Hindawi
Advances in Human-Computer Interaction
Volume 2019, Article ID 1507465, 21 pages
https://doi.org/10.1155/2019/1507465

Research Article

An Energy Efficient Wearable Smart IoT System to Predict Cardiac Arrest

AKM Jahangir Alam Majumder,[1] **Yosuf Amr ElSaadany**,[2] **Roger Young Jr.**,[1] **and Donald R. Ucci**[2]

[1] *University of South Carolina Upstate, SC, USA*
[2] *Miami University, OH 45056, USA*

Correspondence should be addressed to AKM Jahangir Alam Majumder; majumdaa@miamioh.edu

Received 26 October 2018; Accepted 1 January 2019; Published 12 February 2019

Guest Editor: Maurizio Rebaudengo

Recently, many people have become more concerned about having a sudden cardiac arrest. With the increase in popularity of smart wearable devices, an opportunity to provide an Internet of Things (IoT) solution has become more available. Unfortunately, out of hospital survival rates are low for people suffering from sudden cardiac arrests. The objective of this research is to present a multisensory system using a smart IoT system that can collect Body Area Sensor (BAS) data to provide early warning of an impending cardiac arrest. The goal is to design and develop an integrated smart IoT system with a low power communication module to discreetly collect heart rates and body temperatures using a smartphone without it impeding on everyday life. This research introduces the use of signal processing and machine-learning techniques for sensor data analytics to identify predict and/or sudden cardiac arrests with a high accuracy.

1. Introduction

Heart problems have a significant impact on the quality of life of any who suffer from them. Through the widespread use of new technologies, there is a potential for advanced healthcare systems. The development of smart wearable IoT system for health monitoring is revolutionizing our lives [1]. Medical services have made large advancements in recent years. Computing and communication technologies have the potential to offer a wider variety of services for patients. Through this advancement, a patient's quality of life would improve and provide a benefit to a large portion of the population.

Through the availability and advancement of wearable IoT devices, it aids patients in monitoring and controlling their health metrics. An example of the benefits is that a patient can be made aware of the status of their condition with the aid of such devices at any time. That information can then be made available to the treating health care professional to provide prompt action for a condition or save the life of the user in an emergency. Connected health is proving to be a major application for developing technologies.

The concept of connected healthcare systems and smart embedded IoT devices offers a potential for both commercial companies and individuals to benefit. The goal is to use investigations performed on new technologies to enable the creation, enhancement, and expansion of connected health systems with the objective of developing a system that can help patients obtain a better awareness of their health status and provide early medical warnings.

The goal of the IoT is to enable things to be connected anytime and anyplace, with anything and anyone ideally using any path/network and any service [2, 3]. This goal requires more development in many areas including communication 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.

According to Cisco, there will be 50 billion devices connected to the Internet by 2020 [4]. 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.

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 [20]. 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 power for IoT devices. For example, an efficient service announcement and discovery protocol in IP-based ubiquitous sensor networks is proposed [8]. 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 allows seamless integration of low power devices into the Internet [21]. CoAP can run on most devices that support User Data Protocol and the network architecture that use this protocol is a hot research topic [22–26]. IoT devices use different protocols (Bluetooth, Zigbee, etc.) and different networks (LANs; WANs). Thus, an IoT platform has three building blocks: Cloud Computing is used as an enabling platform that supports IoT-based systems to allow connecting a 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.

1.1. Major Contributions.

In this paper, our aim is to develop a smart IoT system that is unique and stands out when it comes to eHealth based IoT systems for predicting a personalized cardiac arrest, because they naturally combine the detection and communication components. Our major contributions are as follows:

(i) *Developing a multisensory smart IoT-based cyber-human system for heart abnormality prediction.*

(ii) *Proposing a smartphone-based heart rate detection system using a wearable Body Area Sensor (BAS) system.*

(iii) *Designing, developing, and implementing a low power communication module to send data to the smartphone.*

(iv) *Implemening a mobile system for remote supervision of users, which can be used to detect personalized health crisis.*

The rest of the paper is organized as follows: in Section 2, we describe the background and relevant related work. In Section 3, we discuss the solution process of designing our system architecture and we explain the circuit design of our system. In Section 4, we discuss the data collection process and follow with Sections 5 and 6, which are data analysis, results, and evaluation of our smartphone-based prototype system. Finally, in Section 7, we conclude the paper with some future research directions.

2. Related Works

There are many research projects that attempt to characterize a user's heart abnormality; however, most of them have lack of key components. 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 using embedded systems and connected eHealth smartphone applications. Our proposed system is more related to connected eHealth smartphone applications since we are developing an application on smartphone that connects with a smart IoT device while most companies focus on comprehensive health care systems that allow users to interact with one another and benefit from resources.

2.1. Comprehensive Health Care Systems.

"PatientsLikeMe" 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 [27]. But, they did not mention data security and the usability of the system.

Another related system is called "DailyStrength". It is 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 [28]. It is very similar to

"PatientsLikeMe" in the sense that both of them are free platforms that involve patients and doctors interacting. Two major discrepancies between them are that "DailyStrength" does not involve research institutes and does not have a mobile application. Also, both systems are not IoT-based system.

In another work, a robust model was developed that included multiple pulse parameters, EEG, and skin conductance sensors into a shirt [42]. Another system was developed for connecting facial expressions and voice recognition with EEG patterns [43]. Other researchers proved that EEG alone exhibits characteristics for different emotions [44]. Facial recognition software has been compared with heart rate variability in order to better understand patterns associated with various emotions [45]. It has also been proven that certain pulse patterns are associated with physical stress and not from emotional stress [46]. But, their systems are mobile and they did not use IoT as a platform for their system.

Another comprehensive health care system is called "Omnio" which is an all-in-one application for Medical Resources [29]. It provides, among its services, clinical resources, diagnostic resources, disease guides, and drug information. Everyday Health [30] is a company which owns websites and produces content relating to health and wellness. It has higher ratings and publishes many health articles that can be very helpful for patients. In addition, it has a smart search that provides users with easy access their materials.

2.2. Connected eHealth Mobile Applications. Even though all the systems mentioned above provide health services, they do not provide any smart devices that can be used to monitor user's daily activities and alert them when needed. 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 (SEHMS) designed and developed on an Android™-based smartphone with an accelerometer; it could detect a fall of the user [31]. It provides a smartphone user interface to display health information gathered from the system. The main advantage of SEHMS is that it has the remote monitoring facility for elderly who and chronically hostile patients. But it cannot warn the user in case of emergency.

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 [32]. It allows care givers and loving one to monitor the user's to facilitate remote diagnosis. The system does not have the capability of monitoring in real time.

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 ECG to predict heart abnormalities. Other systems have different approaches with different hardware implementations. None of them were discussed about power consumption rate during data collection. Our system uses a low power Bluetooth module to collect a longitudinal data wirelessly using a smartphone.

In [33, 34], authors presented a comparison between different data mining techniques for heart attack prediction. They present just prediction algorithms rather than a complete system with a data collection device and a computing platform. 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 power smart IoT system and used a data mining technique in our prediction algorithm. Upon testing our prediction algorithm, we obtained results with a high accuracy for all our healthy and unhealthy test subjects. We illustrate the difference between our system and the other related works in Tables 1 and 2.

To address the drawbacks of the above-mentioned research and systems, in this paper, we propose a smart IoT-based heart rate monitoring system. Our system is designed to address directly some of the drawbacks of the existing systems and potentially yield good prediction results. The most important aspect of our system is the warning that allows the user to prevent an injury before it actually happens. To the best of our knowledge, our system is the first smart IoT-based health assistance which uses a subject-specific Body Area Sensor signals for predicting heart related injuries.

3. System Architecture

The strength of our system relies on existing wireless communications to provide low power with maximum freedom of movement to users in their physical activity. In addition, we have used small, light-weight smart IoT devices that are user friendly, like the smartphone and the wrist-band.

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 low power 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 (RR interval is the duration between two consecutive R peaks in an ECG signal), and ST segments (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). We then design an early warning system to monitor those parameters for signs of

Table 1: Qualitative comparison of existing work based on different features.

Approach	Use IoT Device	Mobility	Low Power System	Cyber Physical System	Cost Effective
PatientsL-ikeMe [27]	No	Yes	No	No	No
Daily Strength [28]	Yes	Yes	No	No	No
Om-nio [29]	Yes	Yes	Yes	Yes	No
Everyday Health [30]	Yes	No	No	No	No
SEHMS [31]	No	Yes	No	Yes	No
RMHM [32]	No	No	No	Yes	No
PHM [33]	Yes	Yes	No	Yes	No
Qardiocore [34]	No	No	No	Yes	No
Maksimović [35]	No	No	No	Yes	No
Stecker [36]	No	No	No	Yes	No
Mancini [37]	No	No	No	Yes	No
Sun [38]	No	No	No	Yes	Yes
Communicore [39]	No	No	N0	Yes	No
Kavitha1 [40]	Yes	No	No	Yes	No
Jagtap [41]	No	No	No	Yes	No
Our Approach	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

Table 2: Quantitative comparison of existing work based on different features.

