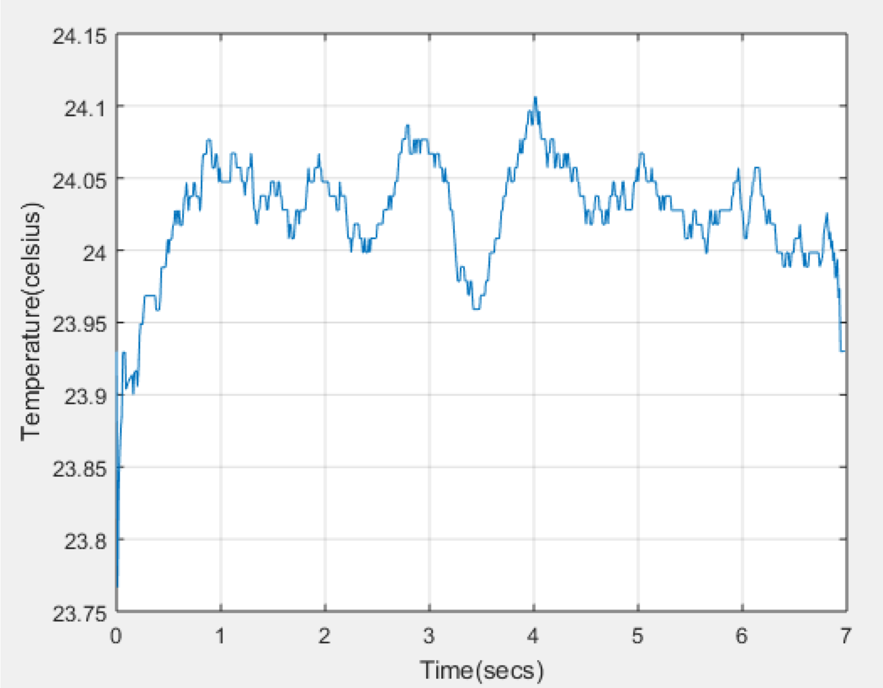


Fig. 16: Temperature data in time domain

The temperature sensor used in our work has an accuracy of +/- 0.5, which allows it to capture changes in temperature very quickly as shown in the 7 second plots in Fig. 17. The one on the left shows the temperature decreasing while the one on the right shows the temperature increasing.

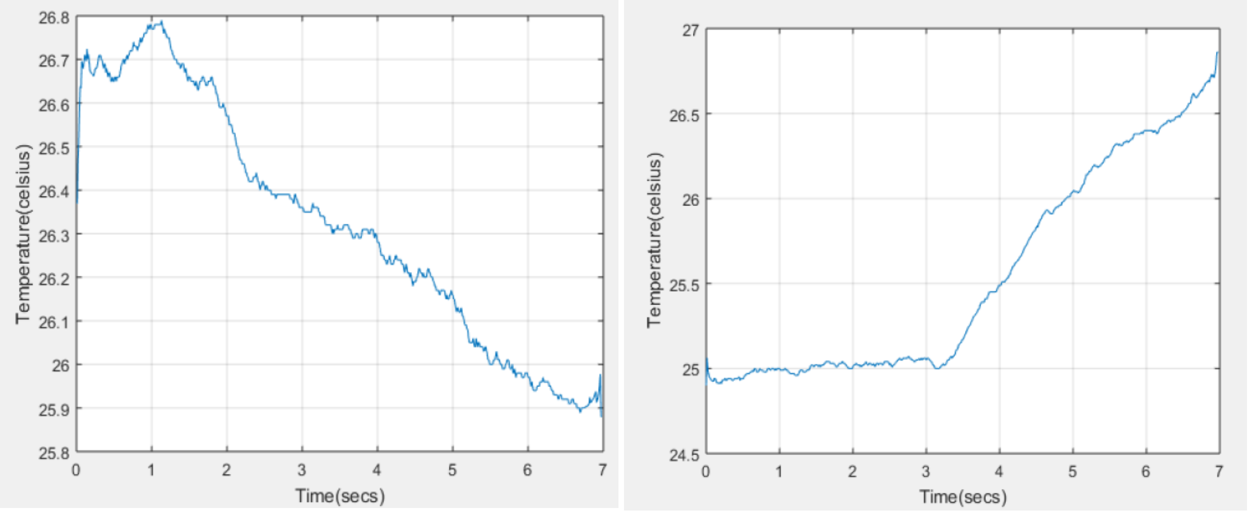


Fig. 17: Temperature accuracy

5.3.2 ECG Data

ECG data was collected from ten subjects and analyzed on MATLAB. In this section, we show the data of four test subjects in the three scenarios, two males and two females. We were able to collect data for the walking scenario using treadmills and for the climbing upstairs scenario using stair steppers at the rec center. For each scenario, we show the ECG signal and its corresponding heart rate. The heart rate was ultimately calculated using the Fourier transform method to make sure it is accurate [50]. Table 7 shows the information of the four test subjects.

Table 7 Test subject information

Subject	Weight (lbs)	Height (cm)	Age	Scenario	Heart Rate
Subject 1 (Female)	125	173	20	➢ Sitting ➢ Walking ➢ Climbing Upstairs	➢ 107 ➢ N/A ➢ N/A
Subject 2 (Male)	141	177	24	➢ Sitting ➢ Walking ➢ Climbing Upstairs	➢ 72 ➢ 98 ➢ 108
Subject 3 (Male)	163	180	23	➢ Sitting ➢ Walking ➢ Climbing Upstairs	➢ 72 ➢ 100 ➢ 134
Subject 4 (Female)	128	184	23	➢ Sitting ➢ Walking ➢ Climbing Upstairs	➢ 79 ➢ 89 ➢ 105

The data collected for Subject 1 is considered useless data for two of the three collected scenarios. The data collected while sitting had no problems. Problems occurred when the data was collected while the subject was walking and climbing upstairs. This is a result of the sensor moving while the subject was performing the different activities. Data had to be collected multiple times carefully before we can consider analyzing it. However, we decided to present the noisy data obtained for Subject 1 to show the major distinction between noisy and proper ECG data. Therefore, the heart rates for Subject 1 for the last two scenarios are displayed as N/A. Figs. 18 - 21 show the data collected for Subjects 1, 2, 3, and 4, respectively.

