Abstract

An automatic health monitoring platform using deep machine learning and artificial intelligence (AI) based on Zigbee wireless sensors

Nahom G Ghebrmemeskel

St. Mary's University, 2018

Project Advisor: Vahid Emamian, Ph.D.

The goal of this research is to develop a human health monitoring platform using deep machine learning and artificial intelligence (AI). The signal collected from human body will be transmitted over a wireless channel to the platform using a Zigbee sensors. During this experimental research, we collect and process real time ECG signals to detect and alert regarding health issues. Several human body signals such as heartbeat, blood pressure, brain activities, and body temperature can be collected using various sensors, however, in this research our focus will be on Electrocardiogram (ECG) signals. ECG signal are collected and transmitted through Zigbee transceiver to the AI & deep learning platform where the signal is processed for diagnosis of heart issues. ZigBee transceivers can acquire and transmit/receive signals over a wireless channel. They offer efficient relay protocol, good transmission range, and flexible network structure with emphasis or power consumption efficiency.

Processing the ECG signals in real time and continuedly will help us detect heart issues at the early stage. We will use an AI and deep machine learning platform to train the machine detect and/or predict early stages of heart disease by real-time and continues processing of the ECG signals.

keywords: Electrocardiogram (ECG), Arrhythmia, Convolutional Neural Network (CNN), Data Augmentation, Xbee, AD 8232, Zigbee Communication, Heart Disease Detection.

Acknowledgments

I am grateful for the support of Dr. Vahid, for all the support, expertise, talent, and time he spent to help me make it to the day of my project defense. My conversations with him were often; however, I learned the most about research through these short talks. He was always there to help me when I needed help. Any time I emailed him day or night; I found his detailed response in a matter of minutes. Whenever he sent me his feedback, he made sure that every comment was completely explained. He was very acceptant, caring, and compassionate. It is rare to find an advisor so knowledgeable and at the same time so compassionate. Thank you, St. Mary's for having me a full-time graduate student. My sincere appreciation for the Department of Electrical & Computer Engineering for giving me opportunity to work as a Research Assistant with talented professors and hardworking graduate students.

I would like to extend my sincere gratitude to my committee members Dr. Mehran Aminian and Dr. Bahman Rezaie for being part of my advisory committee. I also want to thank everyone who supported me with a word or even a simple gesture during my study. Words cannot explain my great gratitude to your kindness and support.

Table of Contents

Chapter 1 Introduction 8

1.1 PROBLEM STATEMENT& BACKGROUND 8

1.2 BASICS OF ECG 9

Chapter 2 System Architecture 11

2.1 METHODOLOGY 11

Chapter 3 Data Acquisition 12

3.1 INTRODUCTION 12

3.2 COLLECTING ECG SIGNAL FROM THE BODY 12

3.2.1 Ad8232 Single Lead Heart Rate Monitor 12

3.3.2 Arduino Uno 14

Chapter 4 Wireless Sensing And Transmission Of Medical Data 16

4.1 INTRODUCTION 16

4.2 ZIGBEE NETWORK 17

4.3 IEEE 802.15.4 17

4.4 XBEE PRO S2B 18

4.4.1 Transparent Operating Mode 19

4.4.2 Api Operating Mode 20

4.5 TRANSPORTING THE RECEIVED ECG SIGNAL TO MATLAB 22

Chapter 5 Detection Of Arrythmia By Using Deep Neural Network 24

5.1 INTRODUCTION 24

5.2 METHODOLOGY 24

5.2.1 Ecg Data Pre-Processing 25

5.2.2 Feature Extraction 27

5.2.3 Aggregation Of Features Across 28

5.3 DATA SET 29

5.3.1 Data Augmentation 29

5.3.2 Training And Evaluation 30

5.3.3 Testing Of Data 31

5.4 RESULTS 31

Chapter 6 Conclusion 34

Reference 35

LIST OF FIGURES

Figure 1 ECG paper with time interval 10

Figure 2 Block Diagram of Proposed System 11

Figure 3 System Architecture 11

Figure 4 AD8232 Heart rate Sensor and the Electro Pads by ANALOG DEVICE 13

Figure 5 Converting Analogue to Digital in Arduino 14

Figure 6 Formula for ADC reading 15

Figure 7 Live ECG signal of human body in Arduino serial plotter using heartbeat sensor AD8232. 15

Figure 8 ZigBee Protocol stack 18

Figure 9 Xbee communication 19

Figure 10 Creating API Frame 21

Figure 11 Extracting API frame at the receiver 21

Figure 12 ECG signal in Arduino serial plotter after received using Xbee S2C sensor 22

Figure 13 Visualize the received signal using MATLAB 22

Figure 14 ECG Signal from the stored Data 23

Figure 15 Three classes of the data set such as Normal sinus rhythm (class 0), Paced Rhythm (class 2), Other rhythms (class 1) 25

Figure 16 MIT-BIH arrhythmia transformed to ECG Image 25

Figure 17 Plotting each ECG beat as an individual 200 x 200 scale image 26

Figure 18 Convolutional neural network of our proposed network 27

Figure 19 Architecture of proposed CNN Model 28

Figure 20 Spectrogram of a sample data instance belonging to each class 29

Figure 21 K – fold cross validation 31

Figure 22 Confusion matrix (a) with normalization and (b) without normalization of the CNN algorithm. 33

Figure 23 (a) train and validation loss graph (b) train and validation accuracy graph 33

LIST OF TABLES

Table 1 Pin connection of AD8232 with Arduino Uno Data Processing 13
Table 2 Comparison of Wireless Standards 16
Table 3 Architecture of proposed CNN Model 32

Chapter 1 Introduction

1.1 Problem Statement& Background

Electrocardiography has been in clinical use for the diagnosis and monitoring of heart abnormalities for more than a century. It remains the best and least invasive method for the task it performs. EKG measurement systems have followed trends in technological advancement becoming more reliable, able to perform a wider range of functions and simpler to use as time has progressed [1]. ECG monitoring is very important in the medical field. The monitoring of the ECG signals and an appropriate warning about the patient to the doctor at the appropriate time will save the patient. The integration of wireless communication into medical applications has immediate benefits. With wireless communications, monitoring of patients can be done remotely and efficiently. This would enable the intelligent monitoring of multiple patients concurrently. When a patient has alarming medical condition, the monitoring user can be informed to take appropriate action. This technology in hospitals is rapidly advancing. The potential is to have a totally monitored [2].

In most of the available literature, multiple features are extracted from the ECG signal fiducial points representing ECG attributes allowing to recognize the specific person using inter-subject variability [3]. However, the fiducial points detection is an error prone strategy that can be affected by changes of the signal slopes, inverted or abnormal waves, noise and artifacts.

Machine-learning (ML) offers an alternative approach to standard prediction modelling that may address current limitations. It has potential to transform medicine by better exploiting 'big data' for algorithm development [4]. ML developed from the study of pattern recognition and computational learning (so-called 'artificial intelligence'). This relies on a computer to learn all complex and non-linear interactions between variables by minimizing the error between predicted and observed outcomes [5]. In addition to potentially improving prediction, ML may identify latent variables, which are unlikely to be observed but might be inferred from other variables [6].

This study reveals, problems in current patient monitoring and established the most important medical parameters to monitor. an automatic and robust deep feature learning process using a convolutional neural network (CNN) is deployed to learn intrinsic features from the raw ECG data to perform human identification without having any complicated feature engineering process.

For a person who is suffering from cardiac diseases, this wireless monitoring of data becomes important to the expert for proper diagnosis. At the platform side, If the monitored data do not read as the normal value, then the system may be programmed to alert the expert. Therefore, this study can not only help us monitoring data, but also provide patient the attention he/she deserve to get healthy. On this experiment, an AD8232 biometric heart rate sensor will use to monitor the ECG signal through the three electrode cables (RL, RA, LA) and then send the signal to the Arduino. From the Arduino the signal will be transmitted over a wireless channel to the platform using a Zigbee sensors which are XBEE PRO S2B. This process involves simple yet effective architecture along with efficiency and is cost effective. The XBEE PRO S2B is a low-cost module, which we can give a microcontroller access to any Zigbee network and further upload data. The XBEE PRO S2B is an effective platform for communicating over long range of distance through wireless channel.

Efficient communication is essential for the well-being of the patient and for the effective treatment by the doctor and hence needs to be managed with accuracy and precision. In this paper, apart from the analog data acquisition between our sensor (ECG electrode) and the Arduino, communication is essentially Wireless by taking patient mobility into consideration. We tend to use the Zigbee network to our advantage for processing the data in remote platform and hence communication over the wireless is one of the most important parts of this proposed system.

1.2 Basics of ECG

Electrocardiography is a method of monitoring and recording the electric currents generated during the alternating contractions of the atria and ventricles of the heart [8]. An electrocardiogram is a device which is used to control and monitor these signals. The activities of cardiac muscle cells can be viewed as batteries that cause charge to move through the body fluid. By putting an electrode on the skin surface, we can detect the moving charges. electrodes are applied to the skin in places where the heart's signals can be measured easily. Usually, these locations are between muscles on the upper arms and lower legs. Figure 1 shows ECG paper with time interval.

