Fifth International Conference on Internet of Things and Applications (IoT 2021)

Sensing, Wireless Transmission, and Smart Processing of Heart Signals

Erfan mansoori
Electrical and Computer Engineering
University of Kashan
Kashan, Iran
Email: e.mansoori@std.kashanu.ac.ir

Alireza Siavashi
Electrical and Computer Engineering
University of Kashan
Kashan, Iran
Email: siavashiar@std.kashanu.ac.ir

Mahdi Majidi
Electrical and Computer Engineering
University of Kashan
Kashan, Iran
Email: m.majidi@kashanu.ac.ir

Abstract— **In recent years, by increasing the number of epidemics, the concept of telemedicine and its combination with the Internet of Things (IoT), has become a more prominent subject in engineering. The purpose of this study is to implement an IoT system that can sense the heart signal and detect the arrhythmia in a heartbeat by using smart signal processing and send them through the internet to a relative of the patient or doctor. In this system, Electrocardiogram (ECG) signals are received by electrodes and sent to the AD8232 chip. Then, by connecting to the ESP32 module, information through the Wi-Fi network and Message Queuing Telemetry Transport (MQTT) protocol, is sent to the broker created on Raspberry Pi. Then, using Python and through the Decision Tree algorithm, the heartbeat's performance is evaluated as a smart process. The accuracy of the algorithm in detecting arrhythmias is over 95%. Finally, with an android application and through the TCP/IP protocol, the health or arrhythmia of the heartbeat will be informed on the doctor's or one of the patient's relative's cell phone, and its real-time display will be provided. Also, the possibility of sending a command from a mobile to the broker has been provided.**

Keywords: Internet of things, Machine learning, MQTT protocol, Smart signal processing, Android program.

I. INTRODUCTION

In today's world, despite the rapid advancement of technology, many people lose their lives in a moment of seconds due to heart arrhythmia and lack of timely care. In this situation, even a fraction of seconds may save a patient's life. Also, due to the exponential trend of population growth, increasing pandemic diseases such as Covid-19, and also, the lack of medical centers in some remote areas, the existence of remote monitoring on medical products has become a critical issue for researchers[1]. Early detection of heart arrhythmia for preventive measures is one of the fundamental issues in telemedicine. For this purpose, in some articles, the concepts of arrhythmia detection in a heartbeat are expressed using different algorithms. In [2], using Recurrent Neural Network, a method is designed to detect Atrial Fibrillation arrhythmia. Also, in [3], based on the combination of SVM and Genetic algorithms, classifications have been made for Electrocardiogram (ECG) arrhythmias.

On the other hand, monitoring heartbeat based on the Internet of Things (IoT), is another significant issue in the field of telemedicine. Therefore, research projects have been done to send and display heartbeat on mobile phones remotely, using Bluetooth and Wi-Fi [4]. In order to transmit the ECG signal farther, in [5], the ECG signals received from the sensor, are sent to the mobile phone via the internet. To share information for more people, higher security, access at the desired time, and also not to lose information, in [6] and [7], a system is implemented to send information to the Cloud server using the HTTP protocol, for doctor's access at any time. To design lighter, energy consumption, and less traffic consumption, in [8], a research has been done on the transfer of information to the Cloud server using the Message Queuing Telemetry Transport (MQTT) protocol. Next, to combine the topics of smart and early detection of arrhythmia in the Internet network and monitoring of heart signal remotely based on IoT, a design is presented in [9]. To receive the heart signal from the AD8232, in [9], a large board with high power consumption is designed that makes it uncomfortable to connect to the body. The board processes the signal features and then sends them, while these processes could be done on a server or cloud to reduce power consumption and size.

According to the mentioned research projects, the need for a complete system containing all the mentioned items is felt. To put it more clearly, in some of the above articles, only the issue of arrhythmia analysis has been examined. In other cases, only heart signal monitoring based on IoT is mentioned. Some of them use the cloud, which costs a lot in the long-term and is not suitable for environments with a small number of patients or a small number of sensors. In some cases, the absence of a mobile phone in the receiver and as a result, the lack of portable information terminal was seen.

The purpose of this paper is to design and implement a prototype system that can cover all the above cases. Early and remote detection of arrhythmia and alerting to the doctor or one of the patient's relatives immediately, monitoring the heartbeat through the internet for the doctor to investigate and decide on the treatment process, in addition to using the MQTT protocol with light design and low traffic consumption are other goals of this paper. The implementation of the above project has become an essential issue in the field of telemedicine.