Approach	Average Max HR	Approximate Accuracy	Average Max Sampling Rate	Number of Device (s) Used	Power Consumption in Watts
PatientsL-ikeMe [27]	160	90%	120	1	~ 500 mWatt
Daily Strength [28]	156	85%	110	1	N/A
Om-nio [29]	140	80%	100	1	N/A
Everyday Health [30]	144	85%	80	1	N/A
SEHMS [31]	155	78%	90	2	N/A
RMHM [32]	162	82%	140	2	N/A
PHM [33]	145	70%	150	1	N/A
Qardiocore [34]	135	78%	110	1	N/A
Maksimović [35]	155	85%	105	2	N/A
Stecker [36]	167	77%	130	1	N/A
Mancini [37]	151	87%	135	2	~ 600 mWatt
Sun [38]	160	75%	95	1	N/A
Communicore [39]	148	72%	150	1	N/A
Kavitha1 [40]	156	68%	155	1	N/A
Jagtap [41]	148	72%	145	2	N/A
Our Approach	**135**	**95%**	**160**	**1**	**~ 444 mWatt**

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 threshold level, 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. Figure 1 illustrates the prototype embedded smart IoT system.

The IoT device constantly collects data from the user and sends it to smartphone via a Bluetooth communication module. All the processing and data analysis take place in the application where the user has the option to view user real-time plots. These plots provide the user a basic idea of his/her body's status. The user does not have maintained 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.

3.1. Hardware. The initial prototype system consists of a low power Bluetooth chip, an Arduino Uno™, a pulse sensor, and a temperature sensor as shown in Figure 2. The other components are the power supply unit along with a smartphone with an application.

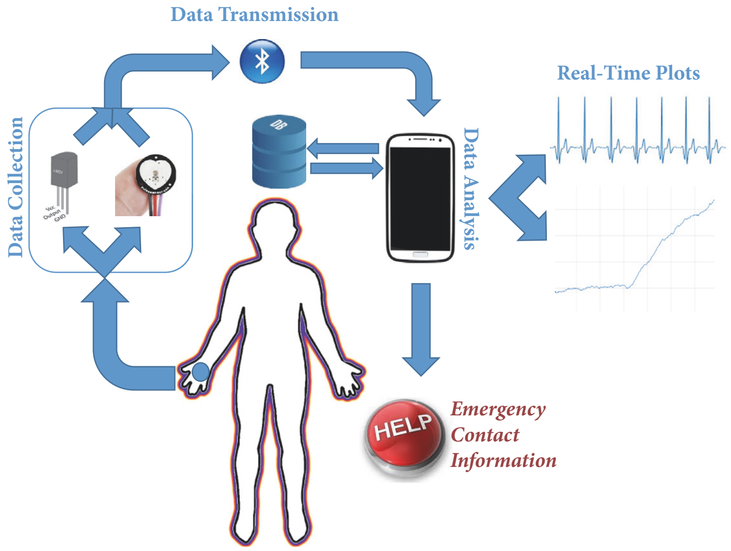
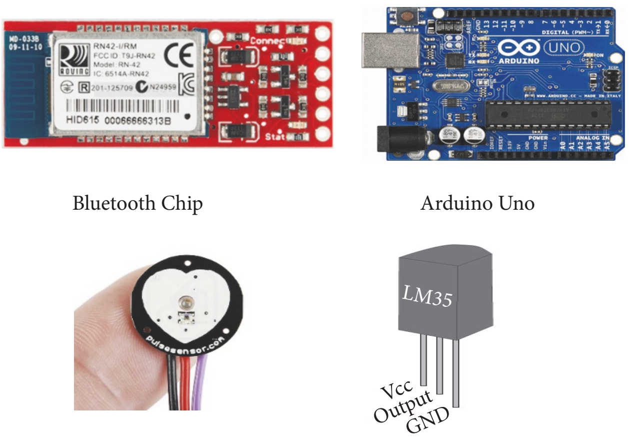


Figure 1: Proposed system architecture.



Bluetooth Chip

Arduino Uno

Pulse Sensor

Temp. Sensor

Figure 2: Hardware components for early prototype system [9–12].

The Arduino simply serves as an Analog to Digital Converter (ADC) [47]. 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.

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. Since the y-axis for ECG signals is also a voltage, all we had to do is scale the digital values to back voltage.

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.

To avoid the inaccuracy in simultaneous reading from multiple analog pins, we not only need a delay between each reading, but also need to read from the same analog pin twice. We read the temperature data from the sensor twice and send the second reading, then do the same for the pulse sensor. 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.

3.1.1. Hardware Modifications. After testing our early prototype system, 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).

(1) New Hardware Components. Rather than using the Arduino Uno, we decided to use the **Arduino Mini** instead. They both have the same microcontroller, clock speed, operating voltage, and range of input voltage. 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 important to have smaller components to be able to design a compact device.

To be able to upload code to the device using Mini USB Adapter, we also needed a **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 Figure 3.

(2) Design of the Wearable Device. After soldering all the hardware components on the PCB board, we design the system using Velcro strips to make it wearable. 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 Figure 4, where Figure 4(a) shows the device with the Mini USB Adapter attached and Figure 4(b) shows the device without the Mini USB Adapter.

Figure 4(b) 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. 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. It is easy to measure pulse from finger during daily activities of the user. 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. A complete smart wearable IoT device is shown in Figure 5.

(3) Smart IoT Device Performance. In this section, we explain 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 successful connection.

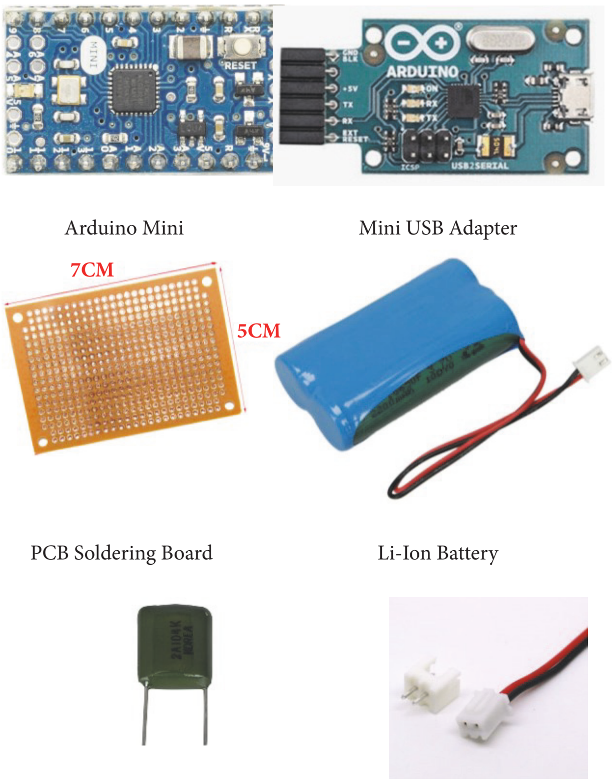
The performance of the device can be determined by measuring the current consumption which tells us how long the device can be powered. The voltage supplied from the battery is constant since the Arduino Mini takes the voltage it needs and supplies to the devices connected to it. The typical way to determine 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 test the device are the idle mode and the connected/transmitting mode. The measuring unit of the battery is in milliamp 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. Similarly, 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 3 shows the current draws, the device's lifetime, and the power consumption during the two modes for the IoT device.

The performance of the smart IoT device shows that the system can collect data for a long period of time 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.

3.2. 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 in the heart rate.

To transmit data to the smartphone through Bluetooth channel, 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. Then the smartphone Bluetooth adapter is opened through the application and it starts searching for the devices. After a successful connection, 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). The detail user interface of our system is shown in Figure 6.

After connecting to the IoT device, the application will automatically start receiving the sensors' data. The application parses the temperature and pulse data into separate arrays that are then sent to different pages where they are



Arduino Mini

Mini USB Adapter

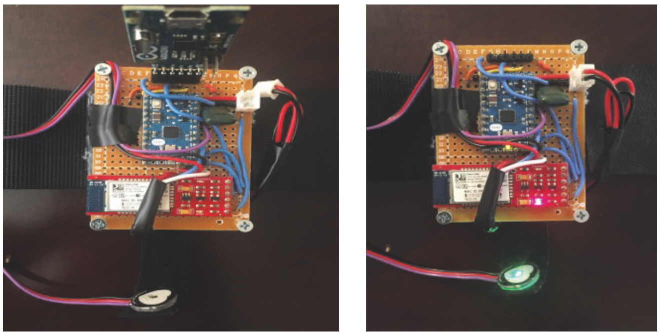
PCB Soldering Board

Li-Ion Battery

Capacitor

Pin Battery Connector
Plug

Figure 3: Hardware components for improved version of the system [13–18].