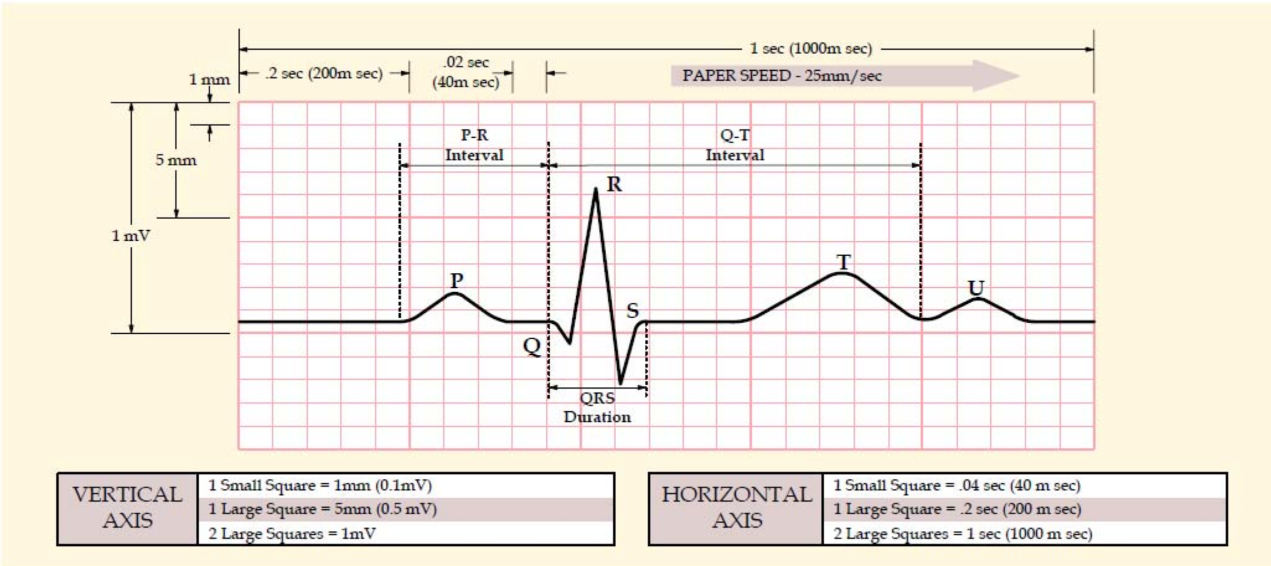


Figure 1 *ECG paper with time interval*

The right and left atria or upper chambers make the first wave called a “P wave" — following a flat line when the electrical impulse goes to the bottom chambers [9]. The right and left bottom chambers or ventricles make the next wave called a “QRS complex." And the final wave or “T wave” represents electrical recovery or return to a resting state for the ventricles [9].

Chapter 2 System Architecture

2.1 Methodology

We will take a bottom to top approach to address our communication procedure, starting from the raw data and ending at our website where the expert can access processed data [7]. The process starts at the sensors (in our case the ECG sensors), which collect data from the patient and send it via cable to the processor, serially. It is sent as a stream of data which is a reading of the voltage and hence is read easily by our microcontroller ATmega328 based Arduino. In the next step our Arduino communicates the data to the XBEE PRO S2B modules serially. The XBEE PRO S2B acts as an important communication interface in our project as it is the bridge between our raw data in the form of electrical signals, and the data over the processing plat form.

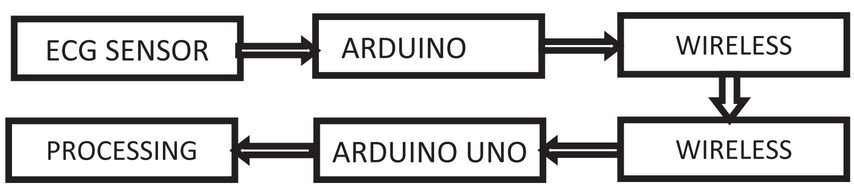


Figure 2 *Block Diagram of Proposed System*

Hence after this step, we will send our data on through wireless channel to the remote processing area and have following steps are a deep learning (CNN). The data is first received by XBEE S2B sensor at the receiving end and then transport to the computer through another Arduino for processing. Later it will present for the user (in our case, the expert) to analyze. This sums up our entire process of notifying the expert, who later with his doctor provides the patient with the best option to getting healthy.

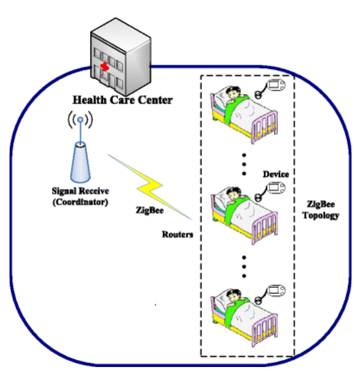


Figure 3 *System Architecture*

Chapter 3 Data Acquisition

3.1 Introduction

The first process for our data acquisition is collecting the data using our sensor network from the patient. The handling in this stage is important since the accuracy in data collection will finally determine the outcome and hence requires the highest level of precision. Then we will deal with the transmission of this data from the sensor network to our microcontroller and finally send it via the Zigbee module to the wireless channel. At the receiver end the data will receive by the other ZigBee and then transported to the computer through the Arduino at the receiver side. Those data transportation will result in some noises introduced in the circuit along with data loss and due to static charge. We can correct or eliminat those noises by proper sampling and quantization. The final stage basically deals with the processing of data at the remote platform and may not have much chances of error. Therefore, the way of ours will determine the efficiency of our outcome to the largest extent. Data Acquisition at the last stage basically deals with algorithms to graph and process our data. To conclude, data handling is probably the most important part of the project and cannot be neglected. We will have a look at the devices that have the biggest role to play in handling of data efficiently.

3.2 Collecting ECG Signal from the Body

3.2.1 AD8232 Single Lead Heart Rate Monitor

The AD8232 Spark Fun Single Lead Heart Rate Monitor is a cost-effective board used to measure the electrical activity of the heart [10]. Our AD8232 module is received the electrical activity as an ECG or Electrocardiogram in which its output as an analog reading. Those analogue signals (ECGs) are noisy and low voltage, therefore our AD8232 performs as an op amp to get a clear signal from the PR and QT Intervals easily. The AD8232 is an integrated signal conditioning block for ECG and other biopotential measurement applications and It is intended to extract, amplify, and filter small biopotential signals in the presence of noisy conditions, such as those created by motion or remote electrode placement [10].

N.B Negative feed Back of AD 8232 module

- It might last like 5 seconds to stabilize, but after that, is really good.
- The bandwidth is too big, I don't like this, because any movement makes noise because of the muscle signal.
- it goes all the way to 1k Hz, and a normal ECG doesn't go more than 250.
- HR monitor works nice, but SNR is really bad.

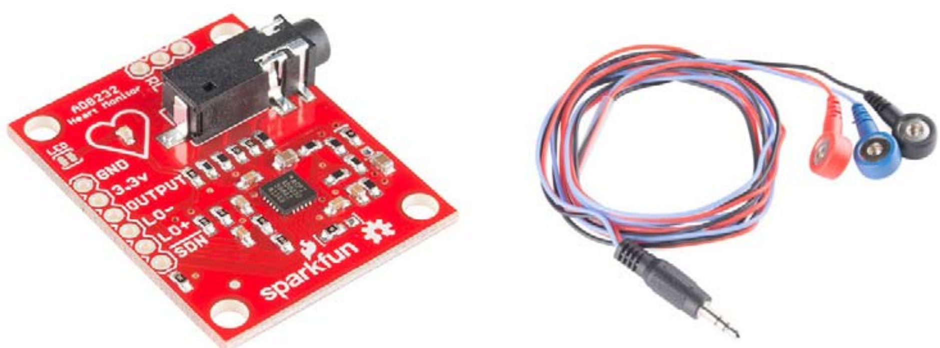


Figure 4 *AD8232 Heart rate Sensor and the Electro Pads by ANALOG DEVICE*

The electrical activities of heart are received by the ECG sensors, which are attached at the right position of the body (RL, RA, LA), over a period. The sensor outputs supplied to the AD8232 heart rate sensor for amplifying, and filtering. Then it will send the filtered and amplified signal serially to the Arduino for further process.

Board Label	Pin Function	Arduino Connection
GND	Ground	GND
3.3v	3.3v Power Supply	3.3v
OUTPUT	Output Signal	A0
LO-	Leads-off Detect -	11
LO+	Leads-off Detect +	10
SDN	Shutdown	Not used

Table 1 *Pin connection of AD8232 with Arduino Uno Data Processing*

3.3.2 Arduino Uno

Arduino Uno Rev. 3 Microcontroller Board is based on the Microchip Technology ATmega328 8-bit Microcontroller (MCU). Arduino Uno features 14 digital input/output pins (six of which can be used as PWM outputs), six analog inputs, and a 16MHz quartz crystal. Uno also includes a USB connection, a power jack, an In-Circuit Serial Programming (ICSP) header, and a reset button. This Arduino MCU board contains everything the user needs to support the MCU. The user can get started by connecting the Uno to a computer with the USB cable or by powering it with an AC/DC adapter or battery.

The Arduino ADC or Analogue to Digital Converter takes an input voltage and converts it into a digital value. With the standard setup you can measure a voltage between 0V and 5V with a resolution of 4.9mV so you can get a lot of detail when measuring analogue voltages. It maps input voltages between 0 and 5 volts into integer values between 0 and 1023.It takes about 100 micro second to read an analog input.The maximum possible sampling rate is 9615 HZ

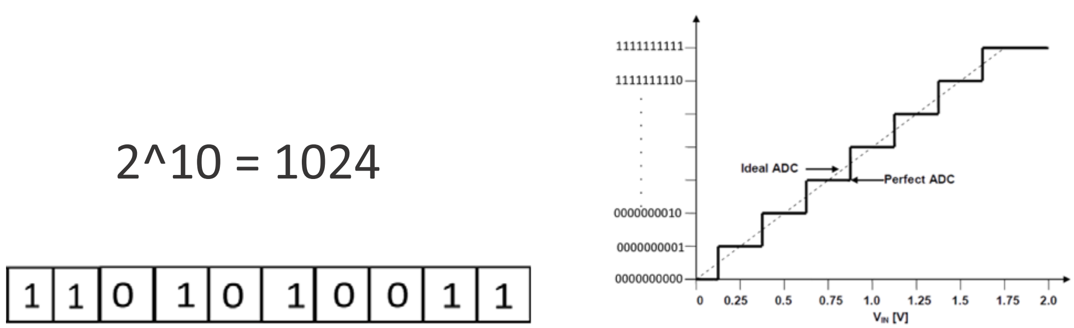


Figure 5 *Converting Analogue to Digital in Arduino*

We Use one Arduino at the transmitter side which is use to convert the Analog ECG signal from the AD8232 heart rate sensor to digital. The converted data from the sensor network is processed with the help of the microcontroller Arduino Uno. Arduino has also connections with XBEE PRO S2B which is through a series of AT commands sends the data to the wireless channel. We also use another Arduino at the receiver side(remotely) to receive the data from the wireless channel through another XBEE PRO S2B sent it to a PC for processing.

$$\frac{Resolution\ of\ the\ ADC}{System\ Voltage} = \frac{ADC\ Reading}{Analog\ Voltage\ Measured}$$

$$\frac{1023}{5.00V} = \frac{x}{2.12V}$$

$$\frac{1023}{5.00V} * 2.12V = x$$

$$x = 434$$

Figure 6 *Formula for ADC reading*

Below figure shows live ECG signal of human body in Arduino serial plotter using heartbeat sensor AD8232 and Arduino before it gets transmitted. It shows sequence of PQRST wave of ECG signal from human body when three ECG leads are connected to human body.