In the rest of the papers, in Section II, it will be talked about the structure of the system and how to implement its hardware and software. The third section deals with the numerical results of the system implementation, and finally, the fourth section will conclude and introduce future perspectives.

Figure 1. Block diagram of the project.

II. System Model and its Hardware and Software Implementation

In Fig. 1, a block diagram of all implemented items in the project can be seen.

A. *Collecting ECG Signal and Transmit to ESP32 Module*

The AD8232 module is a physiological device for measuring the heartbeat, and its features include filtering and amplifying small signals. According to Fig. 1, to collect ECG signals, one of the methods (1) and (2) can be used to supply the signal of module AD8232.

Method (1) involves receiving a real heart signal by attaching electrodes to the body, and method (2) involves producing an artificial signal with the desired arrhythmia by the ECG Simulator. ECG Simulator is a device that produces ECG signals artificially. One of the applications of this device is the possibility of producing different types of ventricular arrhythmias on the normal ECG signal. As shown in Fig. 2, the output of the AD8232 module goes to the ESP32 module, and the information is sent to the Raspberry Pi Broker using the IEEE 802.11 standard on the Wi-Fi network. ESP32 is a beneficial IoT module, with capabilities such as data transfer via Wi-Fi and Bluetooth.

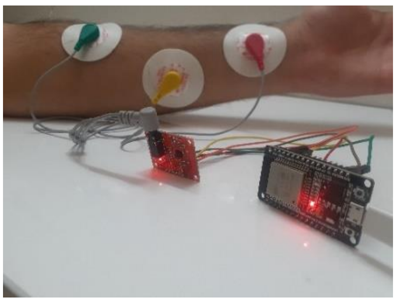

Figure 2. Receiving the heart signal by module AD8232 and transmitting it to the ESP32 module.

Also, by receiving real heart signals from sites such as Physionet.org, it is possible to obtain different people's heart signals. The Physionet site is an authoritative and extensive database of physiological signals. If heart data on medical site is used, by connecting the MATLAB serial port to the ESP32 module, information is sent to the ESP32 module to be sent to the broker via the Wi-Fi network. It should be noted that in the simulation and results observation section, the real heart signals available on medical sites will be used with the help of MATLAB serial port.

B. *Introduction and Configuration of MQTT Protocol in Different Parts*

MQTT is one of the most popular protocols in the IoT that operates based on a publish-subscribe pattern. Key features of this protocol include lightweight design, less power consumption, and less bandwidth than other protocols. The function of this protocol is that each board or sensor sends its information under a specific title to a server known in MQTT as a Broker. On the receiver side, by subscribing to the relevant title, the information of that particular title can be received. The reserved port for this protocol is 1883.

To implement the MQTT protocol, we consider the ESP32 module as the sender client, the Raspberry Pi board as the broker, and the Python programming language as the receiver client. In the Raspberry Pi section, using the Linux commands in the virtual terminal, we implement the Mosquitto broker. Encryption has also been performed on the MQTT broker to ensure high system security. In this case, only people with the username and password of the broker can access it.

In the ESP32, by using the C++ programming language, and by having the Internet Protocol address (IP), username, and password of the broker, we can access the broker. In the Python section, as in the ESP32 section, with broker information, we can enroll for the desired title.

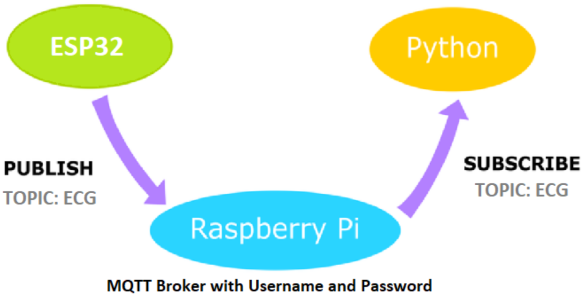

Figure 3. Schematic of the MQTT protocol implemented in the project.

C. Sending Information from ESP32 and Smart Processing in Python to Detect Arrhythmia

Initially, by connecting the ESP32 module to the Access Point, the possibility of sending information over the Wi-Fi network is prepared. In Fig. 4, first, the confirmation of the ESP32 module connection to the Access Point, the local IP taken by ESP32, and the confirmation of its connection to the broker configured on the Raspberry Pi, can be seen in the Arduino Serial Monitor.