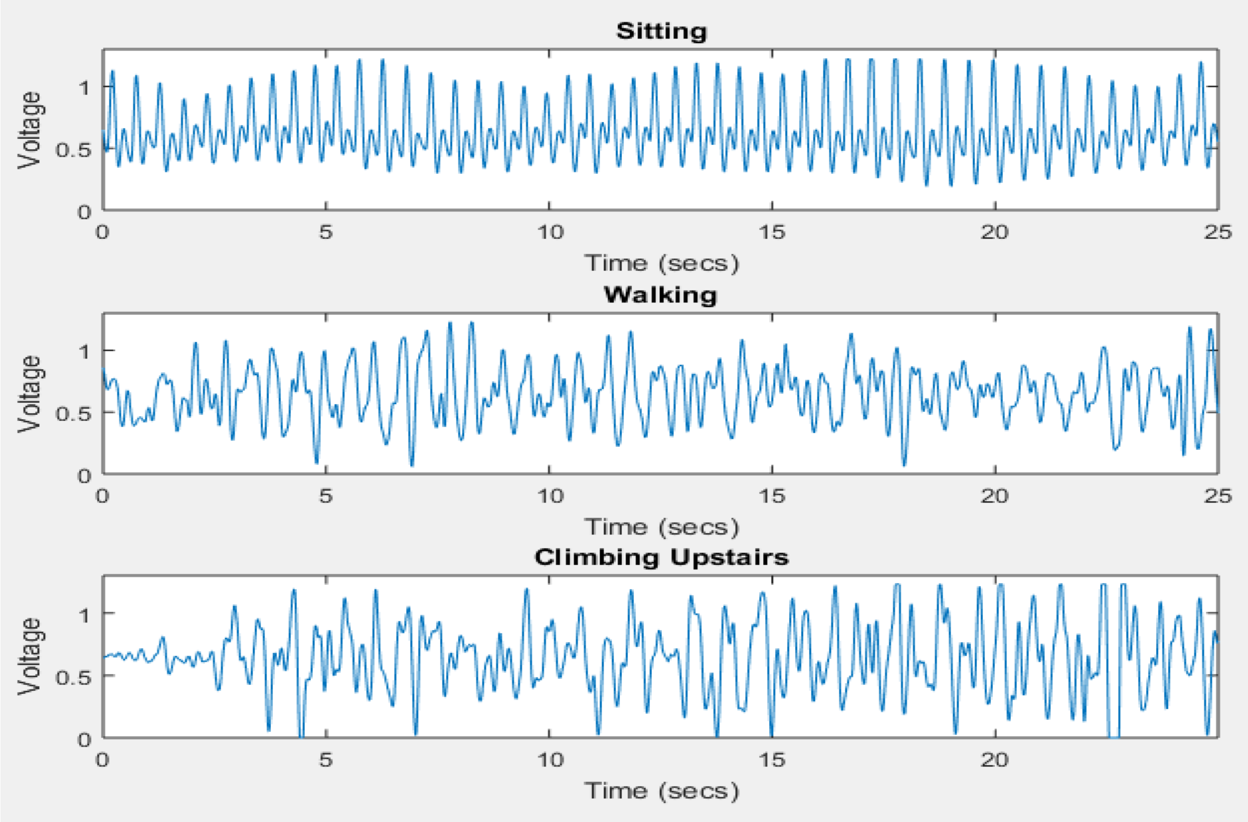


Fig. 18: Test subject 1 ECG data

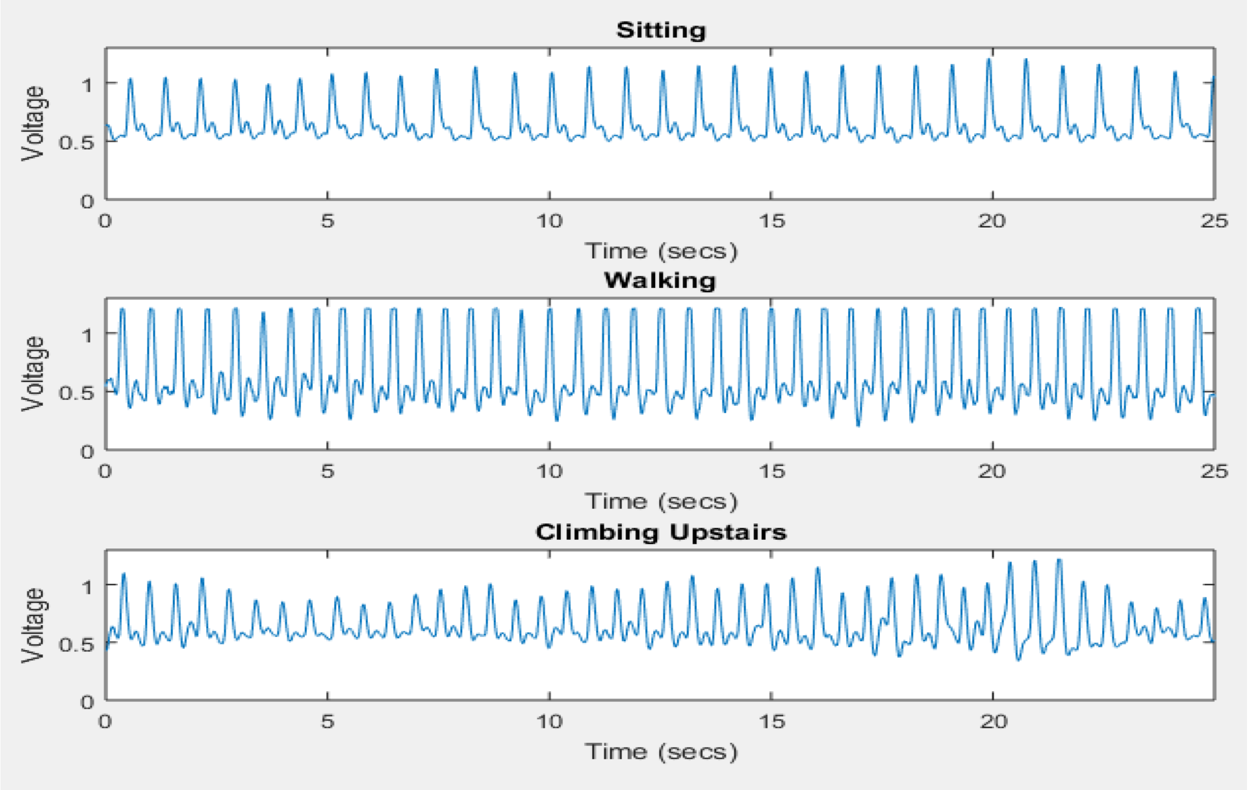


Fig. 19: Test subject 2 ECG data

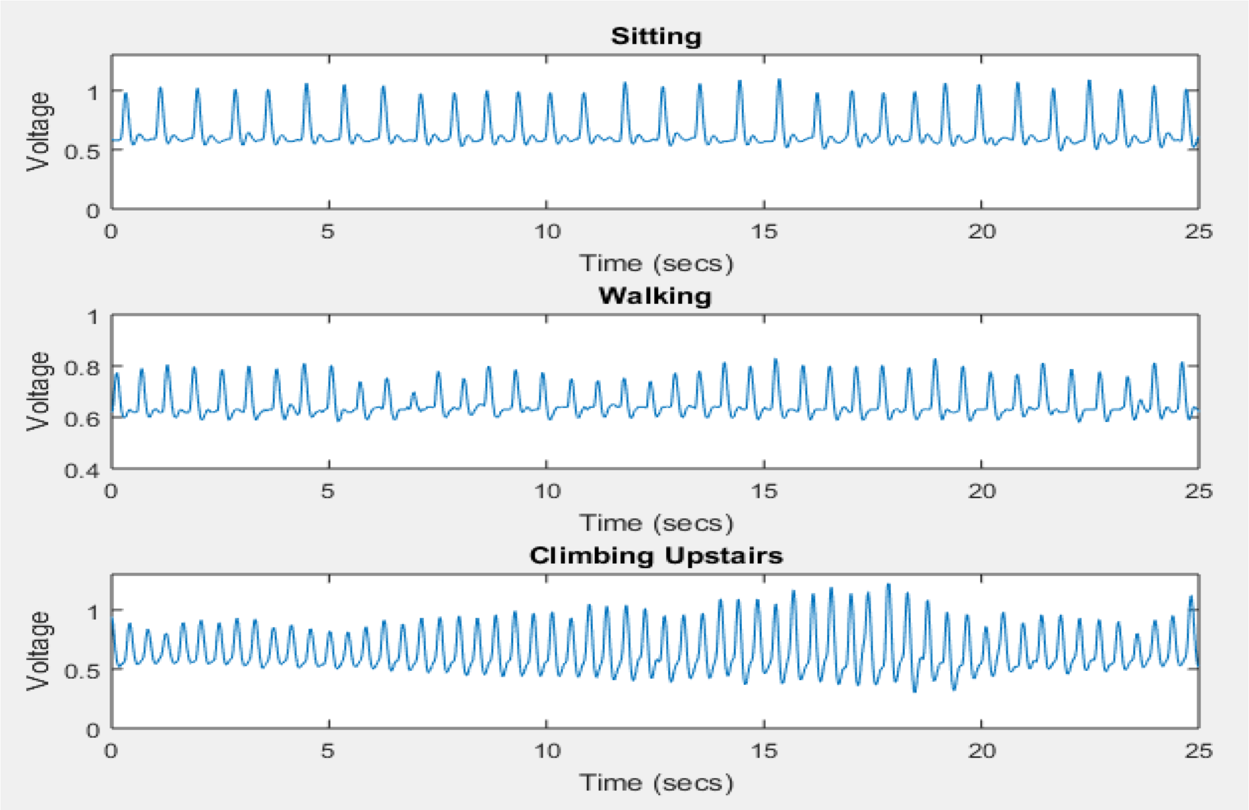


Fig. 20: Test subject 3 ECG data

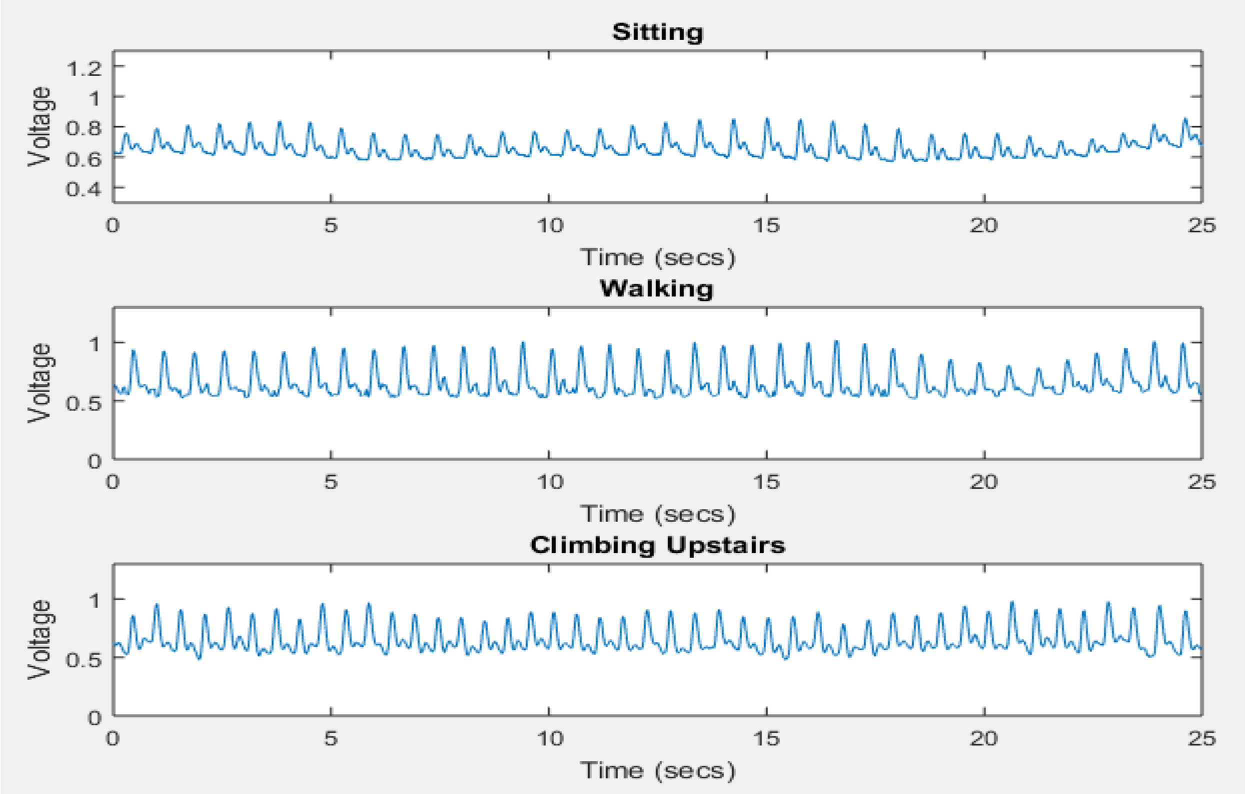


Fig. 21: Test subject 4 ECG data

Chapter 6

Data Analysis Techniques

The purpose of our data analysis is to show the relationship between heart behavior, body temperature, and the criticality of patients' statuses. Our data analysis was mostly done using MATLAB as it is a very useful tool that provides many functions to help with signal processing and data analysis. Before getting into the main focus of this section's discussion, we talk briefly about the essential aspects that all electrical engineers who deal with signals should know.

When we talk about a signal, we must always consider the noise that accompanies that signal. In signal processing, noise is a general term for unwanted alterations that a signal may suffer during collecting, storing, transmitting, or processing data [51]. In our case, we are collecting data from analog sensors and transmitting them over a LE Bluetooth communication channel. The reading can be affected through the process of the transmission and that is why we need data enhancement techniques before we can start analyzing the data. The reason why this is very important is because a noisy signal has information that does not accurately describe a signal and which can very much affect the decision. Since temperature values do not usually have many fluctuations, we are more concerned about the enhancement of the ECG signals. This motivated us to study the different biomedical engineering analysis techniques so that we can apply them to the collected data. One thing we can do with temperature plots is smooth them to give them a better visual representation but, other than that, all the data analysis techniques will focus on ECG enhancement.

6.1 Noise Reduction: Filtering

In this section, we discuss the noise reduction techniques and filtering. Before analyzing the ECG signals to extract features, we had to remove the noise to reduce false outputs in feature extraction. Extracting features from a noisy signal can give a heart rate of 200 when the actual heart rate is 80. Therefore, we ensure that, before we send our signal to the feature extraction method, almost every unwanted part of the signal is removed. There are two types of noise with which we deal in ECG signals that will be described in detail in next subsections.

6.1.1 Baseline Wander Removal

We begin with the most common problem usually found in ECG signals, namely, baseline wander. This is a problem that shows ECG signals in a wavy fashion rather than being more of a constant envelope. Reasons for this phenomenon may be due to respiration or the motion of the patients or the instruments. The technique that has been proven to work best with this is applying a high pass filter to the signal. This improves the “look” of the signal because it removes the low frequency component that manifests itself as a sine-like pattern of the baseline. This is shown clearly in Fig. 22. The figure is divided into three subplots: the first one shows the original signal; the second one shows the low frequency component that needs to be removed; and the third one shows the result after removing that low frequency component. Removing the baseline wander gives a better signal which can help us process data more accurately.

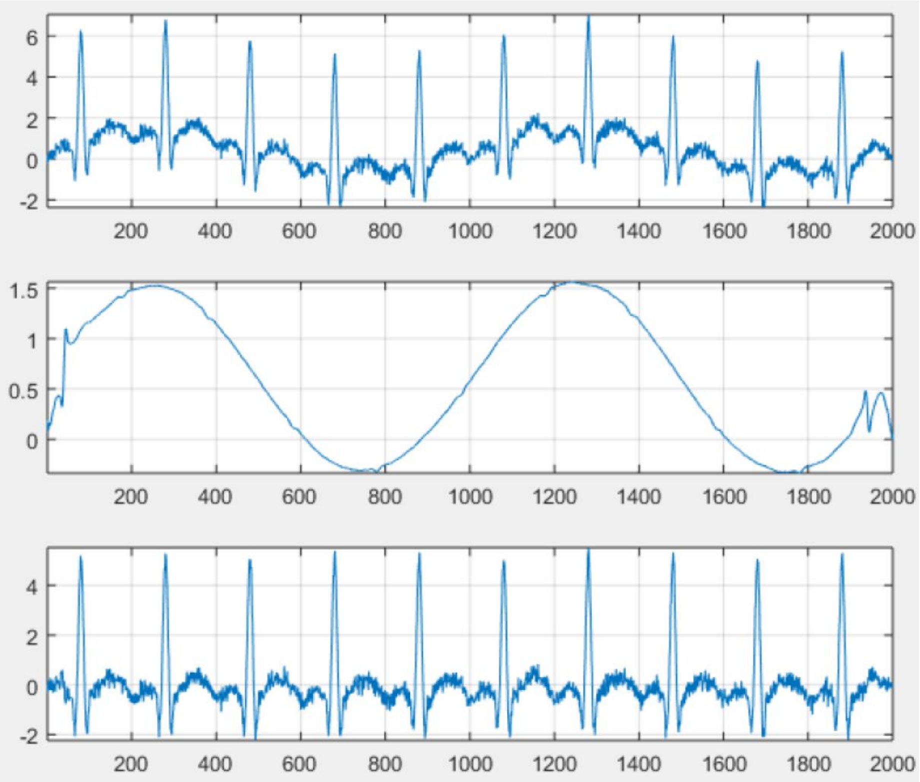


Fig. 22: ECG baseline removal

The mathematical equation for the operation of high pass filtering can be done using Eq. (6.1- 5) where w_c is the cut off frequency and N is the filter order:

$$|H\ (\omega)|^2 = \frac{1}{1+\ (\frac{\omega_c}{\omega})^{2N}} \qquad (6.1\text{- }5)$$

Using a powerful tool like MATLAB allows us to use time domain operations that also achieve the process of high pass filtering. We used two simultaneous equations to remove the baseline wander. First, we smooth the signal using the MATLAB built in function ‘smooth’, which gives us that sine-wave-like signal, then we subtract that sine-wave-like (low frequency component) from the original signal.