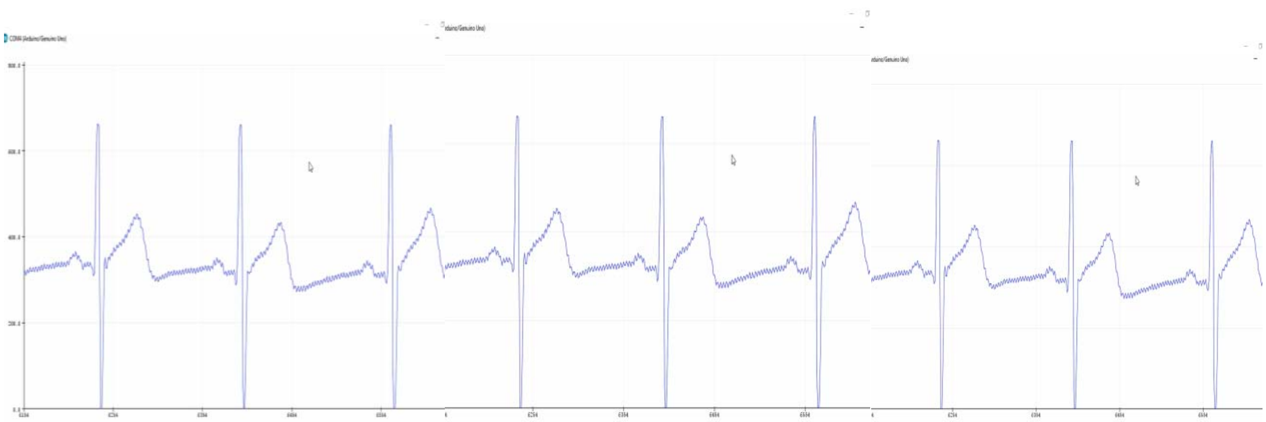


Figure 7 *Live ECG signal of human body in Arduino serial plotter using heartbeat sensor AD8232.*

Chapter 4 Wireless Sensing and Transmission of Medical Data

4.1 Introduction

Many medical applications would benefit from standards based wireless technology that is dependable, secure, and runs on low power. Well-known standards for wireless applications, such as Bluetooth and IEEE 802.11, allow high transmission rates, but at the expense of high-power consumption, application complexity, and cost. ZigBee networks on the other hand, are primarily intended for low duty-cycle sensors, those active for less than 1% of the time. For instance, an off-line node be able to connect to a network in about 30 ms. Waking up a sleeping node proceeds about 15 ms, as performs accessing a channel and transmitting data. Applications such as reading the pressure in an oxygen tank can propel the reading once per hour from a sensor which would then return to sleep. The low-power requirement extends battery life in remote sensors [10].

Difference of Wireless Standards			
Wireless parameter	**Bluetooth**	**Wi-Fi**	**ZigBee**
Frequency band	2.4 GHz	2.4 GHz	2.4 GHz
Physical/MAC layers	IEEE 802.15.1	IEEE 802.11b	IEEE 802.15.4
Range	9m	75 to 90m	Indoors: up to 30m Outdoor: up to 100m
Raw data rate	1 Mbps	11Mbps	250Kbps
ypical network join time	>3 sec	1 sec	30 msec
um quite bandwidth required	15 MHz (dynamic)	22 MHz (static)	3 MHz (static)
imum number of nodes per network	7	32 per access point	64 k
Number of channels	19	13	16

Table 2 *Comparison of Wireless Standards*

4.2 Zigbee Network

The network name comes from the zigzagging path a bee (a data packet) takes to get from flower to flower (or node to node) [11]. ZigBee is best known by its 6-layer OSI model for communication systems. Application (APL) Layer The top layer in the ZigBee protocol stack consists of the Application Framework which Provides a description of how to build a profile onto the ZigBee stack, ZigBee Device Object (ZDO) which Defines the role of a device within the network (coordinator, router or end device), initiates and/or responds to binding and discovery requests, and establishes a secure relationship between network devices., and Application Support (APS) Sublayer which Responsible for providing a data service to the application and ZigBee device profiles[12]. Security Service Provider (SSP) Offers security means for layers that use encryption (NWK and APS). Initialized and set upped through the ZDO. Network (NWK) Layer Controls network address and routing by making actions in the MAC layer. Figure-8 shows the layered protocol architecture. It should be noted that the ZigBee Alliance preferred to use an already existing mac layer and physical layers specification.

4.3 IEEE 802.15.4

802.15.4 is a packet-based radio protocol [13]. It refers to the communication requirements of wireless applications that have low data rates and low power consumption requirements. It is the basis on which ZigBee is built. Figure 8 shows a simplified ZigBee Protocol stack, which includes the two layers specified by 802.15.4: the physical (PHY) and MAC layers.

A. MAC Layer

It gives a reliable communication between a node and its closer neighbors, working To prevent collisions and enhance efficiency. It is also provided an assembled and decomposed data packets and frames.

B. Physical layer

The PHY layer describes the physical/electrical properties of the network. The PHY layer is also responsible for enable/disable the radio transceiver, link quality indication (LQI) for received packets energy detection (ED) within the current channel and clear channel assessment (CCA) [13]

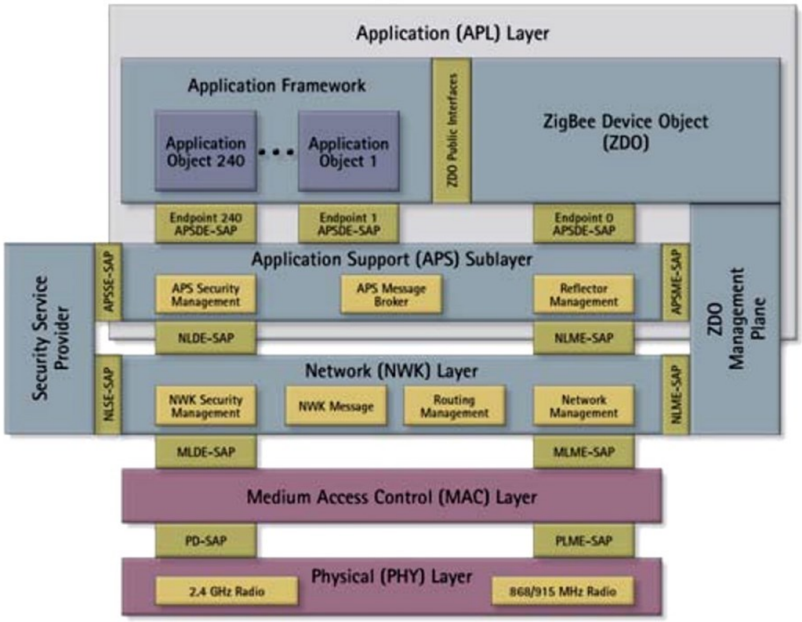


Figure 8 *ZigBee Protocol stack*

4.4 XBEE PRO S2B

In our study implementation, ZigBee network is set up in such a way that it uses one patient per unit being monitored. Every device is set up as a ZigBee End-device. Several devices may exist and account data simultaneously. For completely cover the monitored area, several additional ZigBee routers may be needed. Current version of the ZigBee standard does not give any solution for mobile nodes such as handover or roaming. When ZigBee device insert between coverage areas of different routers the transmission will be disturbed until the node locates new route to the controller. At the beginning of configuring our project the receiver Xbee finds the route to the controller (transmitter Xbee) then later it can reply to the route discovery request from the ECG device and does not require to rediscover the entire route. The application software proceeding on ZigBee devices is responsible for making of proper connection that carries corresponding commands, responses and data. Once the connection created, it is sent on to the ZigBee APS layer for the transmission over the air using API specified by ZigBee stack manufacturer.

Application endpoint has one incoming cluster and two outgoing. Incoming cluster is used for command and control messages, one of the outgoing clusters is used to send response to control messages and the other to send raw ECG data. Node can accept command messages such as" start"," stop" to control transmission, "set FQ" to make sample frequency and others may be defined in the future. At the receiver side, ZigBee device consists of 2 incoming and 1 outgoing clusters as shown in figure 9. Outgoing cluster is for command and control interface. Incoming clusters receive command responses and ECG data. Upon receiving the data, ZigBee coordinator passes it to the platform for further processing and analysis.

4.4.1 Transparent Operating Mode

By default, Xbee RF modules operate in transparent mode. When operating in this model, all UART data received through the DI pin (which is connected to arduino at the transmitting side) is queued up for RF transmission. When data is received, the data is sent out the D0 pin (which is connected to arduino at the reciever side). Data is buffered in the DI buffer until one of the following causes the data to be packetized and transmitted.