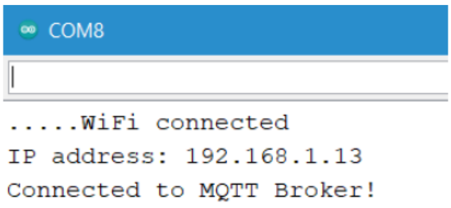

Figure 4. ESP connection confirmation to MQTT broker and Wi-Fi, with local IP.

Then, with the successful connection of ESP32 and Python to the broker, the heart signals are sent to the broker via ESP32, through the Internet network and MQTT protocol, under the title of ECG. Next, in the receiver section, the Python language that had previously registered in the ECG title, will receive the information. After receiving each signal, the heartbeat function is evaluated. Smartly detectable arrhythmias in this section are AFIB (Atrial Fibrillation) and Bigeminy arrhythmias, which are evaluated along with the NSR (Normal Sinus Rhythm) signal by the Decision Tree algorithm.

AFIB is a type of heart arrhythmia and often rapid heartbeat that results in a high mean heartbeat and can increase the risk of heart attack as well as other heart-related complications. Although AFIB is not usually fatal by itself, it is a serious problem and sometimes requires immediate treatment. The heartbeat in this disease may be 100 to 175 beats per minute. In Fig. 5, the AFIB and NSR signals are compared in terms of the number of beats.

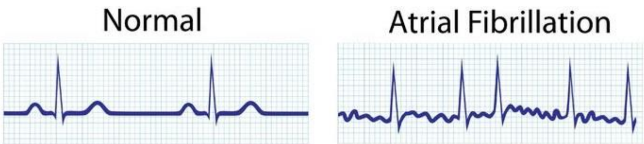

Figure 5. Comparison of AFIB and NSR in terms of the heartbeat.

To interpret Bigeminy arrhythmia, PVC arrhythmia is first described. PVC results from an early impact on the ventricle, and because this shock comes from the ventricle, its QRS expands. There is usually a compensatory phase after PVC arrhythmia, but this is not always true. Also, if a PVC arrhythmia is found after each sinus wave and this condition is repeated intermittently, Bigeminy arrhythmia occurs. Due to the compensatory phase, the R-R distance will increase in this arrhythmia. Also, since a PVC occurs in the Bigeminy arrhythmia after each sinus wave, every other RR interval is larger than the normal. Fig. 6, shows the occurrence points of PVC with an asterisk, along with creating an incremental distance between R-R distances in the Bigeminy arrhythmia.

Considering what was said about two arrhythmias of AFIB and Bigeminy, it can be mentioned the feature of high average heartbeat in AFIB arrhythmia, as well as the increase of R-R distances due to the compensatory phase in Bigeminy arrhythmia. We will use these two features in the Decision Tree algorithm to separate the signals.

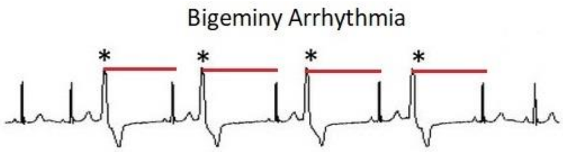

Figure 6. Bigeminy arrhythmia with increasing R-R distances, along with PVC occurrence points in it.

To study the two features of R-R intervals as well as the average heartbeat to create data set, various signals, including the NSR signal, along with the Bigeminy and AFIB arrhythmias, are received from Physionet.org and then examined in the Python environment. Initially, the above signals pass through a low pass Butterworth filter to eliminate 50 Hz noise. Next, the signal peaks are determined by obtaining the signal average and increasing it by 15% (to identify the peaks correctly). Fig. 7, shows the peaks detection in the NSR signal.

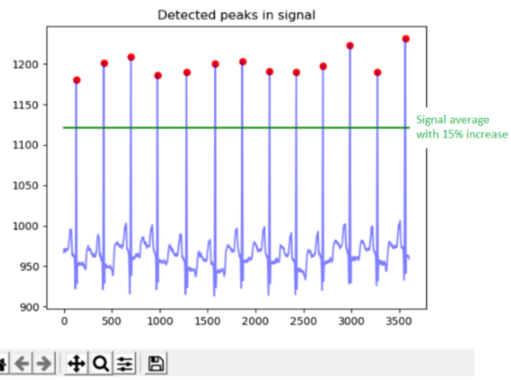

Figure 7. Detection of signal peaks by using the mean and increasing it by 15% as a threshold.

After detecting the peak points in the signal, the R-R distances can be calculated. Also, using the average R-R intervals per minute, the average heartbeat is calculated based on the mean line (index line).