6.1.2 Removal of High-Frequency Component

Another common problem is high-frequency noise that would need a low pass filter rather than a high pass filter. Another solution to that would be applying a synchronized average on the periods of the signal but, for real time applications, a low pass filter would be a better option. The time domain operation of a low pass filter for signals is the mathematical operation called the moving average (often addressed to as smoothing). Figure 23 shows a signal distorted with high frequency components and the enhanced version of it. The distorted signal has major fluctuations that alter the details of the original signal. The enhanced version was achieved by applying a low pass filter with a very satisfying result as can be seen in the plot. The key when using high pass or low pass filters is to choose the correct cut-off frequency. Choosing the wrong cutoff frequency can result in huge alterations in the signal and irrelevant or, worse, erroneous data decisions. The mathematical equation for the operation of low pass filtering can be done using Eq. (6.1- 6).

$$|H\ (\omega)|^2 = \frac{1}{1+\ (\frac{\omega}{\omega_c})^{2N}} \qquad (6.1\text{- }6)$$

Again, MATLAB allows us to use time domain operations that are equivalent to this equation. For this case, all we did was apply a moving average which is achieved by using the smooth function in MATLAB. Using the correct window for smoothing is essential as it can affect the signal’s expected output. For the ECG signal shown in Fig. 23 we used a smoothing window of 20 data points.

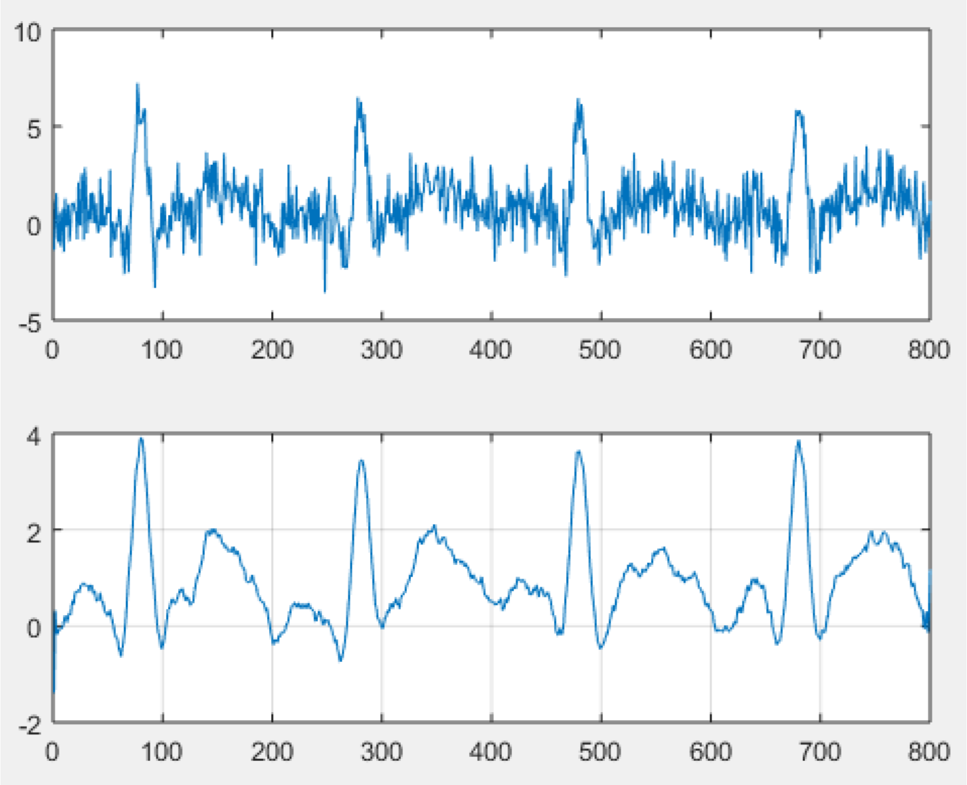


Fig. 23: ECG smoothing

6.2 Extracting Features

Now that we have clean signals, we are ready to extract features that accurately describe the ECG signals. We extract three features from the ECG signal, heart rate, RR intervals, and ST segments. Our algorithm takes those three features along with the body temperature and starts the prediction process. In the next subsections, we describe how we extracted the features from the ECG signal in detail.

6.2.1 Heart Rate

One major feature that needs to be extracted from ECG signals is a person's heart rate or beats per minute. This can be calculated using several techniques including taking the number of QRS peaks in a given time, using autocorrelation, or using Fourier transform. The first technique can sometimes yield inaccurate results, however, when a signal has no baseline wander problem, this technique should work perfectly fine. Autocorrelation and Fourier transform techniques yield very accurate results.

6.2.1.1 Autocorrelation

Autocorrelation is a mathematical operation in which a signal is correlated with a shifted copy of itself as a function of delay or lag. Correlation basically indicates the similarity between observations as a function of the time lag between them. This process works well for our case since ECG signals are periodic. We can tell the heart rate from the autocorrelation of an ECG signal by a series of simple steps using MATLAB. Figure 24 shows a plot of the autocorrelation of an ECG signal. First, we calculate the difference between two peaks which gives use the length of one period in data points. Dividing that number of data points by the sampling frequency gives us the time in seconds of one period. Inversing that and multiplying it by 60 gives us the total beats per minute.

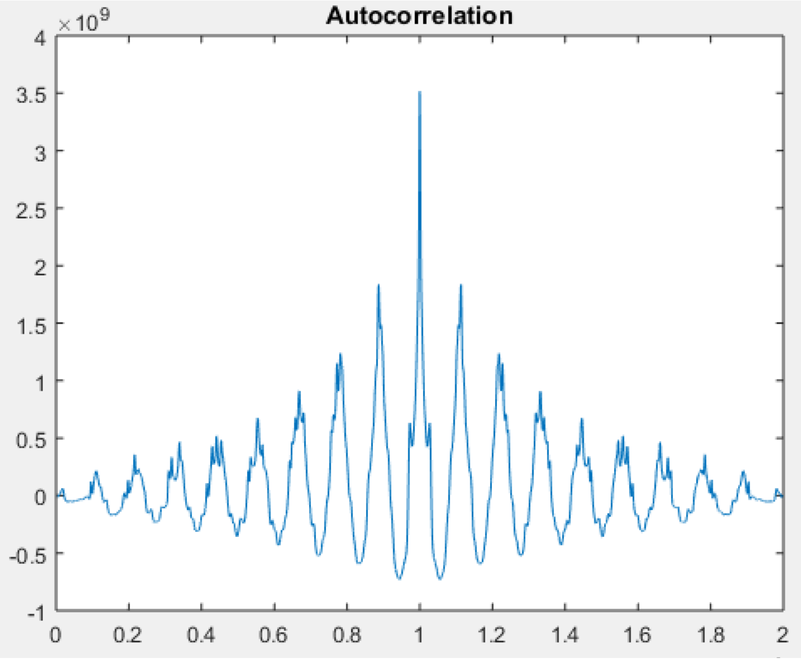


Fig. 24: Autocorrelation of an ECG signal

The mathematical equation for the autocorrelation function for signal processing is shown in Eq. (6.2- 7).

$$R(k) = \sum_{n=N1}^{N2-k} x(m) * x(m+k) \qquad (6.2\text{- }7)$$

The equation shows the summation of the product of a signal (x(m)) and a shifted version of it (x(m+k)). From the equation, one can intuitively understand that at lag zero, the signal will have the highest amplitude since it is a multiplication of itself without any shift. The MATLAB result in Fig. 24 shows the theoretical lag of zero at lag 1 on the plot.

6.2.1.2 Fourier Transform

Given that the Fourier Transform extracts the frequencies and harmonics of any signal, all we need to do is find the location of the maximum harmonic in the frequency plot as shown by the red circle in Fig. 25.

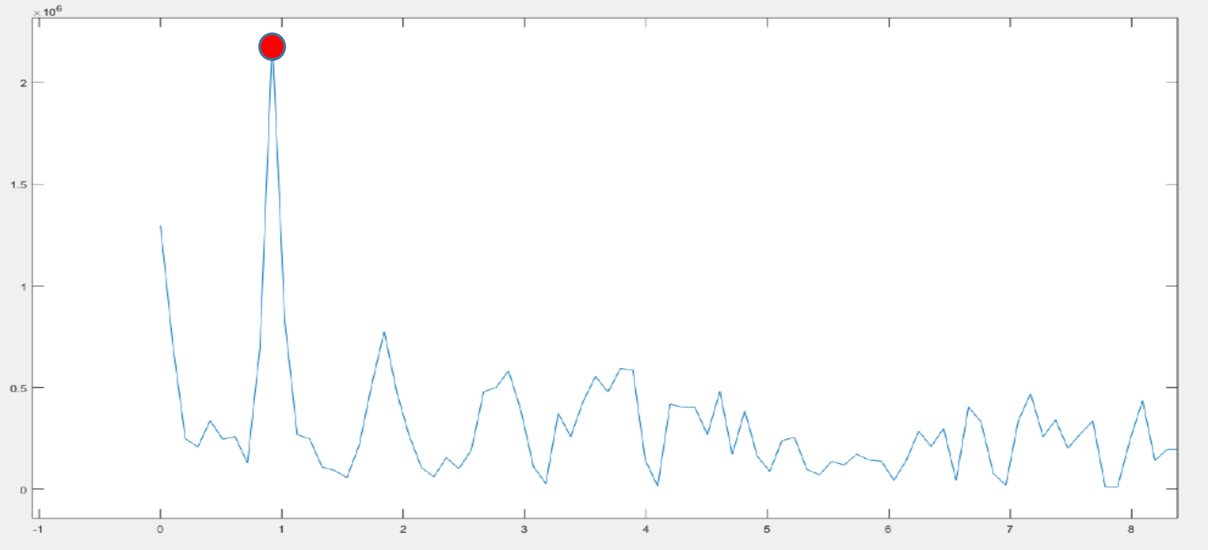


Fig. 25: Fourier transform of an ECG signal

The first significant harmonic in the signal is shown approximately around 0.92 (the red circle), which represents the beats per second. Simply multiplying it by 60 gives us the beats per minute. The other peaks in the signal represent either noise or information are irrelevant in terms of calculating the heart rate. The mathematical equations for the Fourier and inverse Fourier transforms are shown below in Eq. (6.2- 8) and Eq. (6.2- 9), respectively [52].

$$F(\omega) = \int_{-\infty}^{\infty} f(t) * e^{-i\omega t} dt \qquad (6.2\text{- }8)$$

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) * e^{i\omega t} d\omega \qquad (6.2\text{- }9)$$

$F(\omega)$ is the frequency domain of a given signal and $f(t)$ is the time domain of the signal. To go from frequency domain to time domain or vice versa, we multiply the signal by the exponential component shown in the equations. However, for our data analysis, we used a the "fft" function in MATLAB that gives us the plot of the signal in the frequency domain. From there, we get the location of the maximum harmony and multiply it by 60 to get the beats per minute.