(a) With USB adapter

(b) Without USB adapter

Figure 4: Wearable IoT device.

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 signal waiting for any abnormality.

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. The user will then be directed to the home page 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.

In the connect page, at 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 user can then go to the home page where s/he will have several options between viewing his/her real-time plots

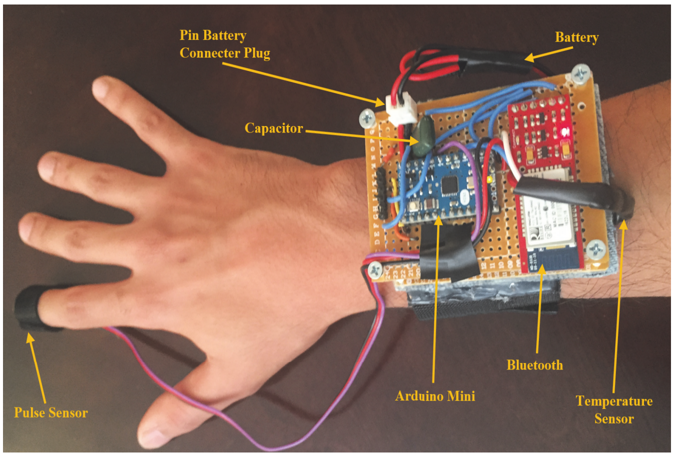


Figure 5: Wearable smart IoT device.

Table 3: Performance metrics of IoT device.

Mode of operation	Current Draw (mA)	Lifetime (Hours)	Power Consumption (mW)
Idle	26	84	192.2
Connected	60	36	444

Table 4: Statistics about subjects participating in our data collection.

Gender	Age [yrs.]	Height [cm]
F: 4	23-26: 8	150-159: 3
M: 16	27-34: 9	160-169: 5
	35-39: 3	170-179: 10
		180-189: 2

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. The time axis in real-time graphs 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.

4. Data Collection

After we finalized the system and were retrieving accurate results, we began testing on test subjects. Since, we cannot test our system with real people who have a chronic heart disease, we recruited a group of participants, a variety of age groups, and a range of heights (see Table 4 for statistics).

The data collection process can be divided into two parts, reading the data from the sensors and sending it to the smartphone. For the first part, one sensor gets the heart's pulse rate and the other one gets the body temperature. The sensors data is parsed and plotted on the device's screen.

4.1. Data Collection Interface. The sampling frequency or rate at which we are collected sensor data is the key challenge in data collection process. 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 by 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 7 shows the block diagram that describes the sensor data collection interface. The Bluetooth chip is also connected to the Arduino which enables the IoT device to transmit the sensed data to the smartphone application.

First, the user 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.

4.2. Test Subject Data Collection. Our proposed system is used to collect data from the users and store it in the smartphone's database and it can plot and process the data in real time. To be able to write our algorithm, we had to collect data from test subjects while doing different activities. 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.

4.3. Test 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 above. In this section, we show the sample ECG data for test subjects. The sample temperature sensor data are just plots to demonstrate the accuracy of the sensor.

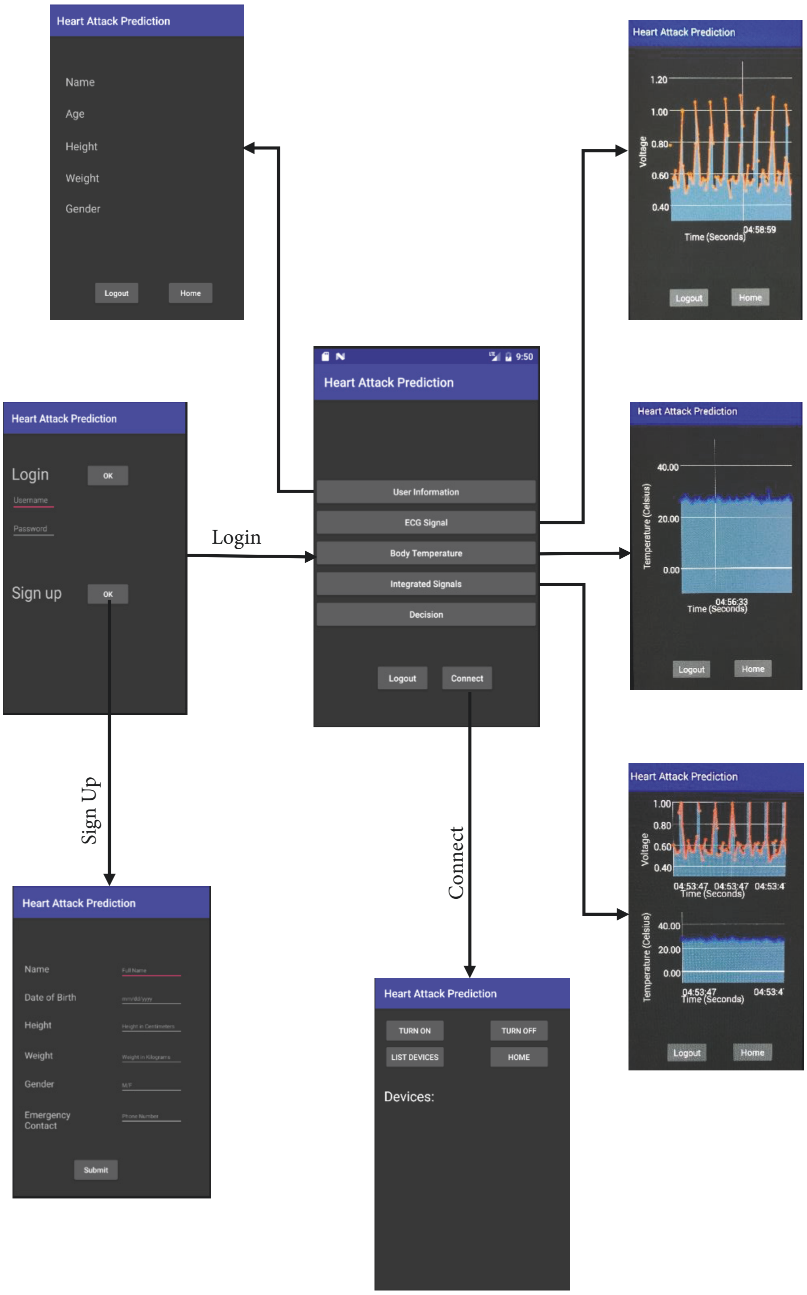


Figure 6: Smartphone user interface for data collection and for real-time graph.

4.3.1. 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. The "noisiness" in temperature signal indicates a need for smoothing. 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.

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 Figure 8. The one on the left shows the temperature decreasing while the one on the right shows the temperature increasing.

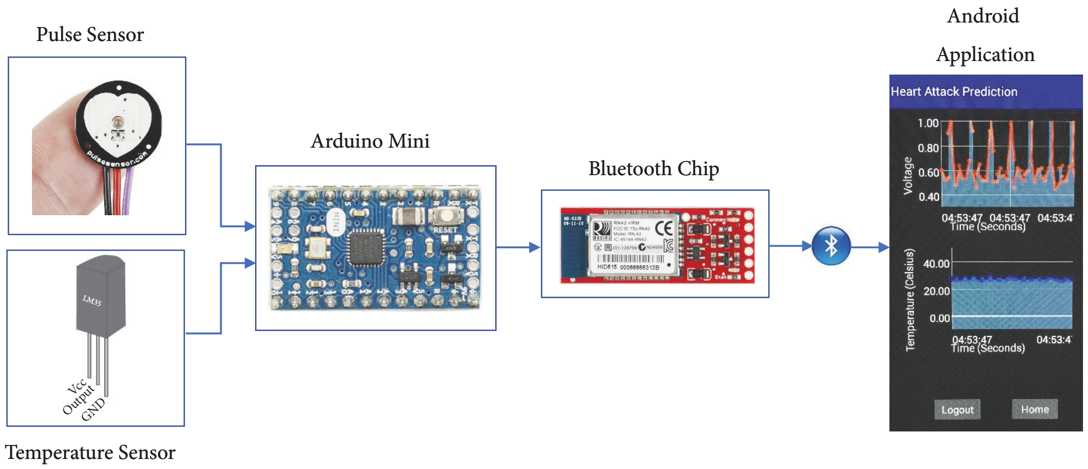


Figure 7: Data collection interface.