In order to determine the features for the Decision Tree algorithm, two features of the average heartbeat per minute (which high heartbeat is one of the main features in AFIB arrhythmia), plus a sum of two larger R-R interval (which larger R-R distance is one of the main features in Bigeminy arrhythmia due to the compensatory phase) are used.

The Decision Tree algorithm builds classification or regression models into the structure of a tree, except that it has grown from the root down (leaf). In fact, the Decision Tree breaks down the data set into smaller subsets. The decision node in this algorithm has two or more branches. The leaf node represents a classification. Also, the root node is the highest decision node in this algorithm, which corresponds to the best decision. By considering various indicators and different methods for determining the Decision Tree, various algorithms are expressed for it. One of the most important in the field of medical sciences is the CART (Classification and Regression Tree) algorithm. This method, which creates a Decision Tree with binary divisions, is designed for quantitative variables but can be used for any variable type. In this method, the Gini index is introduced as a criterion for selecting appropriate variables. In the tree model with binary divisions, other criteria such as entropy can be used, but the advantage of the Gini criterion is the higher speed in calculations. The Gini index indicates the probability of a particular feature being wrongly selected. If all elements are in a class, the Gini coefficient will be zero.

According to [10], in order to determine the best feature for the root node in the machine learning algorithm, we calculate the Gini coefficients for both features, which are expressed as continuous numbers. The relationship of the Gini index to node T is as follows:

$$Gini(T) = 1 - \sum_{i=1}^{C} (P_i)^2, \quad (1)$$

where P_i is the probability of classifying an item in a particular class, and C is the number of classes. If node P is divided into K segments, the total weight of Gini indexes is as follows:

$$Gini\ Split(A) = \sum_{i=1}^{K} \frac{n_i}{n}\, Gini(i), \quad (2)$$

where A is the desired property, K is the number of children in the desired property, n_i is the number of samples in section i, and n is the number of samples in node P. In order to calculate the Gini index for each of the two properties, which are numerically continuous, the values of the continuous samples are first sorted. Next, between every two consecutive numbers, an average is taken, which is introduced as the representative of those two numbers. Finally, for each representative, the Gini index is calculated and the smallest index is introduced as the Gini index of that attribute. After calculating the smallest index between the two attributes, the attribute with the lower Gini index is selected as the root node and the criterion for starting the tree. After the Decision Tree is formed, the different signals are classified by this algorithm and its health or arrhythmia along with the signal itself will be sent to the doctor's cell phone or one of the relatives via TCP/IP protocol, which will be examined later.

D. Sending Information to the Mobile Phone and Real-Time Displaying

After smartly processing the heart signals using the Decision Tree algorithm, the health or arrhythmia of the heartbeat, along with a real-time display of the signal, by the server and via the TCP/IP protocol, will be sent to the doctor's mobile phone or one of the relatives of the patient. The purpose of this section is to implement a two-way communication channel between Python and mobile phone, to exchange information processed over the Internet or Wi-Fi. In order to achieve this, Android programming is used in the Android Studio environment along with the TCP/IP protocol. In this state, Python, which previously acted as a client for the MQTT protocol, this time acts as a server in the TCP/IP protocol. The mobile phone will also receive information from the server as a customer. The port used to exploit the TCP/IP protocol is 8080. In order to provide emergency instructions, the possibility of having a two-way communication between the server and the doctor has been provided.

III. System Implementation and Numerical Results

In order to evaluate the performance accuracy of the machine learning algorithm, 136 heart signals, including 43 NSR signals, 45 Bigeminy signals, and 48 AFIB signals, are considered as data set in the form of two R-R interval features and the average heartbeat based on the index line. Thirty percent of the 136 data sets will be used as experimental data. After examining 136 different signal items, a data set consisting of two features and three classes is formed in order to use the machine learning algorithm. class (0) belongs to the NSR, class (1) belongs to the Bigeminy arrhythmia, and class (2) belongs to the AFIB arrhythmia. A view of the Decision Tree formed can be seen in Fig. 8.