6.2.2 RR Intervals

Another feature that we extracted from the ECG signal is called the RR interval, which is the interval between successive R peaks in an ECG signal. For normal ECG signals, the RR intervals do not fluctuate or suddenly change in a drastic manner. We recorded RR intervals by having a standard deviation analysis that tells us the amount by which the RR intervals fluctuate with time. We will discuss the method by which we determine problems with RR intervals when we present our algorithm. For now, we will explain the extraction method for the RR intervals from the ECG signals. Figure 26 gives a visual representation of an RR interval.

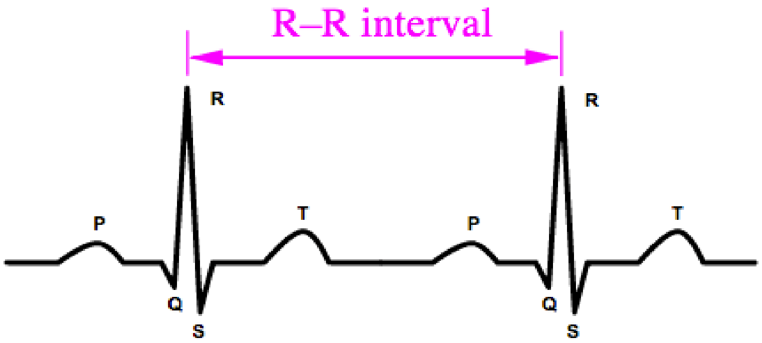


Fig. 26: RR interval [53]

We basically find the R peaks and subtract the locations of them in time, giving us the duration between each beat. We find the peaks using a threshold value that ensures that all the R peaks are included. To do that, we get the maximum of the signal and subtract it by a specified percentage to ensure that all the intervals are above the threshold value. The reason for this was because not all the R peaks have the same voltage value, the voltage values of the peaks usually fluctuate which is why we dynamically calculate that threshold value based on the portion of the ECG signal with which we are dealing. From this point, we create arrays that store the RR intervals of the ECG signal to calculate the variability of the durations.

6.2.3 ST Segments

Another feature we extracted from the ECG signals is the ST segment voltage values. An ECG signal is divided into points with each segment describing what each interval represents. We need to take the ST segment into consideration for heart attack predictions since elevated ST segments are one of the biggest indicators of heart attacks. Figure 27 shows the difference between a normal ECG and an ECG with an elevated ST segment.

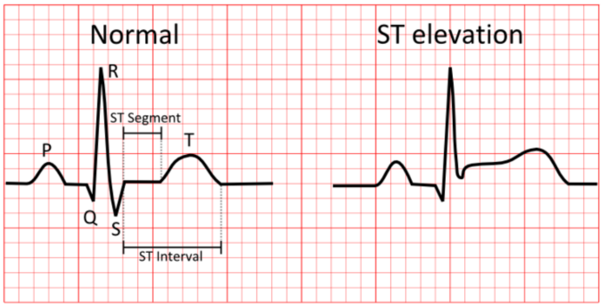


Fig. 27: Elevated ST segment [54]

Figure 28 shows sample data from one of our test subjects. To calculate the ST segment voltage value, we take the average of the points shown in the rectangle. This produces a number that represents the ST segment voltage value. The RR interval is basically the range between both peaks. We take a 20 percent from that range and add it to the location of the first peak which gives us the point where we would start adding the voltage values. Then we take 50 percent of the range and subtract it from the location of the subsequent peak, which gives us the point where we would stop adding the voltage values. Those voltage values are shown in the box in Fig. 28. After adding all the voltage values, we divide by the number of points to get the average voltage value representing the ST segment. Typically, the voltage values of a normal ECG would be much lower than the voltage values of an ECG with an elevated ST segment. We also use a standard deviation analysis to detect if an ST segment suddenly changed. Note that using percentages of the RR interval to get the locations of the ST segment voltage values and then averaging them is not a conventional way to calculate the voltage value of the ST segment. This is based on our analysis, which used trial and error, and that method to extract the ST segment voltage value provided us with the best results.

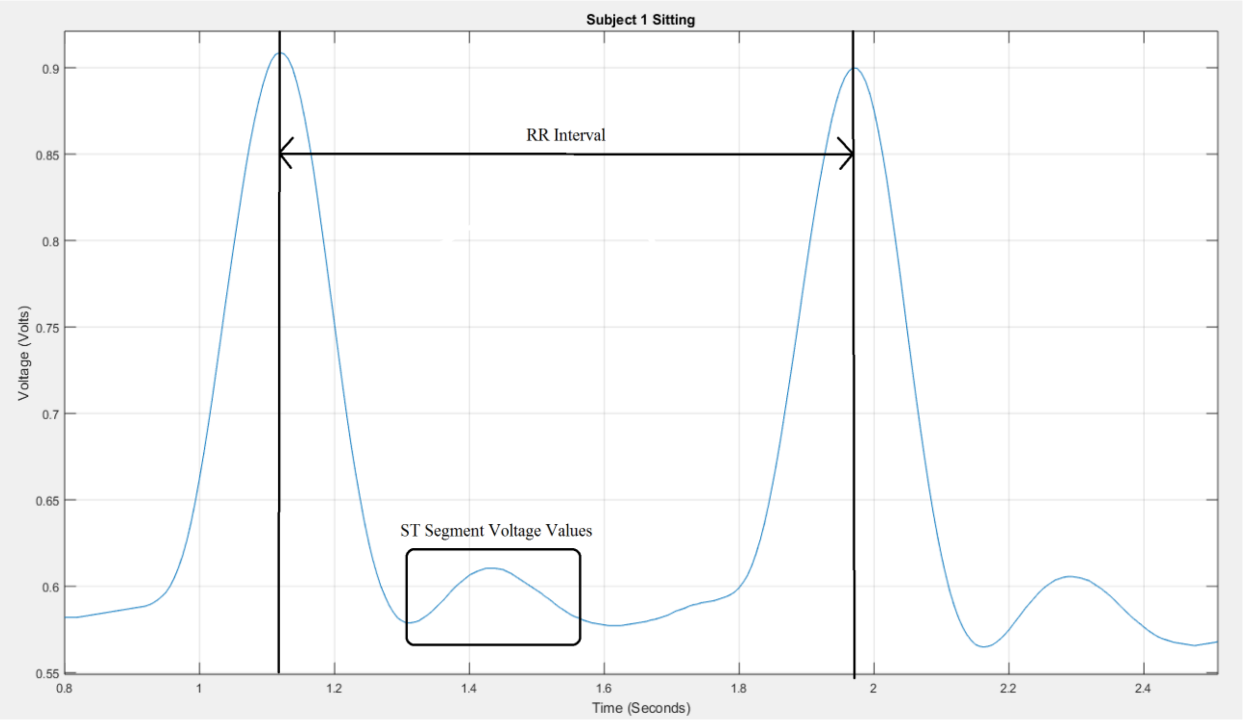


Fig. 28: Sample ECG with RR interval and ST segment

6.3 Algorithm

The algorithm is the most important part of the system. The algorithm functions as shown in the flow chart in Fig. 29. The first step is to read the data from the sensors at 5 Hz for the temperature data and 160 Hz for the ECG data. We then maintain a sampling window of 5 seconds on which to perform all computations. After selecting the sample window, we reduce the noise by applying the filtering techniques discussed in Chapter 6. After removing all the noise components from the signals, we extract the three features from the ECG and pass on those features along with the temperature data to our prediction algorithm. If the results from the algorithm indicate that the current sample window is normal, the window shifts by 1 second and takes the next 5 seconds of data. If the algorithm detects an abnormality, it immediately warns the user. Using a moving window of 1 second creates the need more computation but it provides faster and more accurate feature extraction and prediction results. This means the next sample window will have 1 second of new data and 4 seconds of data from the previous sample window.

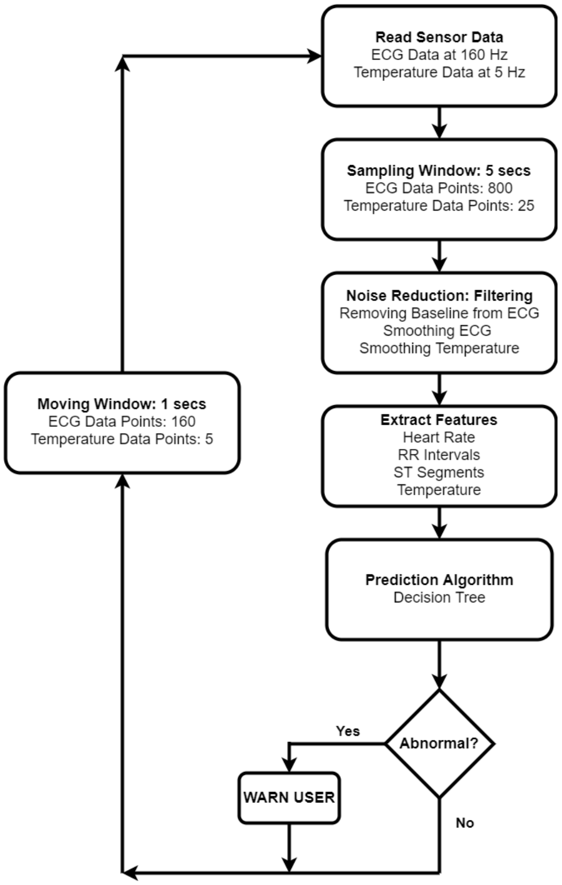


Fig: 29: Algorithm block diagram

Our prediction algorithm is based on a predictive machine-learning model called J48 Decision Tree [55]. This model decides the target value of a new sample based on various attribute values of the available data. We apply that model to our algorithm with the result that the target value would indicate whether the patient is having a heart attack or not and the available data would be contained in the extracted features. We note that the decision tree is a general model that can be used in many applications in many different ways. We created our own unique algorithm that uses the decision tree model with a standard deviation statistical analysis which gives our algorithm the unique name: Heart Attack Prediction using a **Decision Tree based on a Standard Deviation Statistical Analysis** (DTSDSA). Now we examine the method by which the extracted features are processed at the decision tree. Using a standard deviation statistical analysis, we determine whether the features are abnormal or not. Figure 30 shows the structure of our decision tree which refers to the prediction algorithm block in Fig. 29. Our algorithm uses warning levels from 0 to 4 to determine the degree of abnormality for each incoming window.