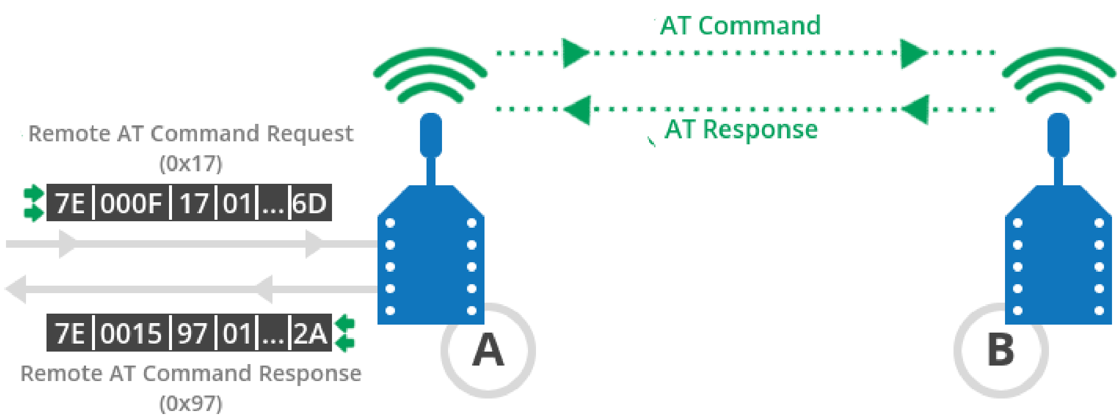


Figure 9 *Xbee communication*

- No serial characters are received for the amount of time determined by the RO (packetization Timeout) parameter.
- The maximum number of characters that will fit in an RF packet (100 byte) is received.

If the module cannot immediately transmit, the serial data is stored in the DI buffer. The data is packetized and sent at any RO timeout or when 100 byte (maximum packet size) are received. If the DI buffer becomes full, hardware or software flow control must be implemented inorder to prevent overflow (loss of data between the host and module).

4.4.2 API Operating Mode

API Operation is an alternative to the default transparent operation. The frame-based API extends the level to which a host application can interact with the networking capabilities of the module.

When in API mode, all data entering and leaving the module is contained in frames that define operations.

Transmit Data Frames (received through the DI pin) include:

- RF Transmit Data Frame.
- Command Frame.

Receive Data Frames (sent out the DO pin) include:

- RF-received data frame.
- Command response.

The API provides alternative means of configuring modules and routing data at the host application layer. A host application can send data frames to the module that contain address and payload information instead of using command mode to modify addresses.

The API operation option facilitates many operations such as the examples cited below:

- Transmitting data to multiple destinations without entering Command Mode.
- Receive success/failure status of each transmitted RF packet.
- Identify the source address of each received packet.

We use an API mode to configure the xbees communication. This Mode help us to synchronize the xbee with Arduino so that the xbee receive two bytes of data from Arduino and make an API frame then transmit at the same time. This avoid any delay due to packaging time out happening in AT mode. Hence, in API mode all data entering the module can immediatley transmit. This overcome delay and lagging due packaging of data until a buffer is full.

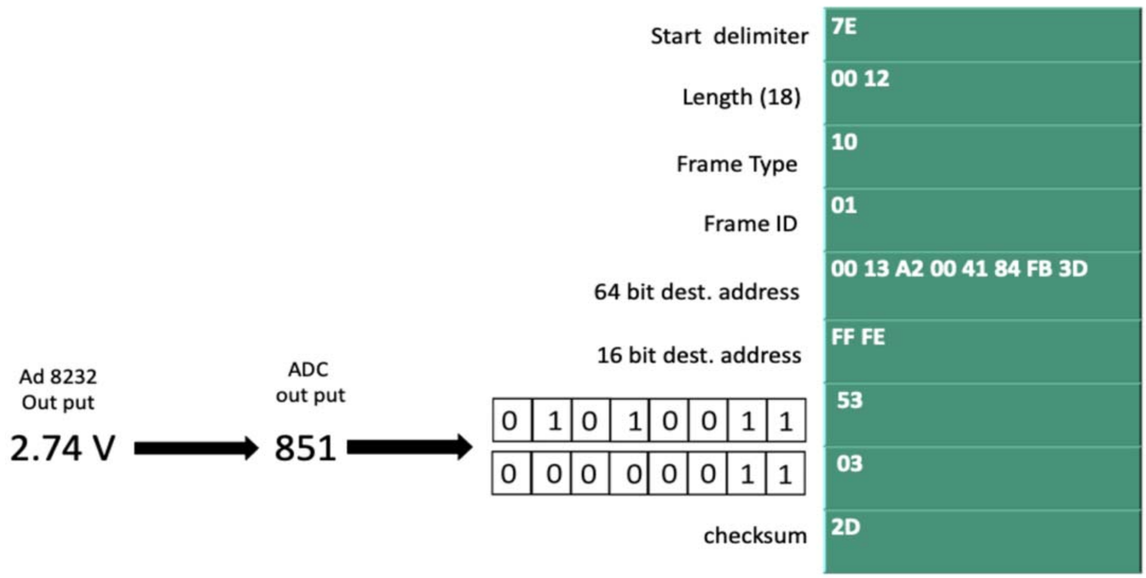


Figure 10 *Creating API Frame*

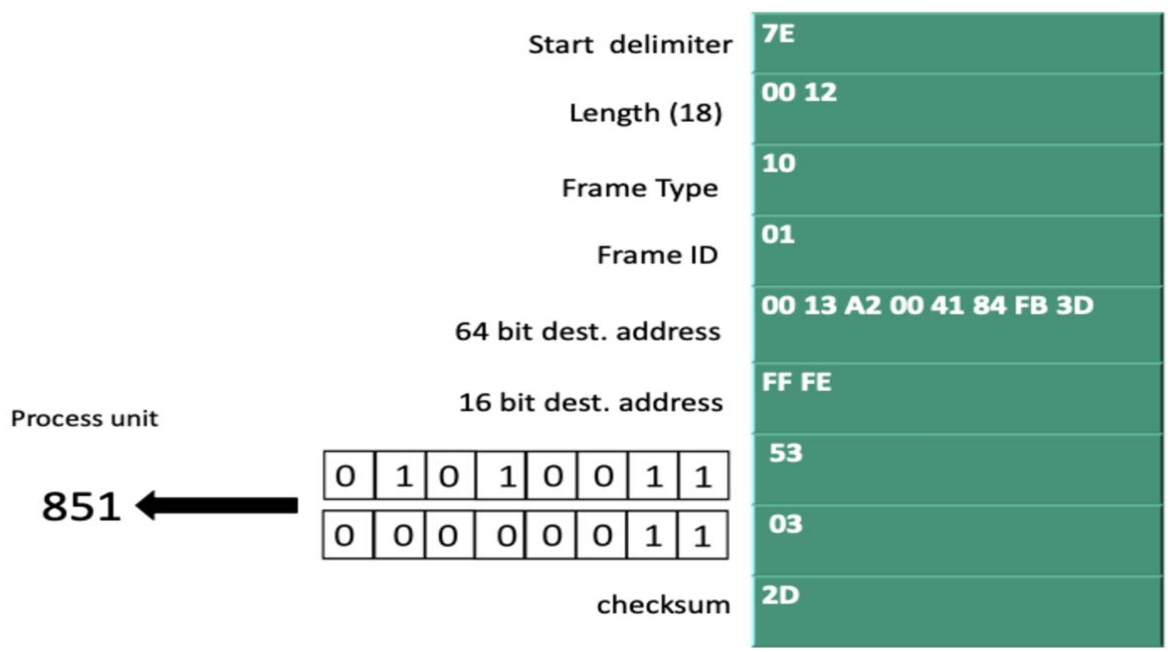


Figure 11 *Extracting API frame at the receiver*

Below figure shows live ECG signal of human body in Arduino serial plotter after received using Xbee S2C sensor.

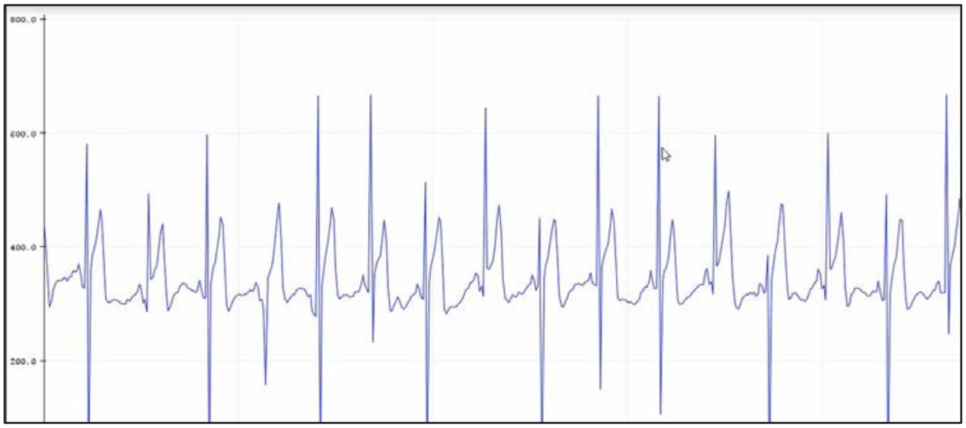


Figure 12 *ECG signal in Arduino serial plotter after received using Xbee S2C sensor*

4.5 Transporting the Received ECG Signal To MATLAB

The ECG Data received by Xbee S2C and send to the Arduino from Arduino to MATLAB. We use MATLAB Support Package for Arduino Hardware to communicate serially with the Arduino from MATLAB. MATLAB code (Script) is written to visualize the coming signal in real time and storing them in EXCEL as data sheet for Back up. Below figures shows live ECG signal of in MATLAB the stored ECG data in EXCEL after received using Xbee S2C sensor.