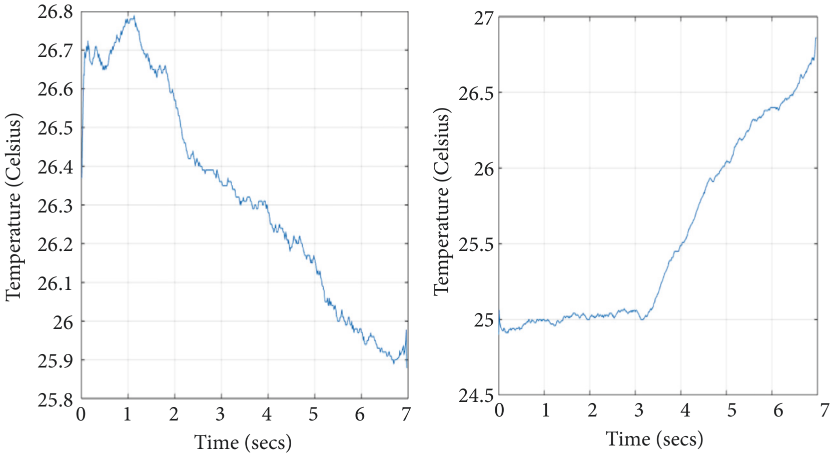


Figure 8: Temperature sensor data accuracy.

4.3.2. ECG Data. ECG data was collected from test 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 [48]. Table 5 shows the information of the four test subjects.

It is observed that the data collected for test subject 1 while sitting had no problems. Variations occurred when the data collected while walking and climbing upstairs. This is a result of the sensor moving while the subject was performing the different activities. We collected data for multiple times before we start analyzing. 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. A sample ECG signals for sitting, walking, and climbing upstairs for a test subject shown in Figure 9.

5. Data Analysis Techniques

Our data analysis was mostly done using MATLAB. In signal processing, noise is a general term for unwanted alterations that a signal may suffer during collecting, storing, transmitting, or processing data [49]. We collected data from analog sensors and transmitting them over a low power Bluetooth communication channel. We need data enhancement techniques before we can start analyzing the data as the reading can be affected by noise through the process. Since temperature values do not usually have many fluctuations, we are more concerned about the enhancement of the ECG signals.

5.1. Noise Reduction: Filtering. 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.

5.1.1. Baseline Wander Removal. The baseline wander is a problem that shows ECG signals in a wavy fashion rather than

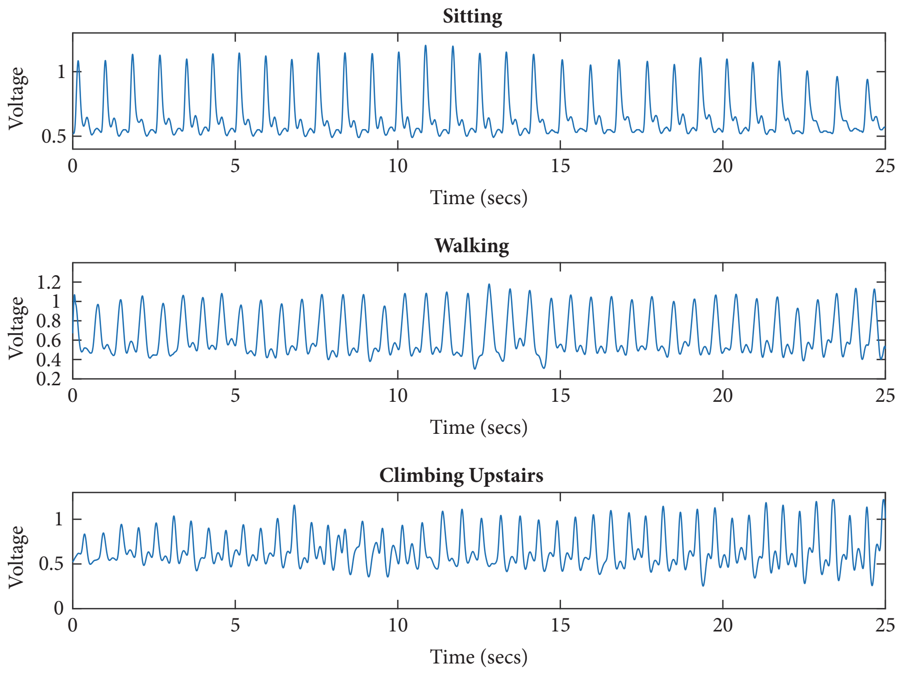


Figure 9: ECG signals for sitting, walking, and climbing upstairs for test subject 1.

Table 5: Test subject information.

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

being more of a constant envelope. A high pass filter to the signal improves the “look” of the signal because it removes the low frequency component that manifests itself as a sine-like pattern of the baseline. Removing the baseline wander gives a better signal which can help us process data more accurately. Equation (1) describes the process of reducing noise using base line wonder, where w_c is the cut-off frequency and N is the filter order:

$$|H(\omega)|^2 = \frac{1}{1 + (\omega_c/\omega)^{2N}} \qquad (1)$$

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 collected ECG signal.

5.1.2. Removal of High-Frequency Component. The time domain operation of a low pass filter for signals is the mathematical operation called the moving average (often addressed to as smoothing). 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.

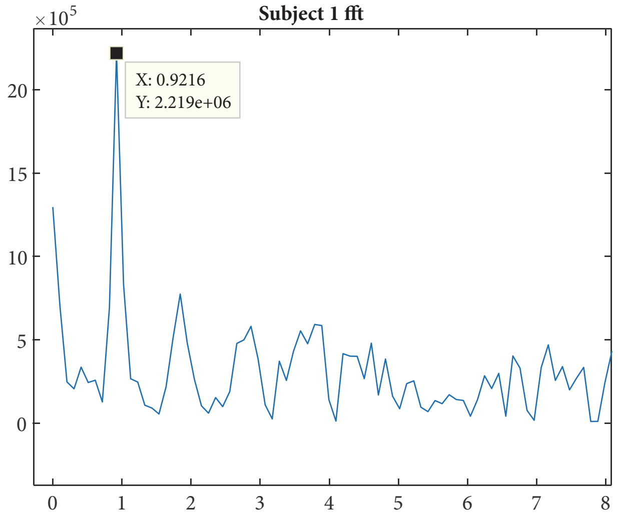


Figure 10: Fourier transform of an ECG signal.

Choosing the wrong cut-off frequency can result in huge alterations in the signal and irrelevant or, worse, erroneous data decisions. Equation (2) describes the operation of low pass filtering.

$$|H(\omega)|^2 = \frac{1}{1 + (\omega/\omega_c)^{2N}} \tag{2}$$

We 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 we used a smoothing window of 20 data points.

5.2. Extracting Features. After noise reduction, we extracted heart rate, RR intervals, and ST segments from ECG signals. We used these features as inputs of our prediction algorithm along with the body temperature. In the next subsections, we describe the process of extracting features from the ECG signal.

5.2.1. Heart Rate. We extracted heart rate or Beats per Minutes (BPM) from collected ECG signals. We can calculate BPM using several techniques including taking the number of QRS peaks in a given time, using autocorrelation, or using Fourier transform. The first technique sometimes yields inaccurate results; however, when a signal has no baseline wander problem, this technique should work. Autocorrelation and Fourier transform techniques yield very accurate results.

(1) Autocorrelation. In autocorrelation, a signal is correlated with a shifted copy of itself as a function of delay or lag. Correlation indicates the similarity between observations as a function of the time lag between them. We used this technique to analyze our data as the collected ECG signals are periodic. First, we calculate the difference between two peaks which gives 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 and multiplying it by 60 give us the total beats per minute.

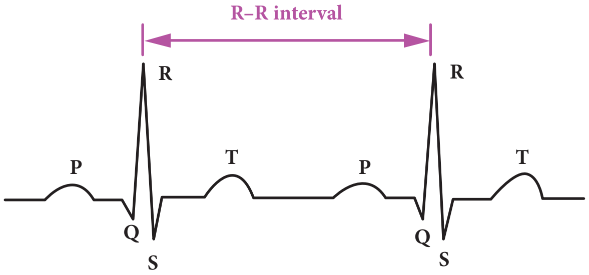


Figure 11: R-R interval of an ECG signal [19].

The mathematical equation for the autocorrelation function for signal processing is shown in

$$R(k) = \sum_{n=N1}^{N2-k} x(m) * x(m+k) \tag{3}$$

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.

(2) Fourier Transform. The Fourier transform extracts the frequencies and harmonics of the signal. So, we find the location of the maximum harmonic in the frequency plot.

The first significant harmonic in the signal is shown approximately around 0.92 (the red circle) as shown in Figure 10, which represents the beats per second. Simply multiplying this 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 equations for the Fourier and inverse Fourier transforms are shown below in (4) and (5), respectively [50].

$$F(\omega) = \int_{-\infty}^{\infty} f(t) * e^{-i\omega t} dt \tag{4}$$

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) * e^{i\omega t} d\omega \tag{5}$$

where $F(\omega)$ is the frequency domain of a given signal and $f(t)$ is the time domain of the signal. For our data analysis, we used an "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.