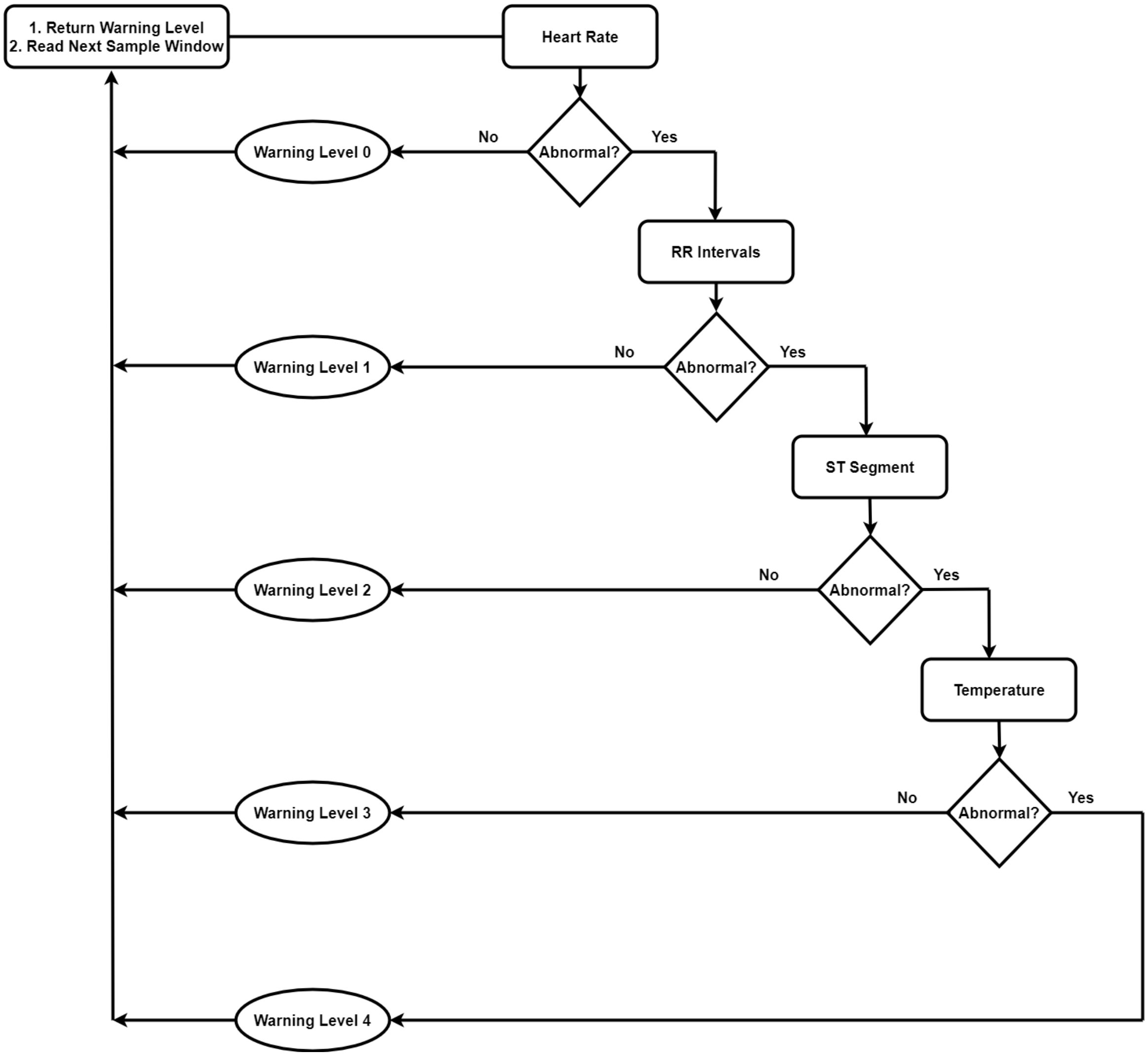


Fig. 30: Decision tree

We employ a sample window and a moving window. The sample window contains the part of the ECG signal that is being processed while the moving window specifies the amount by which that sample window is shifted to start taking the next sample window. Figure 31 illustrates the appearance of both of the windows on one of our test subjects for both sensors. As shown in Fig. 30, the sample window is 5 seconds and the moving window is 1 second. This provides an overlap of 4 seconds for subsequent sample windows. We note that for the 30 second ECG signal shown below, if we did not have a moving window, we would have only had 6 sample windows (30 seconds / 5 second windows). This means that the features would only be updated 6 times throughout the entire 30 seconds. The way we implemented it, we get 26 results instead of 6 for the entire 30 seconds. This represents a far more practical method since heart rates change very fast, especially during cardiac events.

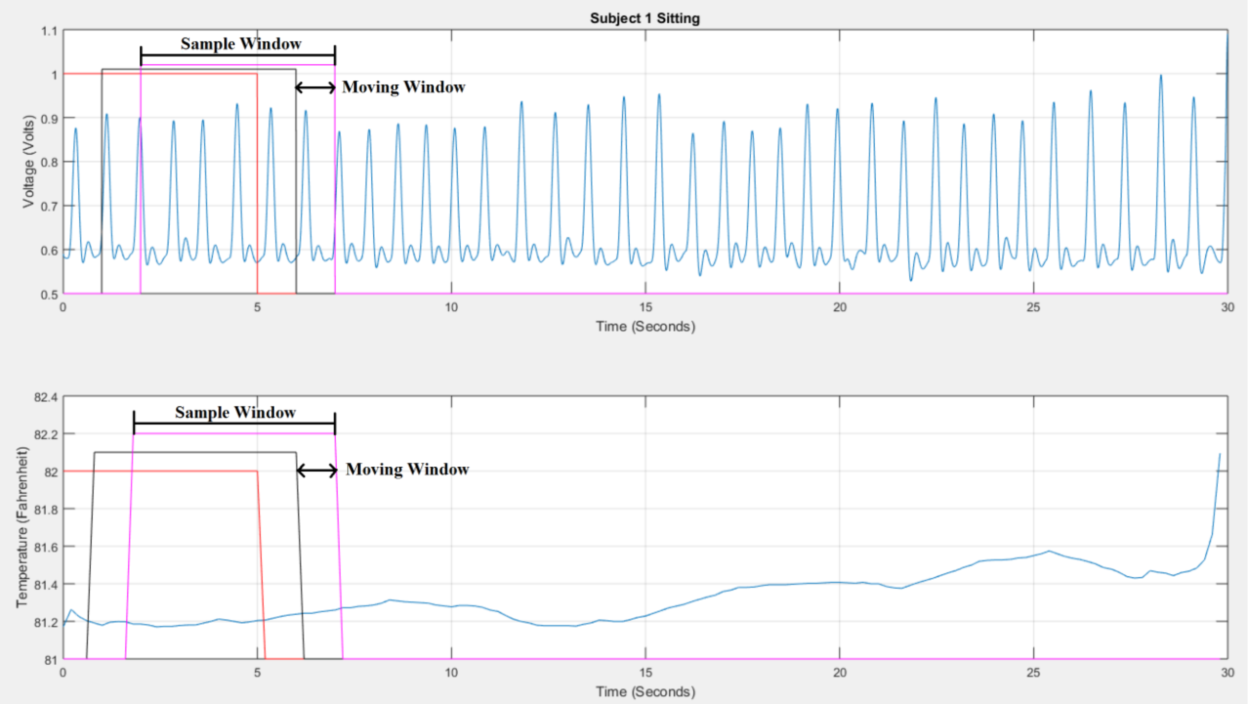
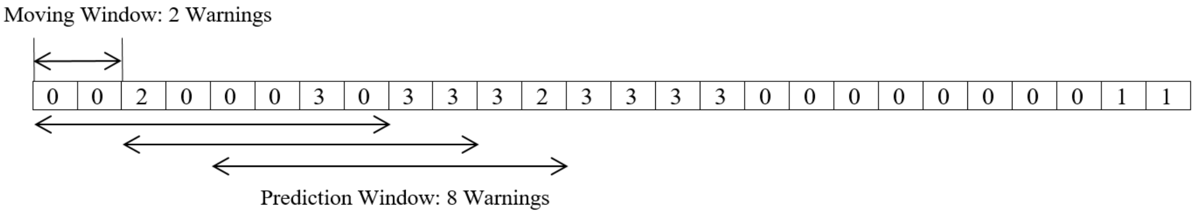


Fig. 31: Illustration of sample and moving windows

For each sample window, the feature extraction function returns a single value for the heart rate, in a one-dimensional array with the RR interval durations, and a one-dimensional array with the ST segment voltage values. Since heart rates are the most important feature that describe the heart's status, we start by checking variations in the heart beats first. We do so by making sure that the heart rate is consistent using our standard deviation analysis. Any heart rate while walking or running is obviously going to be higher than the heart rate while sitting or resting. Since we have a wide range of heart rates that are considered normal, we were not able to simply apply a thresholding technique where a heart rate above a certain threshold value would be a sign of potential heart failure. Heart rates can vary from 55 all the way to 150 depending on the person and what the person is doing. By using our standard deviation statistical analysis, we only detect an issue with the heart rate when it suddenly fluctuates out of the normal range. If the current heart rate has an error above 7 percent, we set the warning level to 1. For example, if a person's average heart rate is between 80 beats per minute for 20 seconds then suddenly goes up to 100, the error would be 25 percent. We only proceed to check the RR intervals if there is a problem with the current heart rate. For the RR intervals and ST segments arrays, with which we are dealing, we calculate the standard deviation of the sample window for both features. If the RR intervals' error is higher than a certain percentage, we set the warning level to 2 and proceed to check the ST segment. If the ST segment also has an error higher than what is considered to be normal, we set the warning level to 3 and proceed to check the body temperature. At this point, we already know that this sample window is abnormal. We still check the body temperature to see if the warning level would go up to 4 or not since up to this point, it can be a false reading based on errors in feature extraction due to noisy signals. Since the temperature is a single value, we calculate the error the same way we did for the heart rates only with different thresholds. We then return the warning level for each sample window to process that warning and read the next sample window. We created a dynamic buffer that attends to the processing of warnings that are returned for each sample window. The buffer is responsible for collecting the warning levels and making a decision. To implement the buffer, we created another window called the prediction window along with a moving window. This window initially waits to collect the results from 8 sample windows (8 warnings). The moving window then shifts the prediction window 2 spots to the right. A decision is made on each prediction window based on a ratio that is calculated from the warning levels. Figure 32 shows the technique by which the prediction and moving windows are established. The moving window is equivalent to 2 warnings and the prediction window is equivalent to 8 warnings, which results in 10 prediction windows for the 30 second segment.



0	0	2	0	0	0	3	0	3	3	3	2	3	3	3	3	0	0	0	0	0	0	0	0	1	1

0	0	2	0	0	0	3	0	Ratio = 5 / 24 = 0.2083	
2	0	0	0	3	0	3	3	Ratio = 11 / 24 = 0.4583	
0	0	3	0	3	3	3	3	Ratio = 15 / 24 = 0.5833	WARNING
3	0	3	3	3	3	2	3	Ratio = 20 / 24 = 0.8333	WARNING
3	3	3	3	2	3	3	3	Ratio = 23 / 24 = 0.9583	WARNING
3	2	3	3	3	3	0	0	Ratio = 17 / 24 = 0.7083	WARNING
3	3	3	3	0	0	0	0	Ratio = 12 / 24 = 0.5	WARNING
3	3	0	0	0	0	0	0	Ratio = 6 / 24 = 0.25	
0	0	0	0	0	0	0	0	Ratio = 0 / 24 = 0	
0	0	0	0	0	0	1	1	Ratio = 2 / 24 = 0.0833	

Fig. 32: Algorithm results using prediction window

Assuming that the body temperatures are normal, the worst case would be a prediction window with all 3's which gives a sum of 24. We add all the warning levels and divide by 24. If the ratio is 0.5 or above, we trigger a warning to the user. The results shown in Fig. 32 are from an ECG signal that was very noisy and did not have any characteristics of a proper ECG. The algorithm therefore started detecting abnormalities in the third prediction window as shown in Fig. 33. Running this algorithm on normal ECG's for healthy subjects gave us ratios that were either zero or close to zero. Those were our first indications that the algorithm does indeed work. However, our next step was to run the algorithm on real test subjects with heart failures for more validation. The results are shown and discussed in more detail in the next chapter.