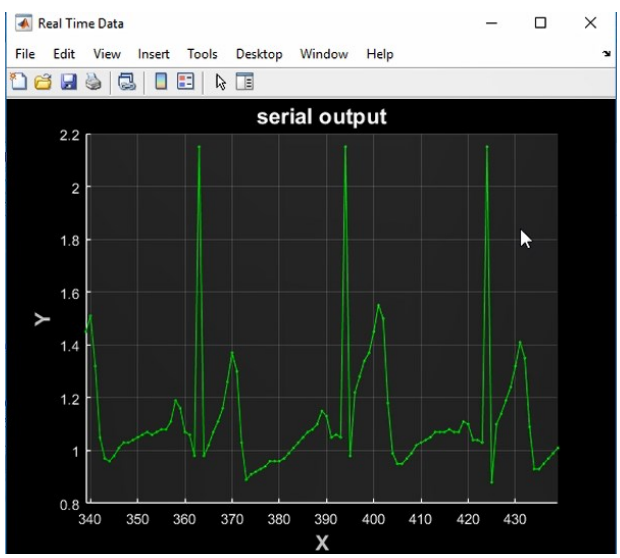


Figure 13 *Visualize the received signal using MATLAB*

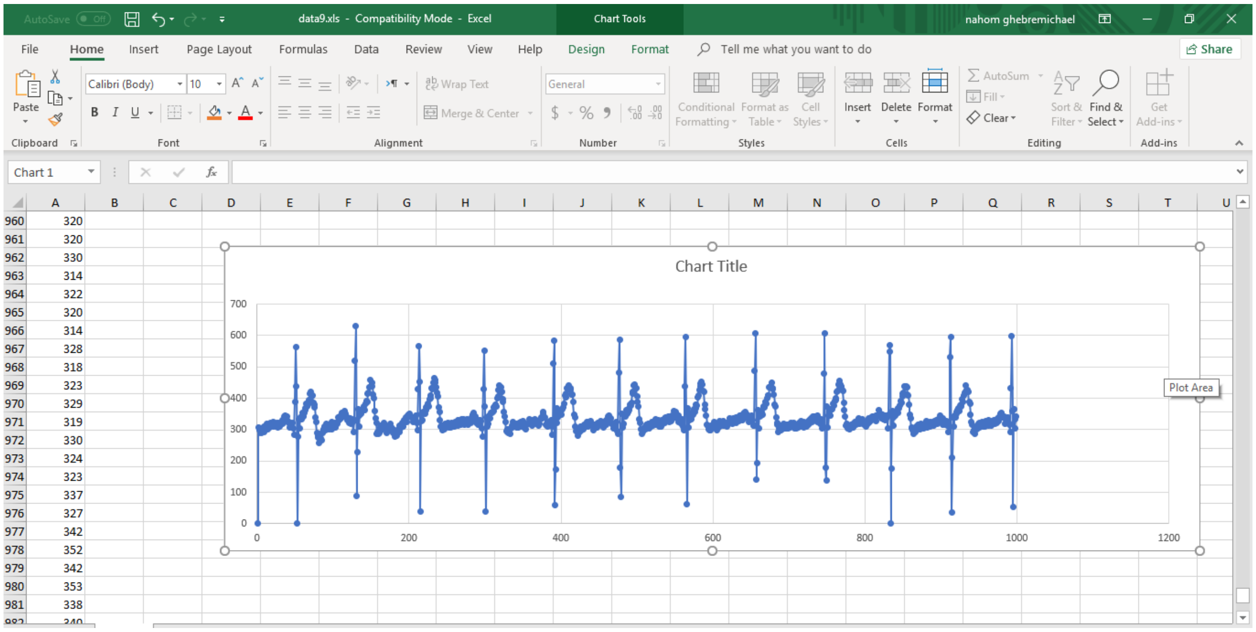


Figure 14 *ECG Signal from the stored Data*

We also use MATLAB to store all the received data as a .csv format for every 10sec at sampling frequency of 43 bit per second. We converted the ECG data from the .csv formatted document into an Image using python in because a two-dimensional CNN requires an image as input data.

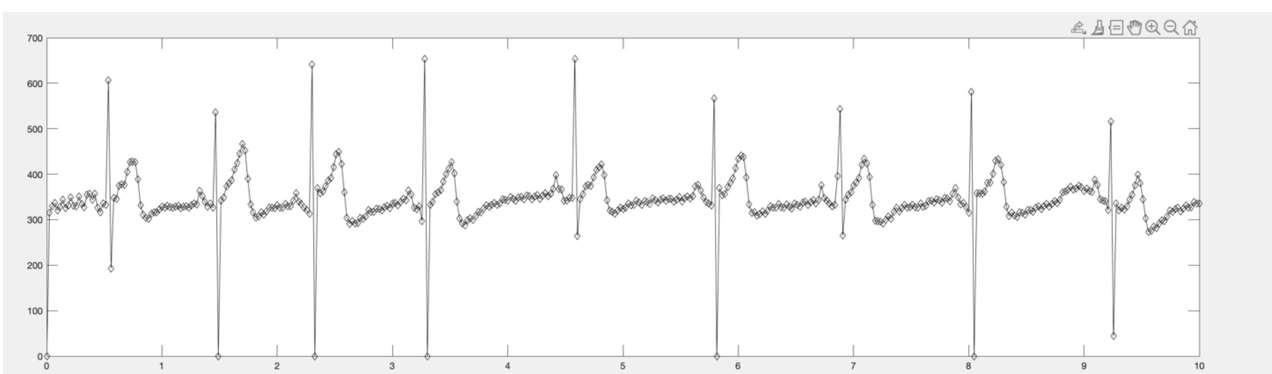


Figure 15 *ECG Reading for 10 sec at 43sample/sec in matlab*

Chapter 5 Detection of Arrythmia By Using Deep Neural Network

5.1 Introduction

CNN was used for automated detection of coronary artery disease and it remains robust despite shifting and scaling invariance, which makes it advantageous [14]. In our research, we propose deep neural network use to process and detect heart issue. In order to do that we use CNN architecture for classifying electrocardiogram (ECG) recordings from a single-channel handheld ECG device into three distinct categories: normal sinus rhythm (N), paced rhythms (A), or other rhythm (O). For the classification of arbitrary-length ECG recordings, we evaluate them using the AF (atrial fibrillation) classification data set provided by the PhysioNet/CinC Challenge 2017. AF happens in 1-2% of the population due to an increase in age and is associated with significant mortality rate and disease. Unfortunately, current AF classification methods are unsuccessful at solving the potential of automated AF classification to have poor generalization capabilities experienced by training and/or evaluation on small and/or carefully selected data sets. Our architecture uses an averaging-based feature aggregation with 24-layer convolutional neural network (CNN). CNNs can extract features invariant to local spectral and spatial/temporal variations, and have led to many breakthrough results, most prominently in computer vision [15].

5.2 Methodology

In order to classify the input ECG signal into three classes of interest, the recordings are first cut, and the data is nominated based on the labels. After nominating, each data is transformed into an image of grayscale 200 x 200. After that, the ECG images are taken into a CNN for training and testing, a 24-layered deep CNN. The output of those layers is used to extract features. At the end, averaging-based feature aggregation across time is used for classifying the features. Our research consists of the following steps: data processing, future extraction using block of convolutional layers, and aggregation of features across time by averaging.

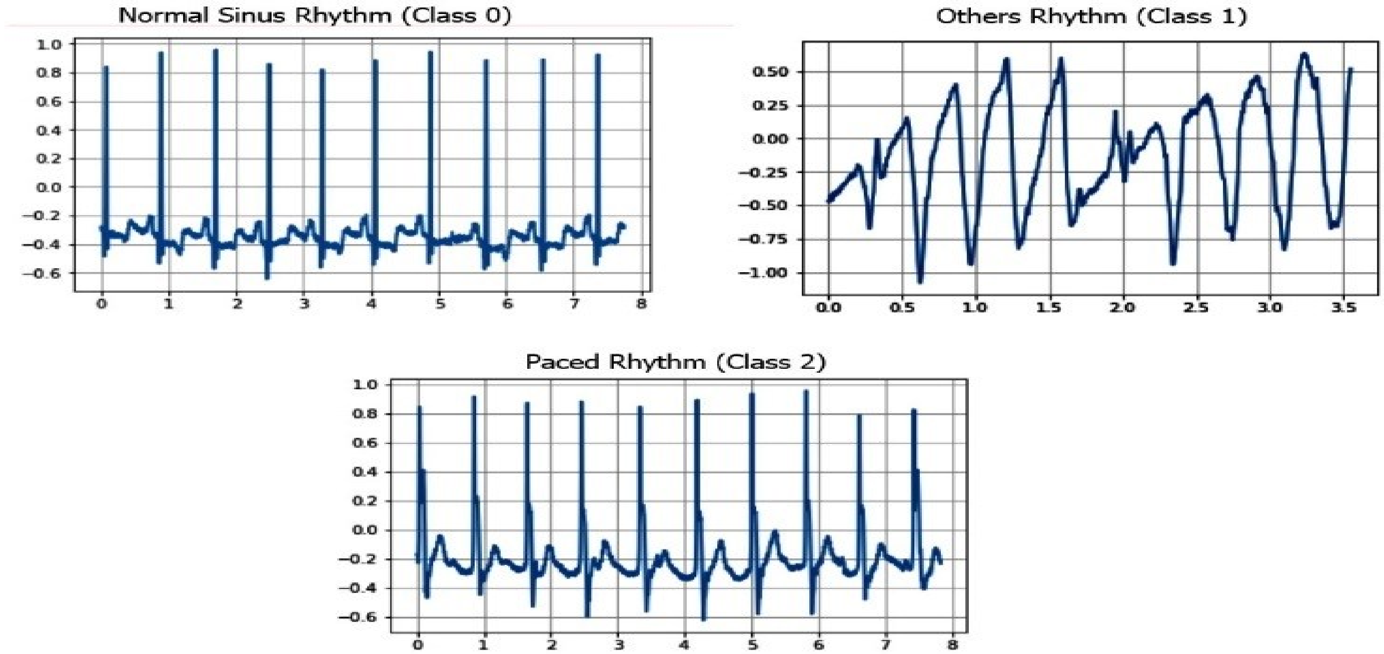


Figure 16 *Three classes of the data set such as Normal sinus rhythm (class 0), Paced Rhythm (class 2), Other rhythms (class 1)*

5.2.1 ECG Data Pre-Processing

In this paper, we used the MIT-BIH arrhythmia database [16] for the CNN model training and testing. The MIT-BIH Arrhythmia Database contains 48 half-hour excerpts of two-channel ambulatory ECG recordings, obtained from 47 subjects studied by the BIH Arrhythmia Laboratory between 1975 and 1979 [17]. The recordings were digitized at 360 samples per second per channel with 11-bit resolution over a 10-mV range [17]. Since the CNN model uses 2D images as input data, we convert the ECG signal into ECG images in the ECG data pre-processing step. The next step is the CNN classifier step in which we use the ECG image to get classification of three ECG types. Overall procedures are shown in Figure 17.