5.2.2. R-R Intervals. Another feature that we extracted from the ECG signal is called the R-R interval, which is the interval between successive R peaks in an ECG signal. For normal ECG signals, the R-R intervals do not fluctuate or suddenly change in a drastic manner. We recorded R-R intervals by having the standard deviation of the signal. Figure 11 gives a visual representation of an RR interval

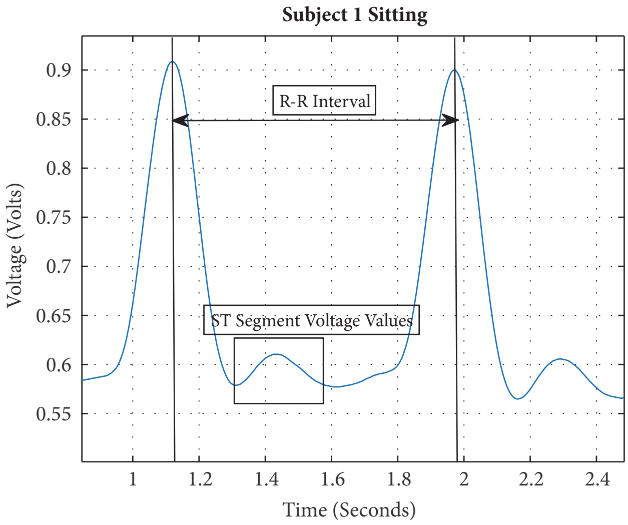


Figure 12: Sample ECG with R-R interval and ST segment.

Basically, we find the R peaks and subtract the peaks locations 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. We create arrays that store the R-R intervals of the ECG signal to calculate the variability of the durations.

5.2.3. ST Segments. Also, another feature is that we extracted ST segment voltage value from the ECG signals. We take the ST segment into consideration for heart attack predictions since elevated ST segments are one of the biggest indicators of heart attacks. Figure 12 shows the 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 R-R 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 Figure 12.

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 R-R 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.

5.3. Algorithm. The algorithm is the most important part of the system. The algorithm functions as shown in the flow chart in Figure 13. 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 Section 5.1. 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.

Our prediction algorithm is based on a predictive machine-learning model called J48 Decision Tree [51]. 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 designed a novel algorithm; Heart Attack Prediction using a **Decision Tree based on a Standard Deviation Statistical Analysis** (DTSDSA); that uses the decision tree model with a standard deviation statistical analysis. 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 abnormal. Figure 14 shows the structure of our decision tree which refers to the prediction algorithm block in Figure 13. Our algorithm uses warning levels from 0 to 4 to determine the degree of abnormality for each incoming window.

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 15 illustrates the appearance of both of the windows on one of our test subjects for both sensors. As shown in Figure 14, 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

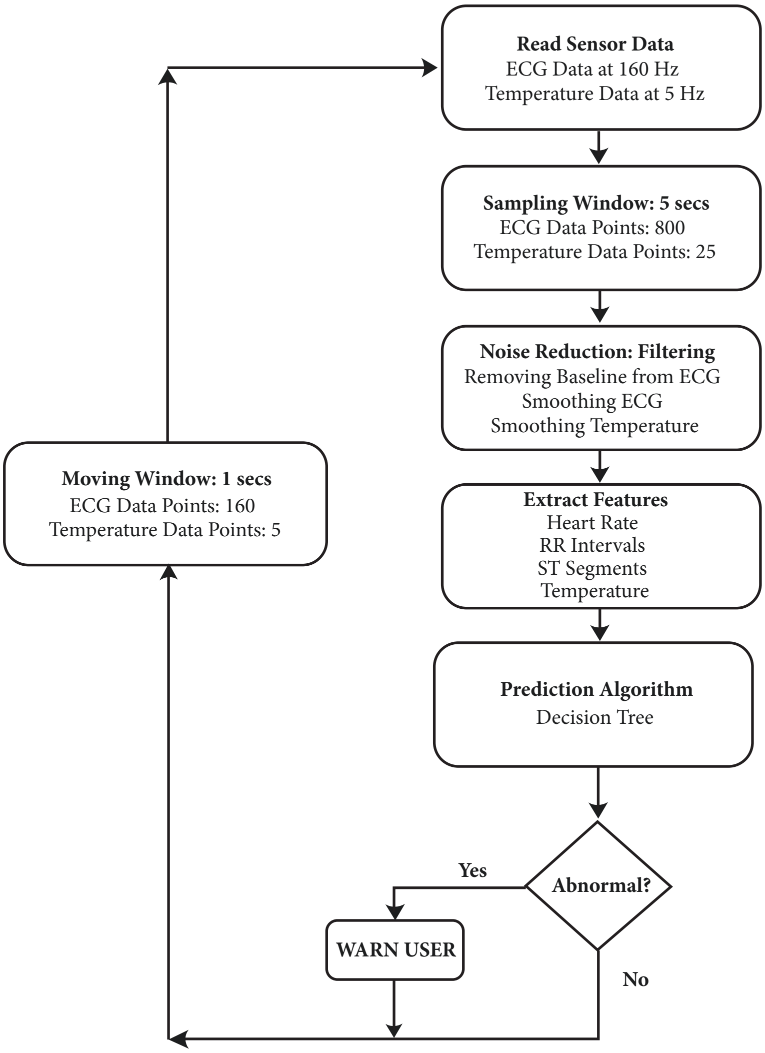


Figure 13: Flow diagram of our algorithm.

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.

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 R-R intervals if there is a problem with the current heart rate. For the R-R intervals and ST segments arrays, with which we are dealing, we calculate the standard deviation of the sample window for both features. If the R-R 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

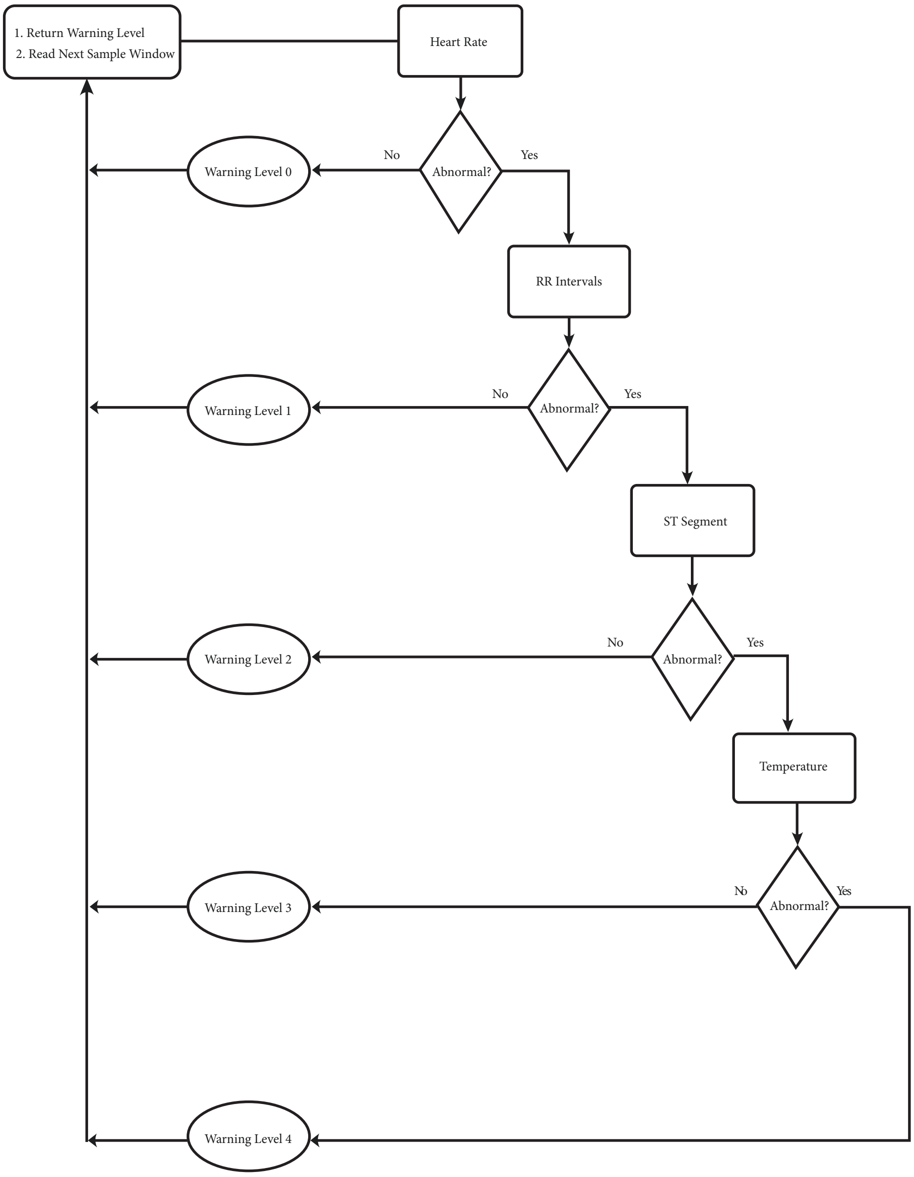


Figure 14: Flow diagram of decision tree algorithm.