Chapter 7

Results and Evaluation

In this chapter, we present some of the results we obtained from our algorithm. We showed the results of one test subject at the end of the previous chapter. The data was for Subject 7 while the subject was asked to walk and as you can see in Fig. 33, the signal was very noisy. This gave us an indication that the algorithm does in fact yield results that make sense. However, we still needed to validate our algorithm with real test subjects that suffered from heart failures.

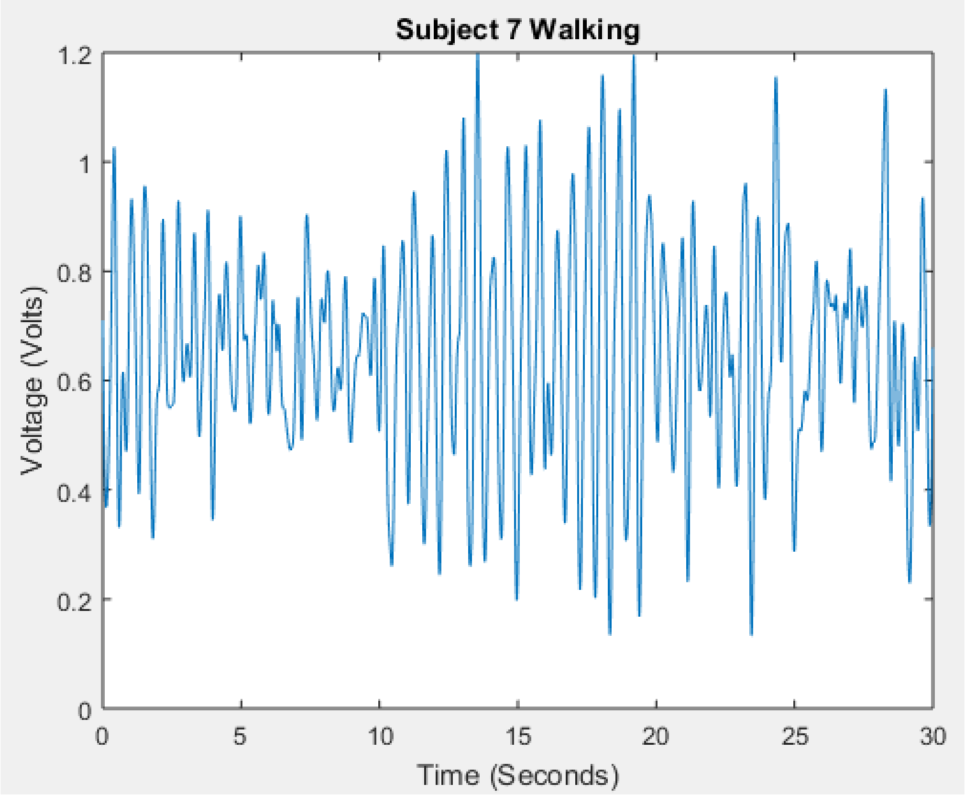


Fig. 33: Subject 7 walking

We were able to download datasets from a database online that has records of patients who suffered from sudden cardiac deaths. Before discussing that aspect of our research, we show some results for the healthy test subjects that we collected using our system. We show the results in the same form of the results shown in Fig. 32.

7.1 Healthy Test Subjects

The results shown are for one test subject in the three different scenarios. Since all subjects had normal body temperatures, we will show the ECG signals and the results of the prediction algorithm for each sample window. The test subject's information is shown in Table 8.

Table 8 Information for test subject 1

Subject	Gender	Age	Scenario	Average Heart Rate
1	Male	24	➢ Sitting ➢ Walking ➢ Climbing Upstairs	➢ 84 ➢ 108 ➢ 135

Figure 34 shows the ECG signal of Subject 1 in the sitting scenario. This ECG is normal and, therefore, the algorithm triggered no warnings. We will start showing the prediction algorithm results for the next two cases since there were no warnings for this case where the subject was sitting.

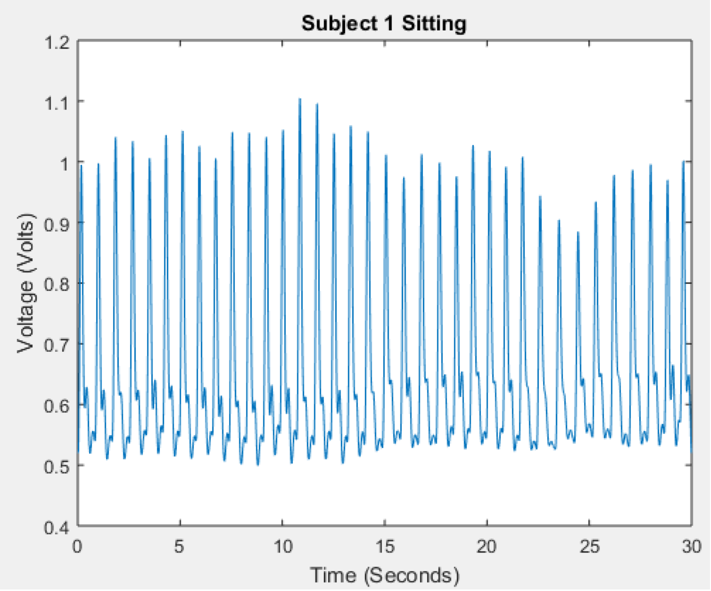


Fig. 34: Subject 1 sitting

Figure 35 shows the ECG signal of the same subject while walking. This ECG was also normal. It had a couple of warnings that are considered to be false errors. This is why we have the prediction window, to eliminate the false warnings.

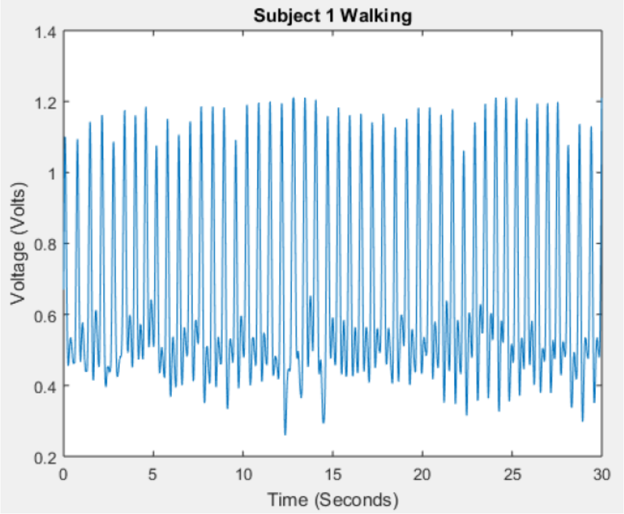
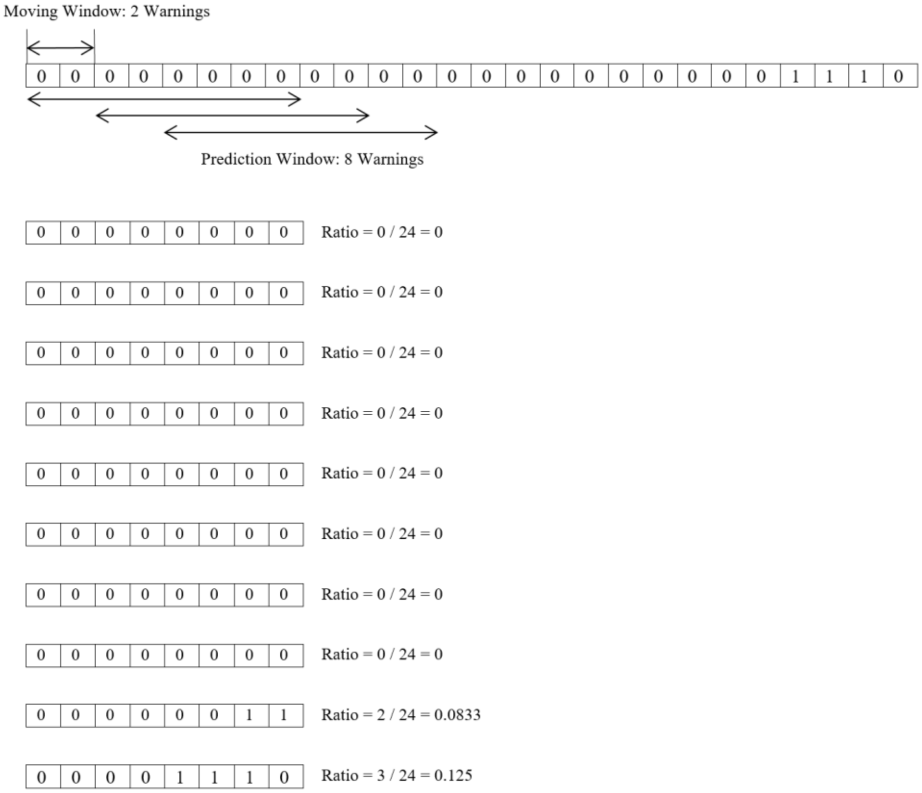


Fig. 35: Subject 1 walking

Figure 36 shows that the results from the prediction algorithm had three warnings of level one. Therefore, there was no need to warn the user since it was a false error.


Moving Window: 2 Warnings

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0

Prediction Window: 8 Warnings

0 0 0 0 0 0 0 0 Ratio = 0 / 24 = 0

0 0 0 0 0 0 0 0 Ratio = 0 / 24 = 0

0 0 0 0 0 0 0 0 Ratio = 0 / 24 = 0

0 0 0 0 0 0 0 0 Ratio = 0 / 24 = 0

0 0 0 0 0 0 0 0 Ratio = 0 / 24 = 0

0 0 0 0 0 0 0 0 Ratio = 0 / 24 = 0

0 0 0 0 0 0 0 0 Ratio = 0 / 24 = 0

0 0 0 0 0 0 0 0 Ratio = 0 / 24 = 0

0 0 0 0 0 0 1 1 Ratio = 2 / 24 = 0.0833

0 0 0 0 1 1 1 0 Ratio = 3 / 24 = 0.125

Fig. 36: Prediction algorithm results for subject 1 while walking

The algorithm triggered some warnings for the case where the test subject was climbing upstairs. Figure 37 shows the ECG for that scenario.