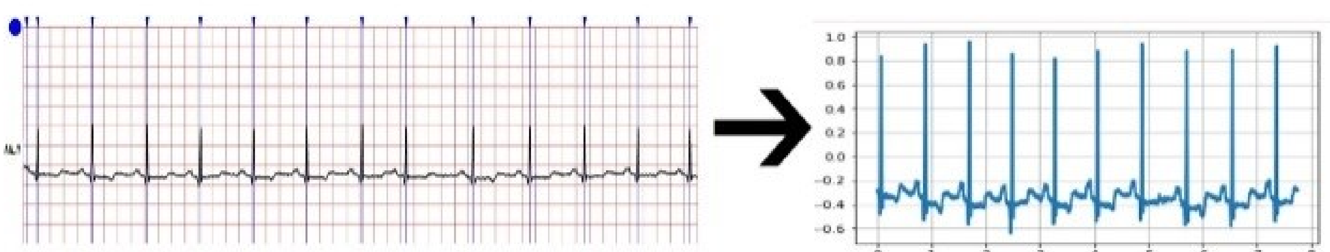


Figure 17 *MIT-BIH arrhythmia transformed to ECG Image*

5.2.1.1 ECG Image

We converted ECG signals (stored data from matlab) into ECG images because a two-dimensional CNN requires an image as input data. We then plotted each ECG beat as an individual 200 x 200 grayscale image. In the MIT-BIH arrhythmia database, every ECG beat is divided based on Q-wave peak time. More specifically, the type of arrhythmia is considered at the Q-wave peak time of each ECG beat. Thus, we defined a single ECG beat image by positioning the Q-wave peak signal while eliminating the first and the last 10 ECG signals from the Q-wave peak signals. Based on the time information, a single ECG beat range can be defined with the following:

$$T(Qpeak(n-1)+10) \leq T(n) \leq T(Qpeak(n+1)-10)$$

For example, for a signal with 10 beats, 8 ECG beat segments would be converted to images.

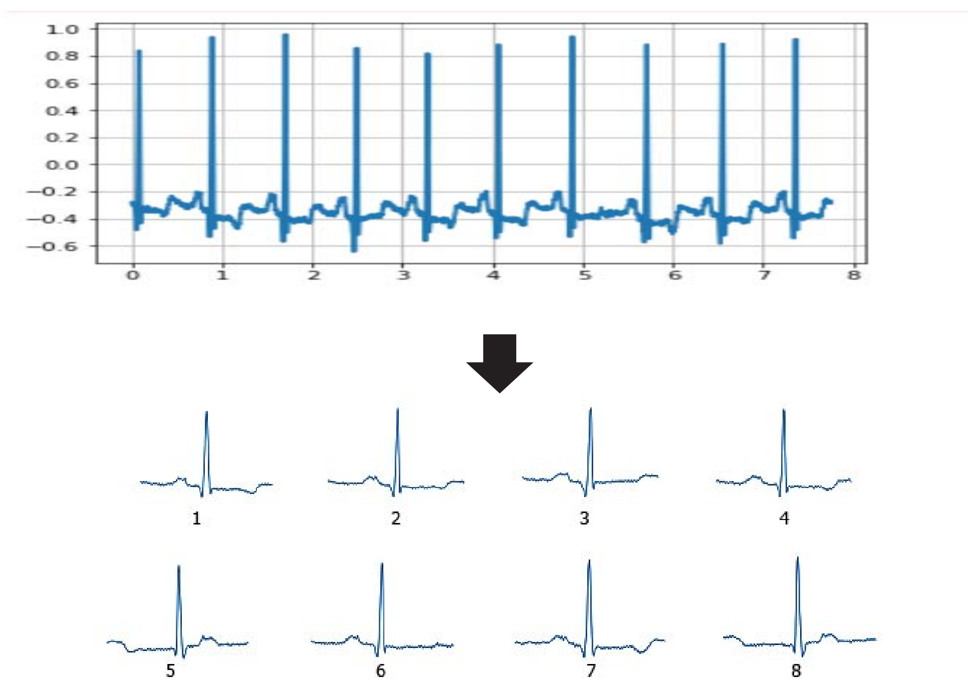


Figure 18 *Plotting each ECG beat as an individual 200 x 200 scale image*

We converted ECG signals into ECG images by plotting each ECG beat. We used the Biosppy module of Python for detecting the R – peak in the ECG signals. After the R-peaks were found, we took the present R-peak and the last R-peak, took half of the distance between the two, and included those signals in the present beat. Using this technique, we segmented R-peaks to a beat. We did this step for the next beat. We used Matplotlib and OpenCV to convert these segmented signals into grayscale images. Figure 18 shows the segmented signals.

5.2.2 Feature Extraction

Convolutional neural networks were first developed by Fukushima in 1980 and were improved in later years [18]. It is a form of DNN which involves one or more convolutional layers followed by one or more fully connected layers as in a standard multilayer neural network [18]. The main advantages of CNNs are that they are easier to train and have fewer parameters than fully connected networks with the same number of hidden layers [18]. CNNs are self-learned and self-organized networks which remove necessities of supervision. Nowadays, image classification, object recognition, and handwriting recognition are important concentrations of CNN. In addition, they play an important role in the medical field for automated disease diagnosis [19]. CNN does not need prerequisites such as pre-processing of datasets and separate feature extraction techniques, but some machine learning algorithms do. This makes CNN advantageous and reduces liability during training and picking the best feature extraction procedure for the automatic detection of arrhythmias [18,19]. We used a kernel size of 3 × 3 for all the convolutional layers, then we proceeded to Batch-normalization and ReLU activation. After the spectrogram conversion, the convolutional layers were arranged into 6 Convolutional Blocks in which each block had four layers. The number of filters was initially set to 32 for the first three convolutional layers but increased by 32 in the last layer of each convolutional block and this last layer also applied stride 2 while all other layers kept a stride of 1[15]. We reduced the size of the output image after each block by using stride 2 for the last layer in each block. We used an ECG image with 200 X 200 grayscale image. This resulted in a 200 x 200 x 1 input dimension of the network. The Convolutional neural network at the output of the last Block provided for the feature aggregation.

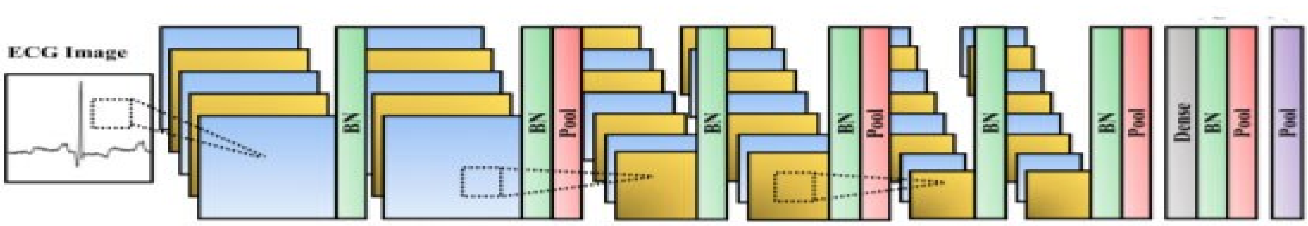


Figure 19 *Convolutional neural network of our proposed network*

5.2.2.1 Activation Function

The role of an activation function is to define the output value of kernel weights in the model. In modern CNN models, nonlinear activation is widely used, including rectified linear units (ReLU), leakage rectified linear units (LReLU) [20], and exponential linear units (ELU) [21]. While ReLU is the most widely used activation function in CNN, a small negative value is generated by LReLU and ELU because the ReLU translates whole negative values to zero. This results in the dropping of participation of some nodes in learning. We used ELU after the experiment as the performance for ECG arrhythmia classification was better than LReLU. ReLU, LReLU, and ELU are shown in the following [19]:

$$\text{ReLU}(x) = \max(0,x)$$

$$\text{LReLU}(x) = \max(0,x) + \alpha min(0,x)$$

$$\text{ELU}(x) = \begin{cases} x & if\ x \geq 0 \\ \gamma(exp(x) - 1) & if\ x < 0 \end{cases}$$

5.2.3 Aggregation of features across

While feature selection removes characteristics from the input file, feature aggregation combines input features into a smaller set of features called aggregated features. Variable length outputs are produced when the Convolutional Blocks process the variable length input of ECG signals in full length. These variable length outputs need to be gathered across time before they are fed to a standard classifier, which typically needs the dimension of the input to be unchanging. Averaging can be used to attain temporal aggregation in our CNN Architecture.

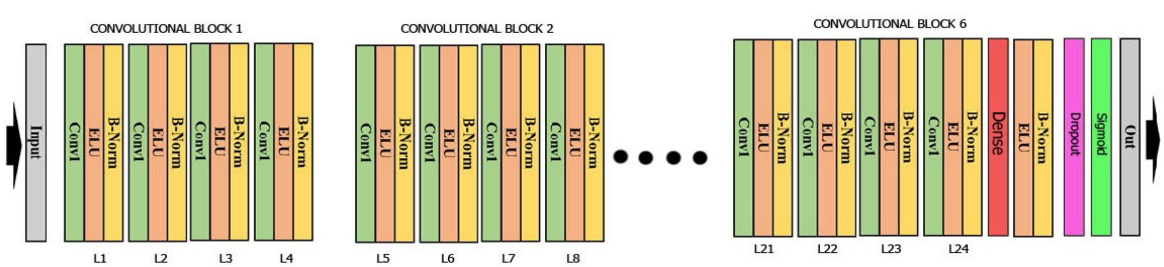


Figure 20 *Architecture of proposed CNN Model*

5.3 Data Set

The ECG arrhythmia recordings were retrieved from the MIT-BIH arrhythmia database. The database holds 8528 single lead ECG recordings of length varying from 9 to 61. The ECG recording is sampled at 360 samples per second. The MIT-BIH database contains approximately 110,000 ECG beats with 15 different types of arrhythmia including normal. The aim of this paper is to validate the performance of the proposed CNN. From the MIT-BIH database, each record was labelled as normal beat (NOR), AF rhythm, other rhythm, and noise record. For our network architectures we used the cross-entropy loss (reweighted as to account for the class frequencies) as a training objective and employed the Adam optimizer with the default parameters recommended in [22]. The batch size was set to 64. We used 7177 Normal beat ECG Images (class 0), 8917 Paced rhythm ECG images (class 2) and 472 Other rhythm ECG images (class 1). In total, we used 16566 images as a data set before using data augmentation and K fold cross validation.