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 16 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.

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 Figure 16 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. 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

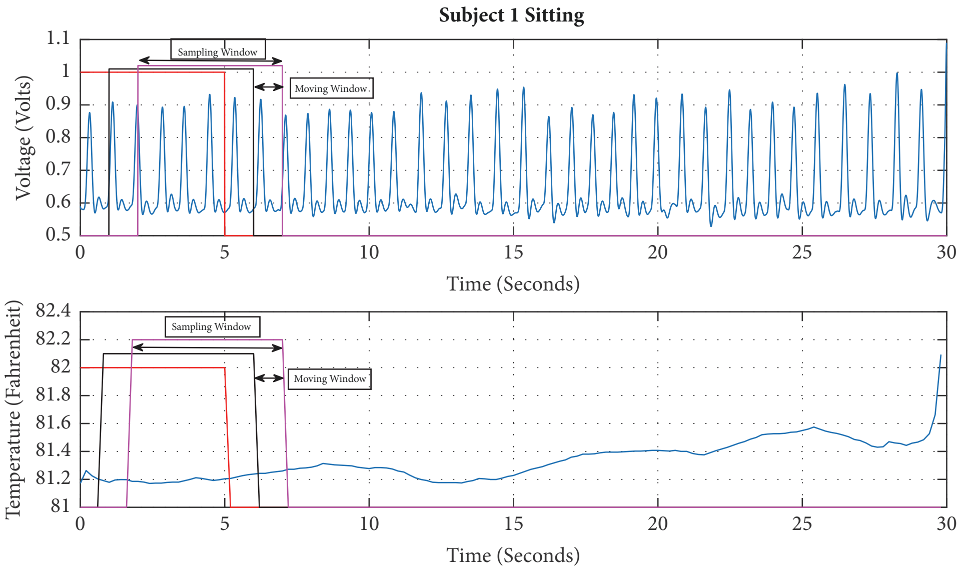
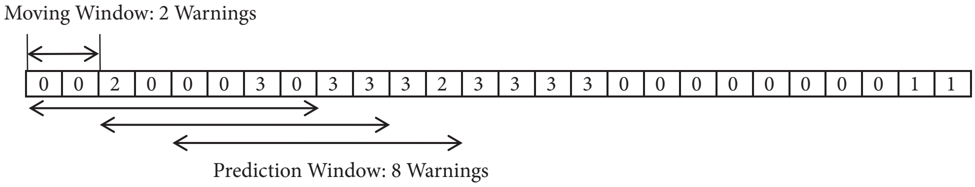


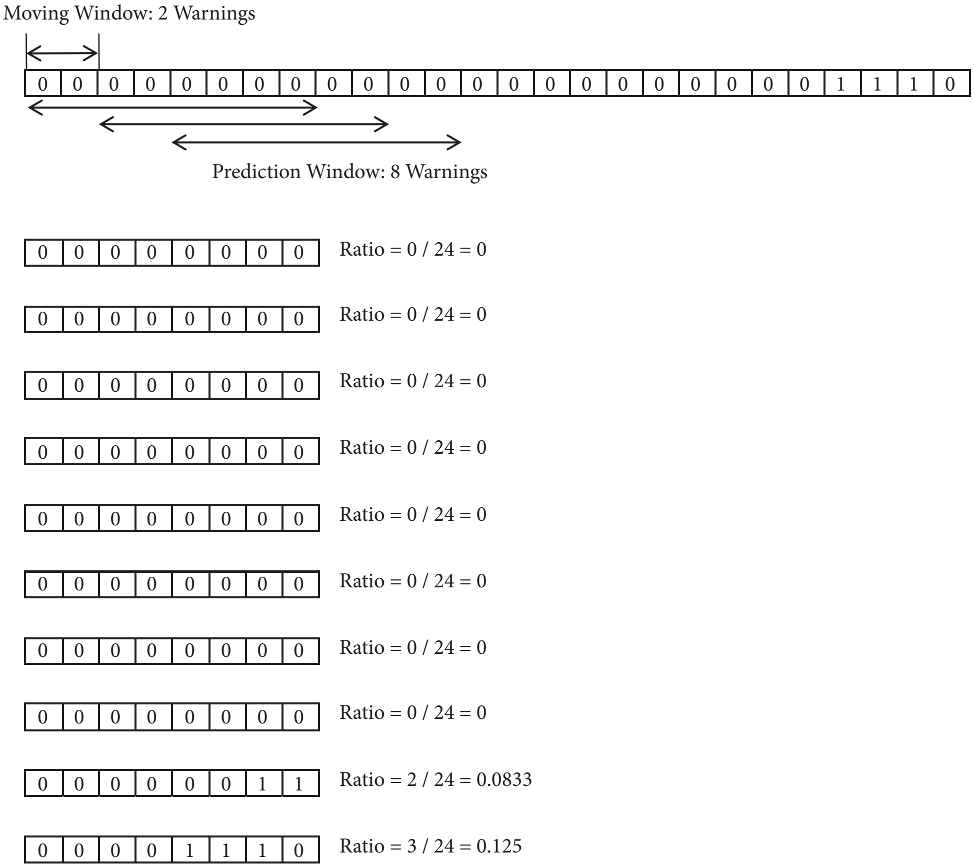
Figure 15: Illustration of sample and moving windows.



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 = 21 / 24 = 0.875	
2	0	0	0	3	0	3	3	Ratio = 11 / 24 = 0.4583	
0	0	3	0	3	3	3	3	Ratio = 15 / 24 = 0.583	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	

Figure 16: Algorithm results using prediction window.



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

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

Figure 17: Prediction algorithm results for test subject 1 while walking.

results are shown and discussed in more detail in the next section.

6. Results and Evaluation

To evaluate our proposed system, we developed a prototype application and investigated its performance. We evaluated the prototype with extensive experiments. In this section, we explain how the data is analyzed and performance is measured for healthy and unhealthy subjects.

6.1. 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 6.

The ECG signal while walking is considered as a normal and, therefore, no warning will trigger. The ECG signal while walking also consider as normal. But, we had a couple of false warnings while walking. We use the prediction window to eliminate the false warnings in our algorithm. Figure 17 shows that the results from the prediction algorithm had three warnings of level one while walking. Therefore, there was no need to warn the user since it was a false error.

The algorithm triggered a few warnings as well while the test subject was climbing upstairs. As shown in Figure 18, there are a few warnings for each prediction window, but, none of which above 50 percent, threshold level for indicating a myocardial infarction (MI).

6.2. Unhealthy Test Subjects. We were able to download datasets from a database online that has records of patients who suffered from sudden cardiac deaths. We also ran the algorithm on our 20 healthy test subjects and the results validated that the algorithm works with a high accuracy for the healthy test subjects. Table 7 shows the information of each test subject [52].

The results showed that the algorithm gives no warnings for all scenarios that had different heart rates. However, to validate our algorithm using only healthy subject data is not enough. Even though we ran our algorithm on noisy data, we still cannot conclude 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.

We tested our algorithm on the ECG signals from all the subjects shown in Table 7 and the results were accurate as expected. We show some details of the algorithm's results for the subject 5 from Table 7. Figure 19 shows the ECG signal for subject 5.