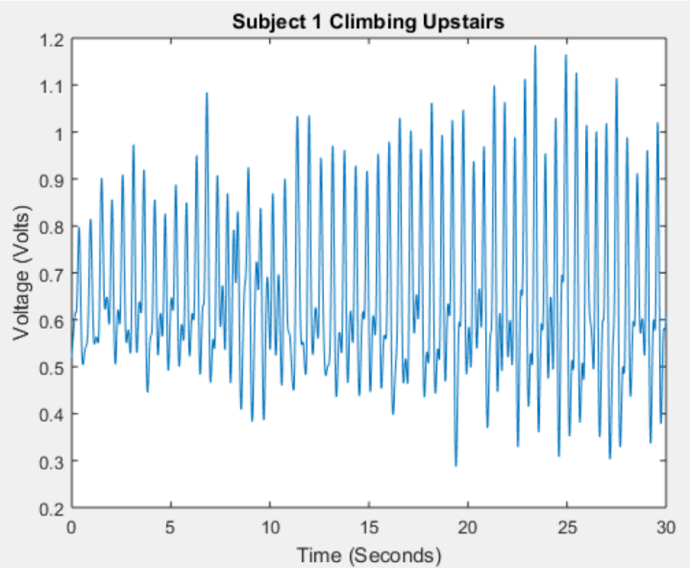
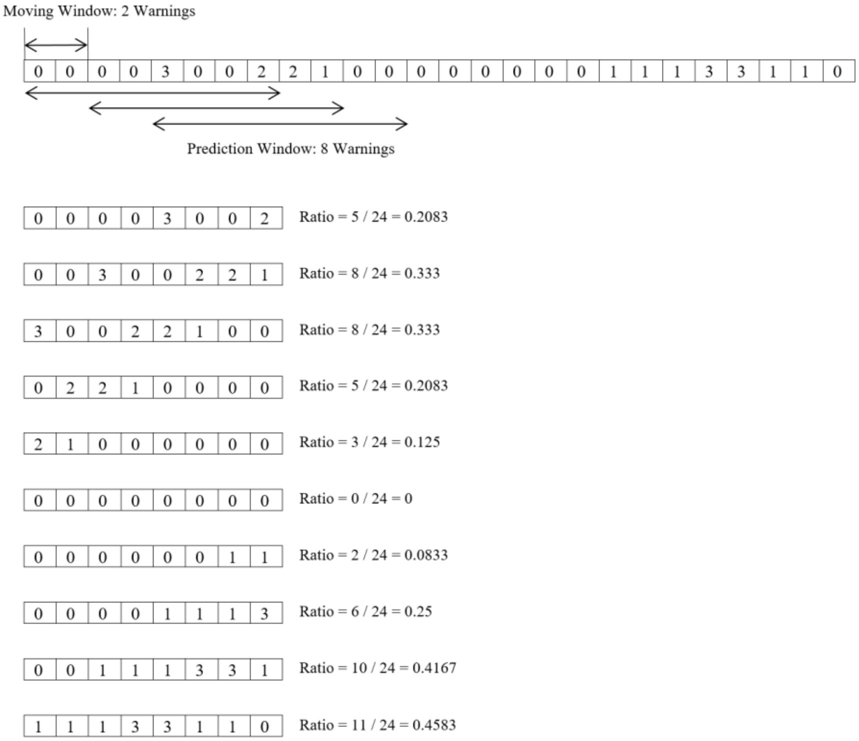


Fig. 37: Subject 1 climbing upstairs

As shown in Fig. 38, the results show that there are a few warnings for each prediction window none of which passed 50 percent. We know that the test subject was not having a heart attack since we collected the data.


Moving Window: 2 Warnings

0 0 0 0 3 0 0 0 2 2 1 0 0 0 0 0 0 0 0 1 1 1 3 3 1 1 0

Prediction Window: 8 Warnings

0 0 0 0 3 0 0 2 Ratio = 5 / 24 = 0.2083

0 0 3 0 0 2 2 1 Ratio = 8 / 24 = 0.333

3 0 0 2 2 1 0 0 Ratio = 8 / 24 = 0.333

0 2 2 1 0 0 0 0 Ratio = 5 / 24 = 0.2083

2 1 0 0 0 0 0 0 Ratio = 3 / 24 = 0.125

0 0 0 0 0 0 0 0 Ratio = 0 / 24 = 0

0 0 0 0 0 0 1 1 Ratio = 2 / 24 = 0.0833

0 0 0 0 1 1 1 3 Ratio = 6 / 24 = 0.25

0 0 1 1 1 3 3 1 Ratio = 10 / 24 = 0.4167

1 1 1 3 3 1 1 0 Ratio = 11 / 24 = 0.4583

Fig. 38: Prediction algorithm results for subject 1 while climbing upstairs

7.2 Unhealthy Test Subjects

The algorithm was able to account for different heart rates as shown for Subject 1 in the previous chapter. We also ran the algorithm on our 10 healthy test subjects and the results validated that the algorithm works very well for healthy test subjects. The results showed that the algorithm gives no warnings for all scenarios that had different heart rates. However, as mentioned before, validating our algorithm using only healthy subject data is not enough. Even though we ran our algorithm on noisy data, we still cannot say for sure that our algorithm can predict heart problems. Therefore, we downloaded 10 datasets from a database online that has ECG signals for patients that suffered from sudden cardiac deaths. The ECG signals we selected for each test subject was moments before the subject passed away. Table 9 shows the information of each test subject [56].

Table 9 Information of unhealthy test subjects

Subject	Gender	Age	History	Medication	Underlying Cardiac Rhythm
1	Male	43	Unknown	Unknown	Sinus
2	Female	72	Heart Failure	Digoxin; Quinidine gluconate	Sinus
3	Female	30	Unknown	Unknown	Sinus
4	Female	72	Mitral valve replacement	Digoxin	Atrial fibrillation
5	Male	75	Cardiac surgery	Digoxin; Quinidine	Atrial fibrillation
6	Male	34	Unknown	Unknown	Sinus
7	Female	89	Unknown	Unknown	Atrial fibrillation
8	Male	66	Acute myelogenous leukemia	Digoxin; Quinidine	Sinus
9	Female	82	Heart failure	None listed	Sinus
10	Male	68	History of ventricular ectopy	Digoxin; Quinidine Gluconate	Sinus

We tested the ECG signals of all the test subjects shown in Table 9 and the algorithm's results were exactly what we expected. In this final section, we show some details of the algorithm's results for test Subject 5 to clarify why the warnings were triggered. Figure 39 shows the ECG signal for Subject 5.

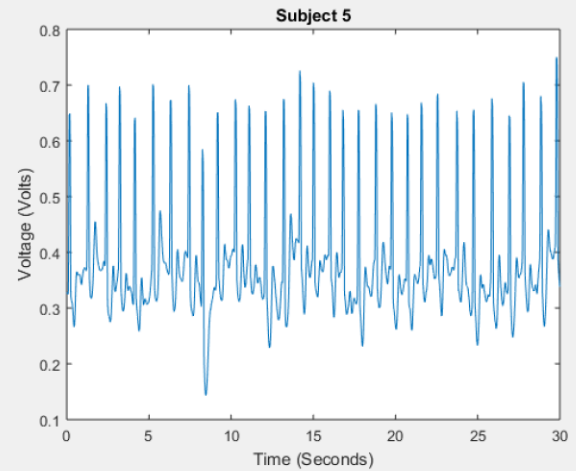


Fig. 39: ECG signal of unhealthy test subject 5

Before showing the prediction algorithm results, we will show the results from the feature extraction to see why the algorithm triggered warnings.

Heart Rates for first 11 sample windows

71.94	131.89	71.942	119.90	119.90	131.89	71.942	71.942	71.942	71.942	131.89

Sample window 2:

1) Heart Rate Error = 100 * |131.89 – 71.94| / 71.94 = 83.3 % → Warning level 1
2) As shown in Fig. 40, the RR Intervals had very high fluctuations which explains why the heart rate jumped from 71.94 to 131.89 in just one second. → Warning level 2

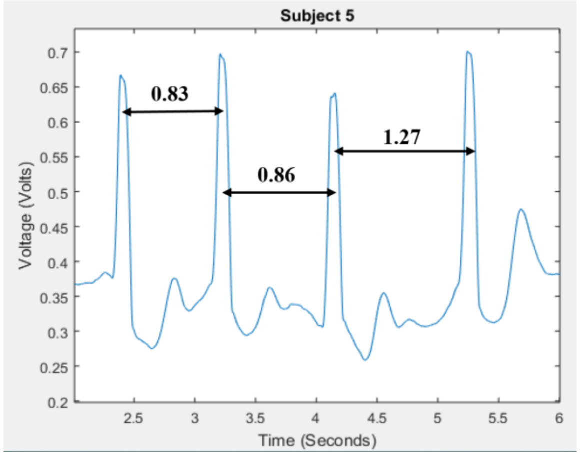


Fig: 40: RR Intervals on sample window 2

3) As shown in Fig. 41, the ST Segment voltage values were also fluctuating in an abnormal fashion. → Warning level 3

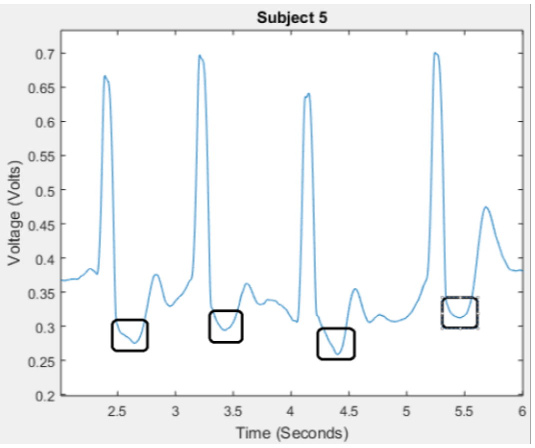
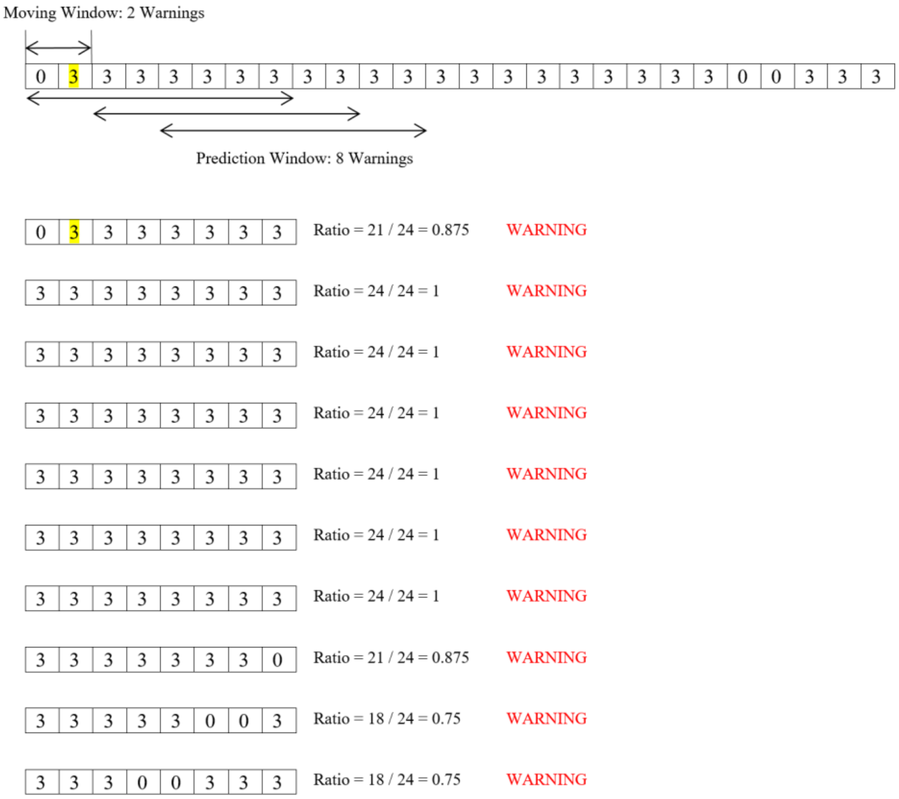


Fig. 41: ST segments on sample window 2

The prediction results for the whole ECG signal are shown in Fig. 42. The warning result from the second sample window, the one we just analyzed, is highlighted in yellow. This ECG signal had several fluctuations in all the features, which is why the algorithm triggered warnings of level 3 for almost all the sample windows. This is what we expected for a patient who had a history of cardiac surgery and passed away shortly after the signal was recorded.