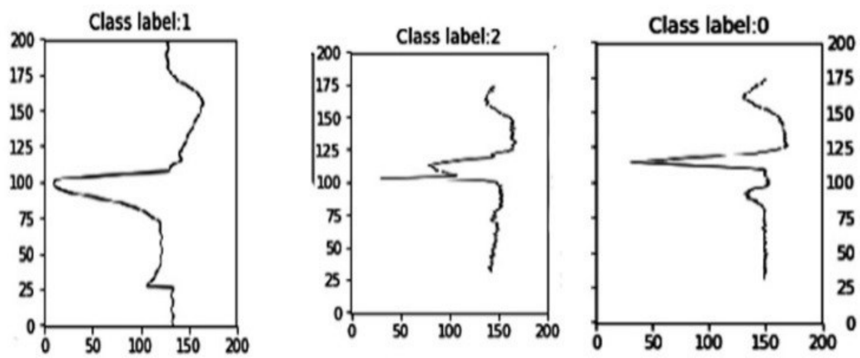


Figure 21 *Spectrogram of a sample data instance belonging to each class*

5.3.1 Data Augmentation

The poor generalization performance of a model is a result of overfitting, which occurs due to training on too few examples. Infinite training data can eradicate overfitting as every possible instance can be considered. Obtaining new training data is not easy in most machine learning applications, especially in image classification tasks, thereby limiting us to the training set at hand. We can, however, generate more training data through data augmentation, which enhances the training data by randomly transforming the existing data by generating new examples. Therefore, overfitting is reduced through the artificial boosting of the size of the training set. Data augmentation can also be considered as a regularization technique. When we were trying our model, we found serious overfitting in preliminary experiments. This can be approved based on the fact that the number of parameters in the proposed architectures is large compared to the size of data set exploited for evaluation. It was demonstrated in [23] that data augmentation can regularize and prevent overfitting in neural networks and improve classification performance in problems with imbalanced class frequencies [24].

In our dataset the third class (Other rhythm ECG images) are very few compared with the other two classes, so we used data augmentation to increase the number of data sets for this class to 7740 images.

Therefore, we augmented Other rhythm ECG images with nine different cropping methods: left top, center top, right top, center left, center, center right, left bottom, center bottom, and right bottom. Each cropping method results in the size of an ECG image, that is 128 x 128 grayscale. These augmented images are then resized to the original size, which is 200 x 200.

5.3.2 Training and Evaluation

After data augmentation K fold cross validation, the proposed CNN algorithms used 953360 ECG beat images for training and 238340 ECG beat images for validation. Furthermore, 5056 ECG image were used for testing. We trained the CNN end-to-end from scratch without encountering any issues. Training the convolutional layers in the CNN from scratch, on the other hand, did not lead to convergence. We therefore used feature averaging across time and the convolutional layers, which were trained together with a linear classifier for 150 epochs. We also used K fold cross validation to overcome overfitting.

5.3.2.1 K-Fold Cross Validation

K-fold cross-validation is a validation technique for steadying the performance of the statistical model when the data set is comparatively small. The whole data set is partitioned into K different subsets and repeatedly completes the training with K-1 subsets while evaluation is made with a single subset until all K subsets are evaluated. Cross-validation is used for overcoming overfitting. We used our original training data to generate several mini train-test splits. After creating train-test splits, we used them to tune our model. According to the definition of k-fold cross-validation, we divided the data into k subsets called folds. Our data is divided into 5 different subsets (or folds). Those 4 subsets are used as training data and we left the remaining subset (or the last fold) as test data. We finalized the model by averaging the model against each of the folds. After that, we tested it against the test set.

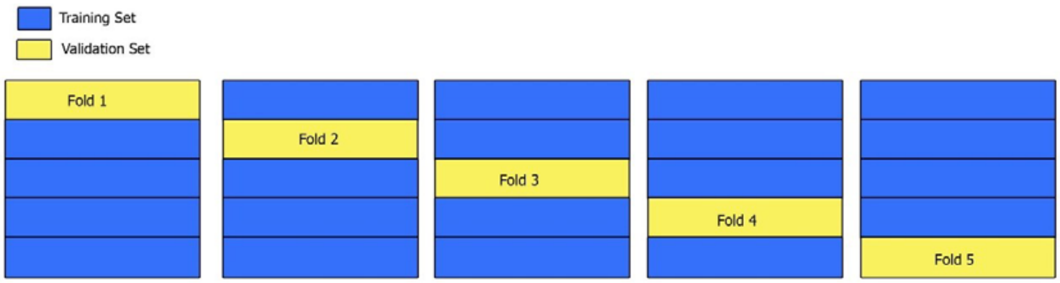


Figure 22 *K – fold cross validation*

5.3.3 Testing of Data

The algorithm does test on the CNN model to give test accuracy after completion of each training epoch. Our CNN algorithms used 150 epochs for the test data set. After completion of every epoch, we used 20% of the data as validation part to improve accuracy. Twenty percent of the total training data (70% of the original dataset) was used as the validation part and was used to improve accuracy.

5.4 Results

Our research further shows the important role of CNN in extracting all the dissimilar features, which are comparatively invariant to local spectral and temporal variations. This has resulted in higher accuracy performance. The proposed CNN algorithm contains three stages: (1) data pre-processing of input, where ECG signals are processed so that the computer can understand different diseases, (2) stacking of convolution layers to extract the features, and (3) layering of a fully connected layer and activation of the sigmoid function, which will predict the disease.

Table3 shows the parameters of the CNN layers and their filter size and output size. The proposed CNN algorithm was used to classify between Normal sinus rhythm (class 0), Paced rhythm (class 2), and Other rhythm (class 1). We used 24 hidden layers. The ReLU function was used to activate each hidden layer and batch normalization was used to normalize the input layer by adjusting and scaling the activations. After the convolutional layers, the resulting outputs were passed to reshape them. At the output of the layer, a linear activation function was then implemented.

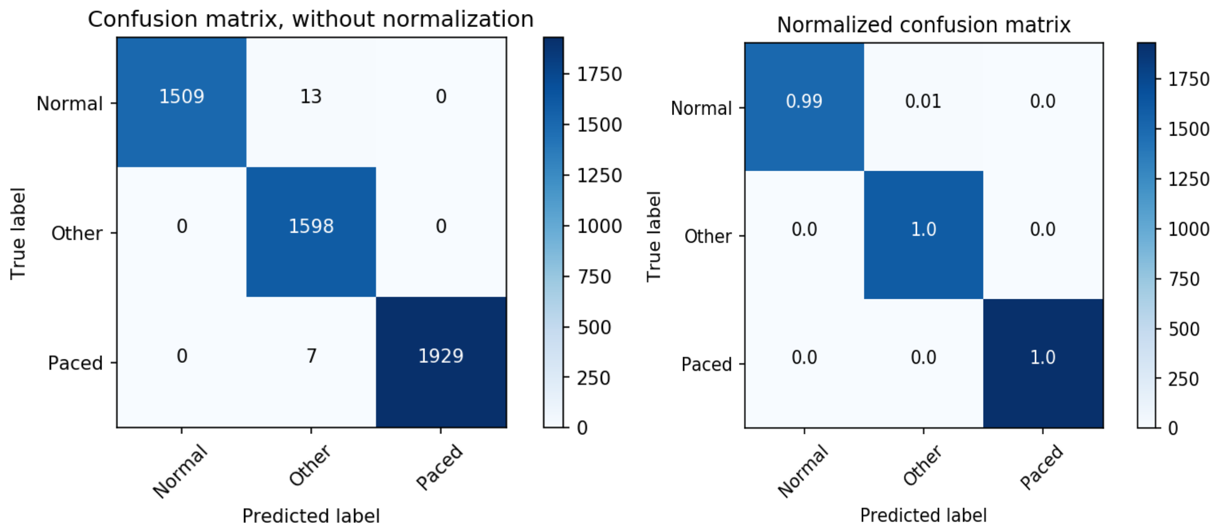
Layers	Type	Kernel Size	Stride	#Kernel	Output size
Layer 1	Conv2D	3 x 3	1	32	200 x 200 x 32
Layer 2	Conv2D	3 x 3	1	32	200 x 200 x32
Layer 3	Conv2D	3 x 3	1	32	200 x 200 x 32
Layer 4	Conv2D	3 x 3	2	64	100 x 100 x 64
Layer 5	Conv2D	3 x 3	1	64	100 x 100 x 64
Layer 6	Conv2D	3 x 3	1	64	100 x 100 x 64
Layer 7	Conv2D	3 x 3	1	64	100 x 100 x 64
Layer 8	Conv2D	3 x 3	2	96	50 x 50 x 96
Layer 9	Conv2D	3 x 3	1	96	50 x 50 x 96
Layer 10	Conv2D	3 x 3	1	96	50 x 50 x 96
Layer 11	Conv2D	3 x 3	1	96	50 x 50 x 96
Layer 12	Conv2D	3 x 3	2	128	25 x 25 x 128
Layer 13	Conv2D	3 x 3	1	128	25 x 25 x 128
Layer 14	Conv2D	3 x 3	1	128	25 x 25 x 128
Layer 15	Conv2D	3 x 3	1	128	25 x 25 x 128
Layer 16	Conv2D	3 x 3	2	160	13 x 13 x 160
Layer 17	Conv2D	3 x 3	1	160	13 x 13 x 160
Layer 18	Conv2D	3 x 3	1	160	13 x 13 x 160
Layer 19	Conv2D	3 x 3	1	160	13 x 13 x 160
Layer 20	Conv2D	3 x 3	2	192	7 x 7 x 192
Layer 21	Conv2D	3 x 3	1	192	7 x 7 x 192
Layer 22	Conv2D	3 x 3	1	192	7 x 7 x 192
Layer 23	Conv2D	3 x 3	1	192	7 x 7 x 192
Layer 24	Conv2D	3 x 3	2	224	4 x 4 x 224
Layer 25	Full			1024	3
Layer 26	Out				3

Table 3 *Architecture of proposed CNN Model*

The network was trained with 150 epochs and 50 steps per epoch. It gave an accuracy over 90% for the MITBIH arrhythmia database. Figure 23 shows the confusion matrix for the validation part of the dataset. The confusion graph is a graph which plots the true label versus the predicted label. As shown in the graph, the blue square indicates the high number of correct responses and the white square indicates the low number of incorrect responses. The dataset contains a total of 23834 ECG recordings. 7177 are Normal sinus rhythm (Class 0), 7740 are Paced rhythm ECG (Class 2), and 8917 are Other rhythm (Class 1). After K-fold cross validation, we used 80% of the data for training, which is 953360 ECG signal images and 20% of the data for validation, which is 238340 ECG signal images.