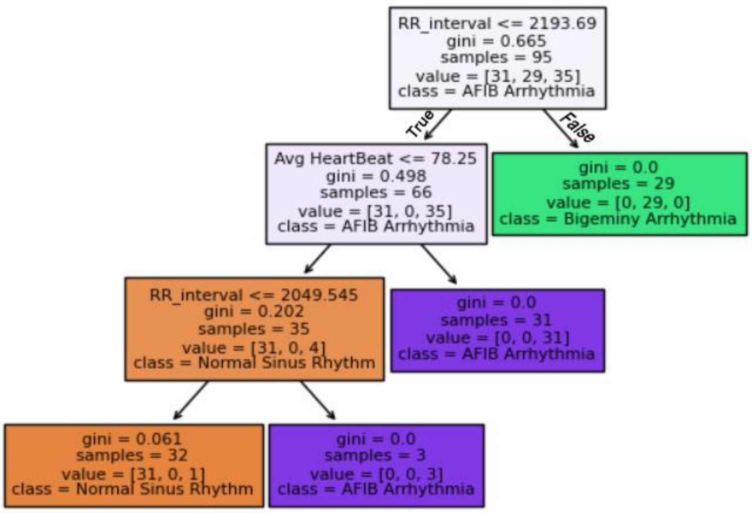

Figure 8. The decision tree formed with the Gini index criterion to classify 3 different heart signals.

In Fig. 8, green belongs to the Bigeminy arrhythmia, purple belongs to the AFIB arrhythmia, and orange belongs to the Normal signal. According to Fig. 8, 95 data are given to the algorithm for learning. After the studies based on the Gini index criterion between the two features of R-R interval and the average based on the index line, the number 2193.69 (The number inside the block at the root node), which was previously present as one of the continuous representatives in the R-R interval feature, due to having the lowest Gini index, as the best feature to consider in the root node is selected. The way of calculating the Gini index value for this representative is as follows:

$$Gini\ Split(R\_R) = \sum_{i=1}^{2} \frac{n_i}{n}\, Gini(i)$$

According to the presence of 3 classes and also by knowing the probability of the presence of each sample in each class, Gini (1) and Gini (2) can be calculated as follows:

$$Gini(1) = 1 - \left(\left(\frac{31}{66}\right)^2 + \left(\frac{0}{66}\right)^2 + \left(\frac{35}{66}\right)^2\right) = 0.4981$$

$$Gini(2) = 1 - \left(\left(\frac{0}{29}\right)^2 + \left(\frac{29}{29}\right)^2 + \left(\frac{0}{29}\right)^2\right) = 0$$

$$\rightarrow Gini\ Split(R\_R) = \frac{66}{95}\, Gini(1) + \frac{29}{95}\, Gini(2) = 0.346$$

Table I. shows some of the Gini indexes calculated for both attributes.

TABLE I. GINI COEFFICIENTS CALCULATED FOR SEVERAL DIFFERENT REPRESENTATIVES IN BOTH PROPERTIES AND SELECTING THE LOWEST COEFFICIENT FOR USING IN THE ROOT NODE.

R-R interval (MS)	Gini index	Average Heartbeat	Gini index
2193.69	0.34609	78.25	0.40254
2226.72	0.36166	78.6	0.40293
2156.15	0.36172	77.85	0.40315
1731.22	0.62378	65.05	0.54731
3066.06	0.62432	106.85	0.63828

If the answer of RR_interval<=2193.69 is negative in the root node, all the Bigeminy arrhythmia data is categorized in the first step in its class. If yes, the second property is considered to separate the remaining two signals. As the process continues, a tree with a depth of 3 will be formed.

The way of working this algorithm in separating the three signals expressed, has been shown in Fig. 9.

In Fig. 9, class (0) belongs to the NSR signal, class (1) belongs to the Bigeminy arrhythmia, and class (2) belongs to the AFIB arrhythmia. Markers surrounded by circles are the result of the test process. The accuracy of the algorithm for detecting any of the mentioned arrhythmias is presented in TABLE II.

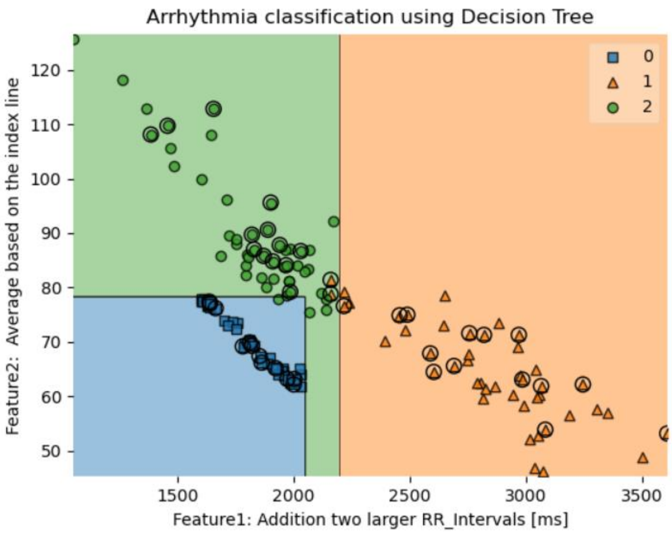

Figure 9. A view of the separation of 3 signals NSR, Bigeminy, and AFIB by Decision Tree algorithm.