0	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	3	3	3

0	3	3	3	3	3	3	3	Ratio = 21 / 24 = 0.875	WARNING
3	3	3	3	3	3	3	3	Ratio = 24 / 24 = 1	WARNING
3	3	3	3	3	3	3	3	Ratio = 24 / 24 = 1	WARNING
3	3	3	3	3	3	3	3	Ratio = 24 / 24 = 1	WARNING
3	3	3	3	3	3	3	3	Ratio = 24 / 24 = 1	WARNING
3	3	3	3	3	3	3	3	Ratio = 24 / 24 = 1	WARNING
3	3	3	3	3	3	3	3	Ratio = 24 / 24 = 1	WARNING
3	3	3	3	3	3	3	0	Ratio = 21 / 24 = 0.875	WARNING
3	3	3	3	3	0	0	3	Ratio = 18 / 24 = 0.75	WARNING
3	3	3	0	0	3	3	3	Ratio = 18 / 24 = 0.75	WARNING

Fig. 42: Prediction algorithm results for unhealthy test subject 5

Chapter 8

Conclusion and Future Work

8.1 Conclusion

We managed to build a system that is capable of collecting data and presenting it to the user in real-time as shown in the previous chapters. We also managed to create a LE consumption communication channel between the IoT device and the application. The application is developed on smartphones which are normally powered via batteries that are limited in size and capacity. The use of a LE Bluetooth chip helped to provide our service with minimal effect on the smartphone's battery life.

The smartphone application is composed of the needed user screens as described in the system's architecture section. We also considered some optimization on the computations on the smartphone's end to help with the low battery consumption. For instance, when collecting data to plot the body temperature, we discard any plotting computations that are related to the incoming data from the pulse sensor and vice versa. This allows the application to focus on one aspect and hence less computations. We also took advantage of using threads to plot the data in real-time to avoid interference with the main UI thread which gives an overall better application performance. We collected many datasets and started analyzing them as described in the data collection and data analysis techniques sections. We created an algorithm that is capable of reading a signal, dividing it into sample windows, reduce noise, extract features, and predict abnormalities using a standard deviation statistical analysis decision tree. Based on the sensed data and the condition of the patient, different levels of warnings are triggered and a conclusion is reached based on prediction windows. The algorithm's results showed that healthy test subjects had no issues. We were even able to distinguish between false readings and abnormalities by using prediction windows and calculating ratios based on the levels of warnings.

8.2 Future Work

In the future, this system can be enhanced by collecting more data from both healthy subjects and patients who have heart problems. The algorithm can be enhanced by adding more features such as the wave durations for each beat. We can also enhance the system by adding more sensors to the IoT device. Some sensors we can take into consideration include skin conductivity sensors, respiratory sensors, blood pressure sensors, and accelerometer sensors. The algorithm would obviously need to be modified to account to the features from the additional sensors. We also mentioned that the IoT device is not ideal for a selling product, so by using more advanced hardware components we can develop a more feasible device that can be introduced to the market. Finally, the prediction algorithm that we created was developed on MATLAB^{TM}. A big step towards the future involves integrating that algorithm to a computing platform where users can benefit from the algorithm using a smartphone. We can also create a warning system that uses a user's GPS information to generate an alert message to an emergency contact.

Bibliography

[1] ITU-T Global Standards Initiatives Recommendation ITU-T Y.2060 (06/2012) http://www.itu.int/en/ITU-T/gsi/iot/Pages/default.aspx, Last accessed May 2016

[2] O. Vermesan, P Friess, Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems, River Publishers Series In Communications, 2013

[3] R. Clarke, Smart Cities and the Internet of Everything: The Foundation for Delivering Next-Generation Citizen Services, Cisco 2013

[4] D. Evans, The Internet of Things How the Next Evolution of the Internet Is Changing Everything, Cisco IBSG, 2011

[5] IEEE Standards Association, P2413 - Standard for an Architectural Framework for the Internet of Things (IoT), https://standards.ieee.org/develop/project/2413.html, [Last Accessed 15 Jul 2017]

[6] IEEE Standards Association (IEEE-SA), Internet of Things (IoT) Ecosystem Study, IEEE 2015

[7] IETF, Internet Protocol Version 6 (IPv6) Specification, Network Working Group, The Internet Society (1998)

[8] B. Djamaa , R. Witty, An efficient service discovery protocol for 6LoWPANs, Science and Information Conference (SAI), IEEE 2013

[9] J. Penney, Choosing an IoT security provider, http://info.deviceauthority.com/blog-da/choosing-an-iot-security-provider, 2016

[10] B. Djamaa , R. Witty, An efficient service discovery protocol for 6LoWPANs, Science and Information Conference (SAI), IEEE 2013

[11] Internet Engineering Task Force, The Constrained Application Protocol (CoAP), https://tools.ietf.org/html/rfc7252 [Last Accessed April 2017]

[12] P. Sethi, S. Sarangi, “Internet of Things: Architectures, Protocols, and Applications”, Journal of Electrical and Computer Engineering, 2017

[13] P. Pereira, J. Eliasson, J. Delsing, "An authentication and access control framework for CoAP-based internet of things". 40th Annual Conference of the IEEE Industrial Electronics Society, 2014

[14] H. Khattak, M. Ruta, P. di Bari, “CoAP-based Healthcare Sensor Networks: a survey”, 11th International Bhurban Conference on Applied Sciences and Technology, 2014

[15] M. Kovatsch, "CoAP for the Web of Things: From Tiny Resource-constrained Devices to the Web Browser", 4th International Workshop on the Web of Things (WoT 2013), UbiComp '13 Adjunct, Sept 8-12, 2013, Zurich, Switzerland, 2013

[16] Internet Engineering Task Force (IETF), The Constrained Application Protocol (CoAP), https://tools.ietf.org/html/rfc7252, 2012

[17] M. Maksimović, V. Vujović, B. Perišic, "A Custom Internet of Things Healthcare System", Tenth Iberian Conference on Information Systems and Technologies (CISTI), 2015

[18] E. Stecker, K. Reinier, C. Rusinaru, A. Uy-Evanado, J. Jui, S. Chugh, 2017, Health Insurance Expansion and Incidence of Out-of-Hospital Cardiac Arrest: A Pilot Study in a US Metropolitan Community, Journal of the American Heart Association, Downloaded from http://jaha.ahajournals.org/ by guest on July 21, 2017

[19] M. Mancini, M. Cazzell, S. Kardong-Edgren, C. Cason, "Improving Workplace Safety Training Using a Self-Directed CPR-AED Learning Program, AAOHN J.. 2009;57:159–167).

[20] EMS Report: U.S. out-of-hospital cardiac arrest survival rates 'unacceptably low', Retrieved July 20, 2017, from https://www.ems1.com/ems-products/cpr-resuscitation/articles/2255015-Report-U-S-out-of-hospital-cardiac-arrest-survival-rates-unacceptably-low/

[21] Communicore. (1996). Sudden cardiac arrest: A treatable public health crisis. Retrieved July 20, 2017, from http://publicsafety.tufts.edu/ems/downloads/sca_whtp.pdf

[22] G. Kavitha1, E. Mariya, "Endowed Heart Attack Prediction System Using Big Data", International Journal of Pharmacy & Technology IJPT, April-2017

[23] S. Jagtap, "Prediction and Analysis of Heart Disease", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 5, Issue 2, February 2017

[24] T. Marikani, K. Shyamala, "Prediction of Heart Disease using Supervised Learning Algorithms", International Journal of Computer Applications, Volume 165 – No.5, May 2017

[25] United Nations, Department of Economic and Social Affairs, Population Division, "World Population Ageing 2009," pp. 66-71.

[26] United States, Heart Association and Stroke Association, "Heart Disease and Stroke Statistics – At-a-Glance," 2015

[27] Aaron Smith, "Older Adults and Technology Use", Internet, Science and Technology, PewResearchCenter, April 3, 2014

[28] S. Riazul Islam, D. Kwak, H. Kabir, M. Hossain, K. Kwak, "The Internet of Things for Health Care: A Comprehensive Survey", IEEE Access, Volume 3, 678 – 708, 2015

[29] B. Buckiewicz, “Overview of Medical Body Area Networks”, Retrieved fromhttps://www.lsr.com/white-papers/overview-of-medical-body-area-networks, accessed 21 July, 2017

[30] Noakes, T.D. Sudden death and exercise. In: Encyclopedia of Sports Medicine and Science, T.D. Fahey (Editor). Internet Society for Sport Science: http://sportsci.org. 8 Nov 1998

[31] Patients Like Me, https://www.patientslikeme.com/, Accessed 11 March 2017

[32] DailyStrength, http://www.dailystrength.org/, Accessed 11 March 2017

[33] Omnio, http://omnio.com/, Accessed 11 March 2017

[34] Everyday Health, http://www.everydayhealth.com/, Accessed 11 March 2017

[35] J. Lee, Y. Chuah, K. Chieng, "Smart Elderly Home Monitoring System with an Android Phone," International Journal of Smart Home, vol. 7, pp. 17-32, May 2013

[36] Y. Zhang, H. Liu, X. Su, P. Jiang, D. Wei, “Remote Mobile Health Monitoring System Based on Smart Phone and Browser/Server Structure”, Journal of Healthcare Engineering, Vol. 6, No. 4, Page 717–738, 2015

[37] P. Pawar, V. Jones, B. Beijnum, H. Hermens, “A framework for the comparison of mobile patient monitoring systems”, Journal of Biomedical Informatics, 2012

[38] Qardiocore, https://www.getqardio.com/qardiocore-wearable-ecg-ekg-monitor-iphone/, Accessed 15 April 2017

[39] https://learn.sparkfun.com/tutorials/analog-to-digital-conversion [Last Accessed July 2017]

[40] https://www.sparkfun.com/products/12576 [Last Accessed July 2017]

[41] https://robotechshop.com/shop/arduino/arduino-board/arduino-uno-r3-china/?v=7516fd43adaa [Last Accessed July 2017]

[42] https://www.sparkfun.com/products/11574 [Last Accessed July 2017]

[43] https://www.slideshare.net/AmeerKhan3/zigbee-based-patient-monitoring-system-21031306 [Last Accessed July 2017]

[44] https://store.arduino.cc/usa/arduino-mini-05 [Last Accessed July 2017]

[45] https://store.arduino.cc/usa/arduino-usb-2-serial-micro [Last Accessed July 2017]

[46] http://www.ebay.com/itm/100PCS-7CM-X-5CM-PCB-Soldering-Printed-Circuit-Board-Blank-/220845344036 [Last Accessed July 2017]

[47] http://18650-battery.net/wp-content/uploads/2016/03/74V-2200mAh-18650-Rechargeable-Li-ion-Battery-0-2.jpg [Last Accessed July 2017]

[48] http://www.jameco.com/z/MY-1--1-uF-100-Volt-Mylar-Capacitor_26956.html [Last Accessed July 2017]

[49] https://www.aliexpress.com/store/product/50-SETS-JST-XH-2-5-2-Pin-Battery-Connector-Plug-Female-Male-with-140MM-Wire/506373_32598005148.html [Last Accessed July 2017]

[50] N. Sani, W. Mansor, K. Lee, N. Zainudin, S. Mahrim, “Determination of heart rate from photoplethysmogram using Fast Fourier Transform”, International Conference BioSignal Analysis, Processing and Systems (ICBAPS), 2015

[51] Vyacheslav Tuzlukov, Signal Processing Noise, Electrical Engineering and Applied Signal Processing Series, CRC Press. 688 pages. ISBN 9781420041118, 2010

[52] https://en.wikipedia.org/wiki/Fourier_transform [Last Accessed July 2017]

[53] https://upload.wikimedia.org/wikipedia/commons/c/c1/ECG-RRinterval.svg [Last Accessed July 2017]

[54] http://www.nataliescasebook.com/tag/s-t-elevation [Last Accessed July 2017]

[55] K. Thenmozhi, P.Deepika, “Heart Disease Prediction Using Classification with Different Decision Tree Techniques”, International Journal of Engineering Research and General Science Volume 2, Issue 6, October-November, 2014

[56] https://physionet.org/physiobank/database/sddb/ [Last Accessed May 2017]