From the Validation data 5056, 1509 Normal sinus rhythm, 1929 Paced rhythm ECG and 1598 other rhythm signals were successfully classified by the algorithm, an improvement in the accuracy of the CNN model. Figure

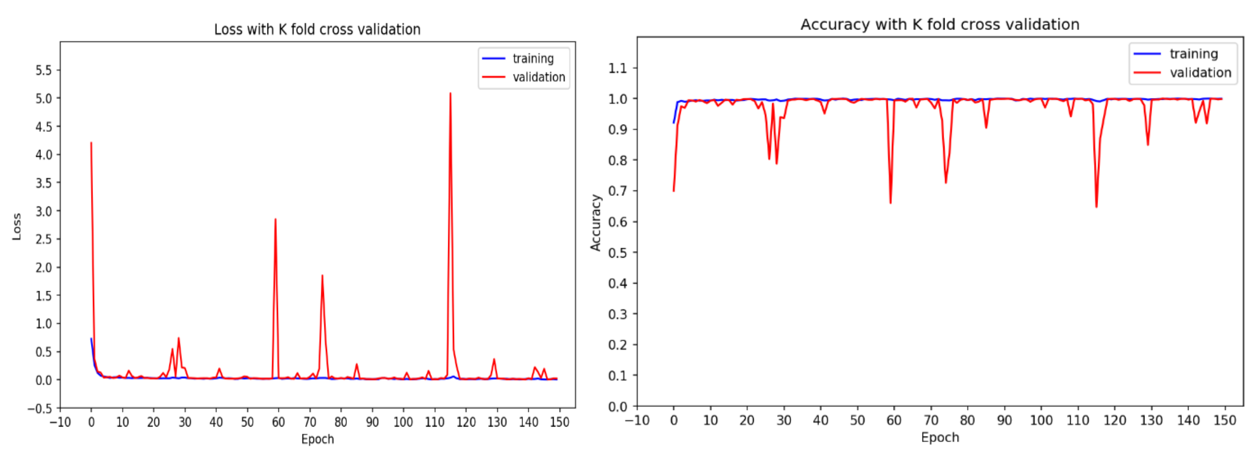
10 shows a graphic representation of the confusion matrix for the CNN algorithm. The network provides a reasonable prediction accuracy for the diseases. We expect a reasonable confusion because of unbalanced classes in the data set.



(a) (b)

Figure 23 *Confusion matrix (a) with normalization and (b) without normalization of the CNN algorithm.*

Fig 24 shows that the model converges very quickly and presents over 90% accuracy for the validation set. The noticeable peaks in the validation accuracy are most likely due to the unbalanced classes in the data set. This effect might be reduced by adding weight factors to the loss function, which would penalize those weights that belong to higher-frequency classes [25].



(a) (b)

Figure 24 *(a) train and validation loss graph (b) train and validation accuracy graph*

Chapter 6 Conclusion

The first process of this project in which data is collecting using ECG sensor AD 8232 from the patient. Then convert the analog signal from the sensor to digital and make it ready for transmission using Arduino. The ECG signals from the acquisition system are then wirelessly transmitted to a remote PC using Zigbee technology. MATLAB is used to plot the signal in PC and to store the signal in .csv format. Xbee S2 modules are used for transmission and reception. We use deep neural network to process the stored data.

In order to process the stored data to predict if there is any arrhythmia with ECG, we used an applicable ECG arrhythmia classification technique using convolutional neural networks with ECG images as inputs. 200 x 200 grayscale images were converted from a PhysioNet/CinC Challenge 2017 dataset ECG recording. 238340 ECG beat images were attained with three types of ECG beats including Normal sinus rhythm (Class 0), Paced rhythm ECG (Class 2), and Other rhythm (Class 1). An enhanced CNN model was created with significant concepts such as data augmentation, regularization, and *K*-fold cross-validation. The proposed algorithms resulted in successful classification of disease states in each signal with significant accuracy, using CNN models (Table A1). As a result, the proposed algorithms can achieve efficient diagnoses of various cardiovascular diseases with the accuracy of over 90%. The results show that developing a real time human health monitoring platform using deep machine learning and artificial intelligence (AI) can be an important method to help the experts analyze cardiovascular diseases using ECG signals. Furthermore, the proposed ECG arrhythmia classification method can be applied to medical robots or scanners that can monitor the ECG signals and help medical experts identify ECG arrhythmia more precisely and easily.

REFERENCE

[1] William Brims, "Wireless ECG," vol. I Thesis, The University of Queensland, October 2002.

[2] Cowley, M. (2002). A Practical Guide to ECG Monitoring and Recording. Found http://www.mikecowley.co.uk/leads.htm.

[3] J. M. Irvine, S. A. Israel, W. T. Scruggs, and W. J. Worek, “Eigen Pulse: Robust human identification from cardiovascular function,” Pattern Recognition, vol. 41, no. 11, pp. 3427–3435, 2008.

[4] Obermeyer Z, Emanuel EJ. Predicting the Future—Big Data, Machine Learning and Clinical Medicine.

[5] The New England journal of medicine 2016; 375(13): 1216–9. pmid:27682033.

[6] Dreiseitl S, Ohno-Machado L. Logistic regression and artificial neural network classification models: A methodology review. Journal of Biomedical Informatics 2002; 35(5–6): 352–9. pmid:12968784

[7] Berglund E, Lytsy P, Westerling R. Adherence to and beliefs in lipid-lowering medical treatments: A structural equation modeling approach including the necessity-concern framework. Patient Education and Counseling 2013; 91(1): 105–12. pmid:23218590

[8] AD8232 based Smart Healthcare System using Internet of Things (IoT):International Journal of Engineering Research & Technology (IJERT) http://www.ijert.org ISSN: 2278-0181 IJERTV7IS040040 Vol. 7 Issue 04, April-2018

[9] The ECG Biomedical Engineering the University of Connecticut. BME Program Director, John D. Enderle.

[10] Electrocardiogram (ECG or EKG). https://www.heart.org/en/health-topics/heart-attack/diagnosing-a heart attack/electrocardiogram-ecg-or-ekg

[11] SparkFun Single Lead Heart Rate Monitor AD8232 https://www.sparkfun.com/products/12650

[12] ZigBee Networks Open the Door to More Wireless Medical Devices, Medical Design, April 2005.

[13] Getting Started with ZigBee and IEEE 802.15.4, Daintree Networks: http://www.science.smith.edu/~jcardell/Courses/

[14] Acharya, U.R.; Fujita, H.; Lih, O.S.; Adam, M.; Tan, J.H.; Chua, C.K. Automated detection of coronary artery disease using different durations of ECG segments with convolutional neural network. *Knowl. Based Syst.* **2017**, *132*, 62–71. [CrossRef].

[15] Nilanon, T.; Yao, J.; Hao, J.; Purushotam, S.; Liu, Y. Normal/abnormal heart recordings classification by using convolutional neural network. In Proceedings of the IEEE Conference on Computing in Cardiology Conference (CinC), Vancouver, BC, Canada, 11–14 September 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 585–588.

[16] Convolutional Recurrent Neural Networks for Electrocardiogram Classification. Martin Zihlmann, Dmytro Perekrestenko, Michael Tschannen, Dept. IT & EE, ETH Zurich, Switzerland.

[17] A convolutional neural network to detect atrial fibrillation from a single lead ECGhttps://github.com/awerdich/physionet.

[18] Shadi, G.; Mostafa, A.; Nasimalsadat, M.; Kamran, K.; Ali, G. Atrial fibrillation detection using feature-based algorithm and deep conventional neural network. In Proceedings of the Conference on Computing in Cardiology (CinC), Rennes, France, 24–27 September 2017; IEEE: Piscataway, NJ, USA, 2017.

[19] Cardiac Arrhythmia Classification by Multi-Layer Perceptron and Convolution Neural Network Shalin Savalia and Vahid Emamian.

[20] Maas AL, Hannun AY, Ng AY (2013). Rectifier nonlinearities improve neural networkacoustic models. International Conference on Machine Learning 30(1):3.

[21] Clevert DA, Unterthiner T Hochreiter S (2015). Fast and accurate deep network learningby exponential linear units (elus). arXiv preprint arXiv:1511.07289.

[22] Kingma DP, Ba J. Adam: A method for stochastic optimization. In Proc. Int. Conf. on Learn. Representations (ICLR). 2015;

[23] Simard P, Steinkraus D, Platt J. Best practices for convolutional neural networks applied to visual document analysis. In Proc. Int. Conf. on Document Analysis and Recognition. 2013; .

[24] Chawla N, Bowyer K, Hall L, Kegelmeyer W. Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 2002.

[25] Chow, G.V.; Marine, J.E.; Fleg, J.L. Epidemiology of arrhythmias and conduction disorders in older adults. *Clin. Geriatr. Med.* **2012**, *28*, 539–553. [CrossRef] [PubMed]

MIT-BIH Arrhythmia Database https://www.physionet.org/physiobank/database/mitdb/