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

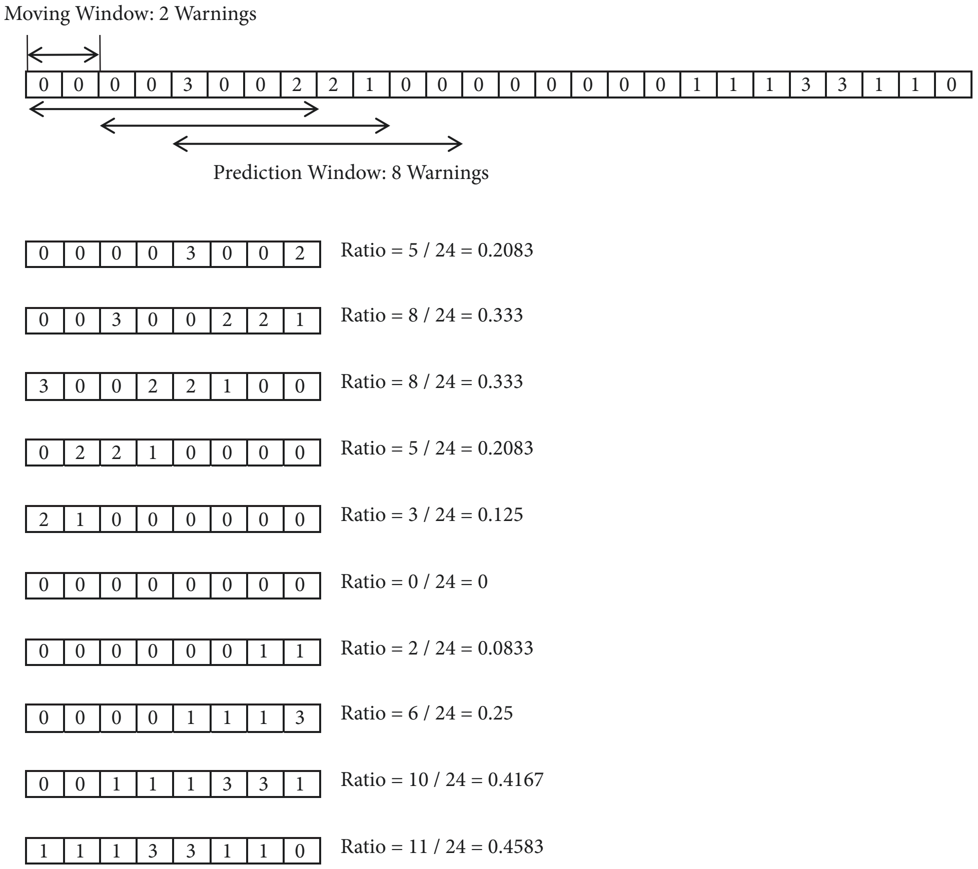

Moving Window: 2 Warnings
0 0 0 0 3 0 0 2 2 1 0 0 0 0 0 0 0 0 1 1 1 3 3 1 1 0
Prediction Window: 8 Warnings
0 0 0 0 3 0 0 2 Ratio = 5 / 24 = 0.2083
0 0 3 0 0 2 2 1 Ratio = 8 / 24 = 0.333
3 0 0 2 2 1 0 0 Ratio = 8 / 24 = 0.333
0 2 2 1 0 0 0 0 Ratio = 5 / 24 = 0.2083
2 1 0 0 0 0 0 0 Ratio = 3 / 24 = 0.125
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

Figure 18: Prediction algorithm results for subject 1 while climbing upstairs.

Table 6: Information for test subject 1.

Subject	Gender	Age	Scenario	Average Heart Rate
1	Male	24	(i) Sitting (ii) Walking (iii) Climbing Upstairs	(i) 84 (ii) 108 (iii) 135

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 Figure 20, the R-R Intervals had very high fluctuations which explain why the heart rate jumped from 71.94 to 131.89 in just one second. ⟶ Warning level 2

(3) As shown in Figure 21, the ST segment voltage values were also fluctuating in an abnormal fashion. ⟶ Warning level 3

The prediction results for the ECG signal are shown in Figure 22. The warning result from the second sample window, the one we discussed, is highlighted in yellow. We observed a remarkable fluctuation in all the features and the algorithm triggered warnings of level 3 for almost all the sample windows as expected for a patient who had a history of cardiac surgery and passed away shortly after the signal was recorded.

7. Conclusion

In this paper, we designed and developed an integrated smart IoT system to predict and monitor heart abnormality in

Table 7: 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

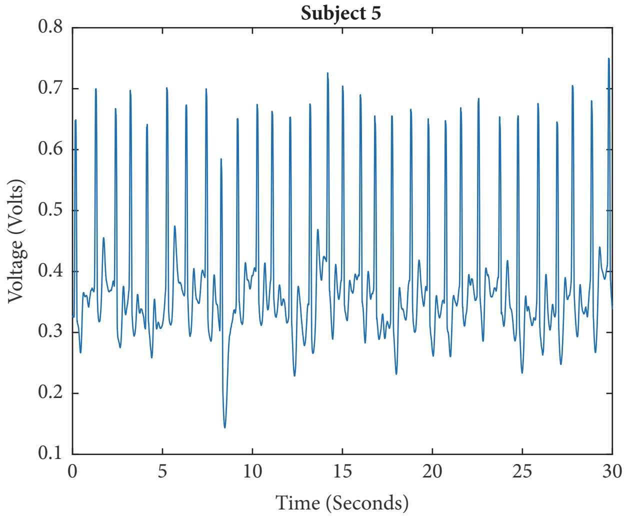


Figure 19: ECG signal of unhealthy subject 5.

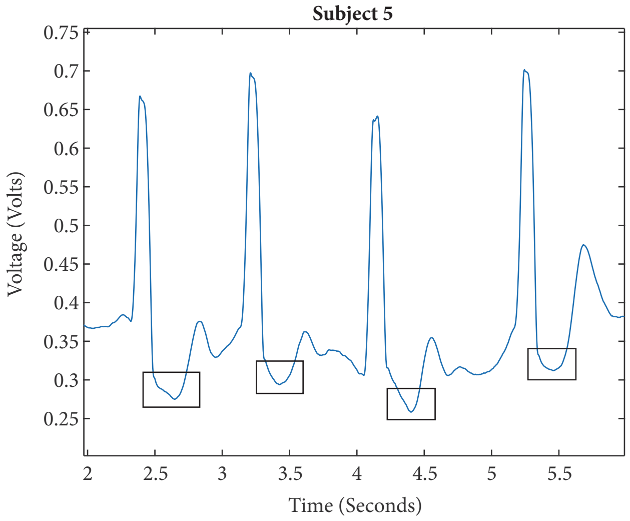


Figure 21: ST segments on sample window 2.

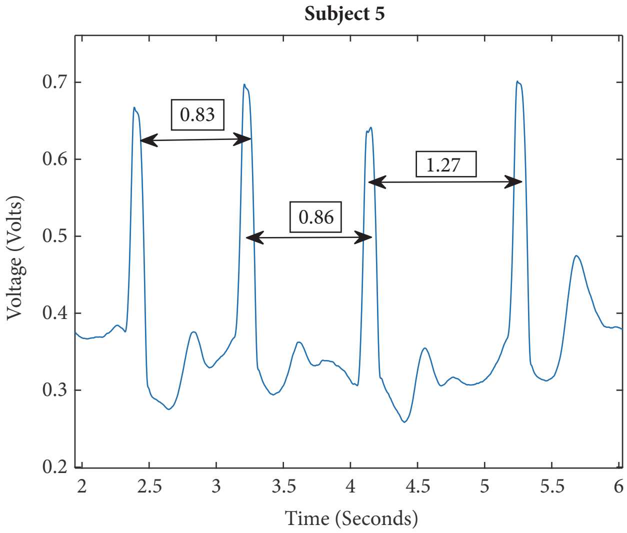
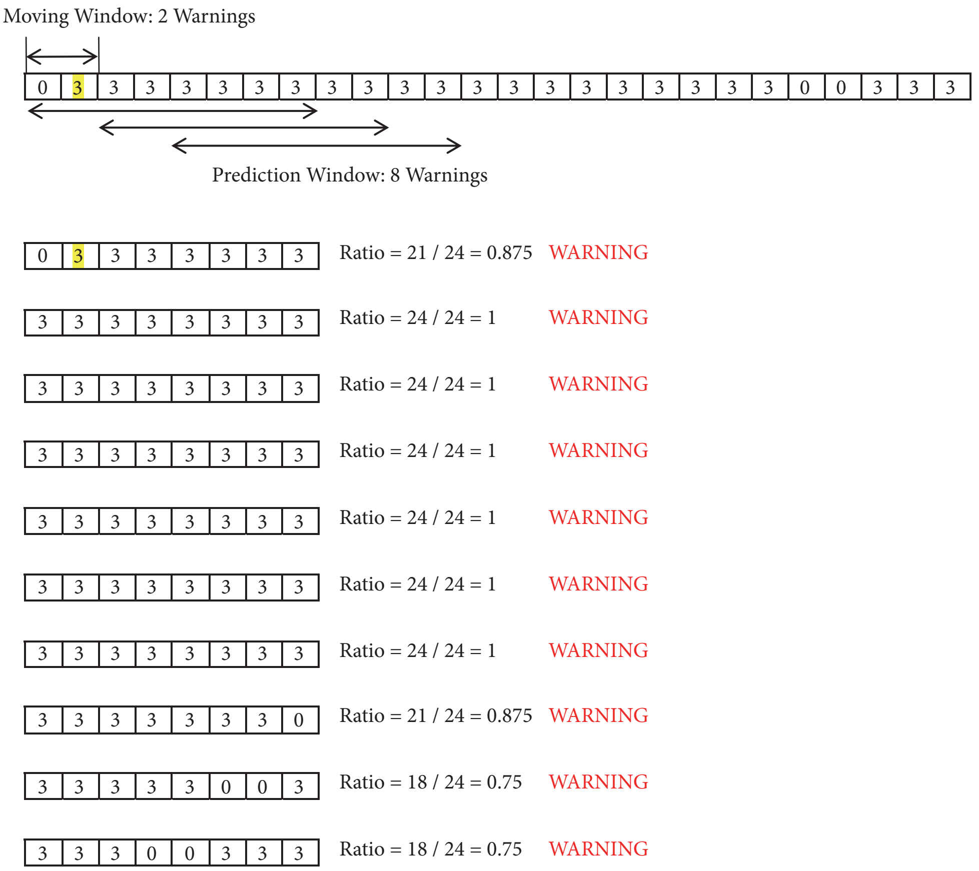


Figure 20: R-R Intervals on sample window 2.

user. We also managed to create a low power consumption communication channel between the smart IoT device and the smartphone application. This research provides users a noninvasive device that allows them to better understand how they may feel about their ECG. The results from different data sets are also presented to show that this approach provides a high rate of classification correctness in distinguishing between at normal and abnormal ECG patterns. The system may also find multiple applications in behavior detection for people with various disabilities.

To test the chronological durability and long-term feasibility of our approach in the future, we plan to test our system with data from the people who suffer from heart problems. We plan to test the power consumption rate for whole working life of the device during test on the field. We also plan to measure the different physiological parameters of the user during daily activities. Additionally, the system can be used in the smart home monitoring system for future wireless technology. Also, we can enhance the system by



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

Figure 22: Prediction algorithm results for unhealthy subject 5.

adding more sensors, like, Galvanic Skin Response (GSR), accelerometer, to the IoT device.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Disclosure

This paper is based on the MS thesis work by the author Yosuf Amr ElSaadany [53].

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the Department of Electrical and Computer Engineering, Miami University, Oxford, OH, USA. We would like to thank the Electrical Engineering Department at Miami University for funding the project. This especially includes Ms. Tina Carico and Jeff Peterson. We would also like to thank Ishmat Zerin for reviewing the early drafts of this paper.

References

[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.

[2] O. Vermesan and 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.

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

[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 and R. Witty, "An efficient service discovery protocol for 6LoWPANs," in *Proceedings of Science and Information Conference, SAI 2013*.

[9] https://www.sparkfun.com/products/12576.

[10] https://robotechshop.com/shop/arduino/arduinoboard/arduino-uno-r3-china/?v=7516fd43adaa.

[11] https://www.sparkfun.com/products/11574.

[12] https://www.slideshare.net/AmeerKhan3/zigbee-based-patient-monitoring-system-21031306.

[13] https://store.arduino.cc/usa/arduino-mini-05.

[14] https://store.arduino.cc/usa/arduino-usb-2-serial-micro.

[15] http://www.ebay.com/itm/100PCS-7CM-X-5CM-PCB-Solder-ing-Printed-Circuit-Board-Blank-/220845344036.

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

[17] http://www.jameco.com/z/MY-1-1-uF-100-Volt-Mylar-Capacitor_26956.html.

[18] 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.

[19] https://upload.wikimedia.org/wikipedia/commons/c/c1/ECG-RRinterval.svg.

[20] J. Penney, *Choosing An IoT Security Provider*, 2016, http://info.deviceauthority.com/blog-da/choosing-an-iot-security-provider.

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

[22] P. Sethi and S. R. Sarangi, "Internet of things: architectures, protocols, and applications," *Journal of Electrical and Computer Engineering*, vol. 2017, Article ID 9324035, pp. 1–25, 2017.

[23] P. P. Pereira, J. Eliasson, and J. Delsing, "An authentication and access control framework for CoAP-based Internet of Things," in *Proceedings of the 40th Annual Conference of the IEEE Industrial Electronics Society*, 2014.

[24] H. Khattak, M. Ruta, and P. di Bari, "CoAP-based Healthcare Sensor Networks: a survey," in *Proceedings of the 11th International Bhurban Conference on Applied Sciences and Technology*, 2014.

[25] M. Kovatsch, "CoAP for the Web of things: From tiny resource-constrained devices to the web browser," in *Proceedings of the 4th International Workshop on the Web of Things (WoT 2013), UbiComp '13 Adjunct*, Switzerland, Zurich, Sept. 2013.

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

[27] Patients Like Me, https://www.patientslikeme.com/.

[28] DailyStrength, http://www.dailystrength.org/.

[29] Omnio, http://omnio.com/.

[30] Everyday Health, http://www.everydayhealth.com/.

[31] J.-V. Lee, Y.-D. Chuah, and K. T. H. Chieng, "Smart elderly home monitoring system with an android phone," *International Journal of Smart Home*, vol. 7, no. 3, pp. 17–32, 2013.

[32] Y. Zhang, H. Liu, X. Su, P. Jiang, and D. Wei, "Remote mobile health monitoring system based on smart phone and browser/server structure," *Journal of Healthcare Engineering*, vol. 6, no. 2, Article ID 590401, pp. 717–738, 2015.

[33] P. Pawar, V. Jones, B.-J. F. van Beijnum, and H. Hermens, "A framework for the comparison of mobile patient monitoring systems," *Journal of Biomedical Informatics*, vol. 45, no. 3, pp. 544–556, 2012.

[34] Qardiocore, https://www.getqardio.com/qardiocore-wearable-ecg-ekg-monitor-iphone/.

[35] M. Maksimović, V. Vujović, and B. Perišić, "A custom Internet of Things healthcare system," in *Proceedings of the 10th Iberian Conference on Information Systems and Technologies (CISTI '15)*, Aveiro, Portugal, June 2015.

[36] E. C. Stecker, K. Reinier, C. Rusinaru, A. Uy-Evanado, J. Jui, and S. S. Chugh, "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*, vol. 6, no. 7, 2017.

[37] M. E. Mancini, M. Cazzell, S. Kardong-Edgren, and C. L. Cason, "Improving workplace safety training using a self-directed CPR-AED learning program.," *AAOHN journal : official journal of the American Association of Occupational Health Nurses*, vol. 57, no. 4, pp. 159–169, 2009.

[38] Feng-Tso S., Cynthia K., Heng-Tze C. et al., "Activity-aware mental stress detection using physiological sensors," *Mobile Computing, Applications, and Services*, pp. 211–230, 2012.

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

[40] G. Kavitha and E. Mariya, "Endowed heart attack prediction system using big data," *International Journal of Pharmacy & Technology IJPT*, vol. 9, no. 1, p. 285, 2017.

[41] S. Jagtap, "Prediction and analysis of heart disease," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 5, no. 2, 2017.

[42] C. Min, M. Yujun, S. Jeungeun et al., "Smart clothing: connecting human with clouds and big data for sustainable health monitoring," *Mobile Networks and Applications*, vol. 21, no. 5, pp. 825–845, July 2016.

[43] G. Chanel, J. J. M. Kierkels, M. Soleymani, and T. Pun, "Short-term emotion assessment in a recall paradigm," *International Journal of Human-Computer Studies*, vol. 67, no. 8, pp. 607–627, 2009.

[44] P. C. Petrantonakis and L. J. Hadjileontiadis, "Emotion recognition from EEG using higher order crossings," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 2, pp. 186–197, 2010.

[45] F. Bousefsaf, C. Maaoui, and A. Pruski, "Remote assessment of the heart rate variability to detect mental stress," in *Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare and Workshops, PervasiveHealth 2013*, pp. 348–351, Venice, Italy, May 2013.

[46] E. M. Tapia, S. S. Intille, W. Haskell et al., "Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor," in *Proceedings of the 11th IEEE International Symposium on Wearable Computers*, pp. 37–40, Boston, MA, USA, October 2007.

[47] https://learn.sparkfun.com/tutorials/analog-to-digital-conversion.

[48] N. Sani, W. Mansor, K. Lee, N. Zainudin, and S. Mahrim, "Determination of heart rate from photoplethysmogram using fast fourier transform," in *Proceedings of the International Conference BioSignal Analysis, Processing and Systems (ICBAPS)*, 2015.

[49] T. Vyacheslav, *Signal Processing Noise, Electrical Engineering and Applied Signal Processing Series*, ISBN 9781420041118, CRC Press, 2010.

[50] https://en.wikipedia.org/wiki/Fourier_transform.

[51] K. Thenmozhi and P. Deepika, "Heart Disease prediction using classification with different decision tree techniques," *International Journal of Engineering Research and General Science*, vol. 2, no. 6, 2014.

[52] https://physionet.org/physiobank/database/sddb/.

[53] Y. ElSaadany, *A Wireless early prediction system of cardiac arrest through IoT. [Dissertation, thesis]*, 2017, https://etd.ohiolink.edu/.