TABLE II. PERCENTAGE OF CORRECTLY DETECTING THE PERFORMANCE OF THE DECISION TREE ALGORITHM IN DETECTING ARRHYTHMIA.

Arrhythmia	Percentage of diagnosis
Bigeminy	95.55
AFIB	97.91
NSR	99.8

In our system after learning and in the deployment phase, when the ECG signal is received from AD8232, the two features of R-R interval and average heartbeats will be extracted with a python script that has been run in the Raspberry Pi. These features of the patient's heart signal, supply input for our decision tree algorithm. Then, the algorithm using these inputs and creates logical conditions in each node detects which arrhythmia has occurred. Also, after detecting the heartbeat function at each level, a real-time representation of the above signal is presented, and the type of arrhythmia is expressed in the title of the shape being drawn on the monitor attached to the Raspberry Pi, as shown in Fig. 10.

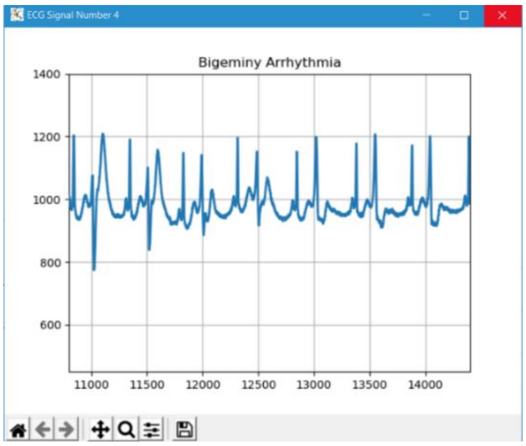

Figure 10. Smart detection of heart arrhythmia with a real-time drawing on the monitor connected to a Raspberry Pi.

The signal observed on the doctor's mobile phone for examination is as shown in Fig. 11.

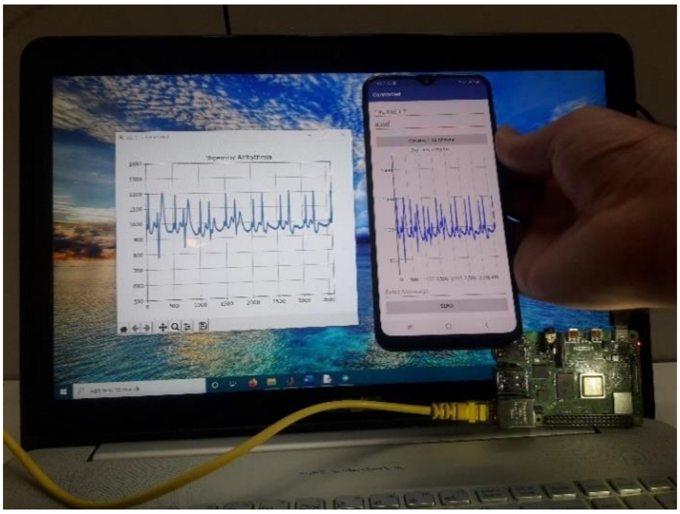

Figure 11. Evaluation of heart function, along with real-time displaying of ECG signal on the mobile phone.

Finally, a list of equipment used in this project along with their applications and features in the TABLE III Provided.

TABLE III. APPLICATIONS AND FEATURES OF THE DEPLOYED HARDWARE MODULES.

Device	Applications and Features
AD8232	ECG sensor Two-poles high-pass filter Three pole low-pass filter
Raspberry Pi 4	Smart signal processor MQTT broker and HTTP server 4GB RAM, Wi-Fi, LAN Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz, 5V3A
ESP32 dev	Transmits the ECG Signals by Wi-Fi MQTT Client 2-ADC with 7channel-12bit accuracy UART, Wi-Fi, ... 512 kB RAM Dual core, 240MHz CPU Clock
Cell Phone	Signal displaying – Transmit Command HTTP Client Supports Android 8 or up

IV. CONCLUSION AND FUTURE PERSPECTIVE

In this paper, a prototype system for detecting arrhythmia in a heartbeat, using the Decision Tree algorithm and a lightweight protocol called MQTT, along with ECG monitoring on the Internet, was designed. The implementation of two-way communication between the server in the Python section and the doctor on the mobile side using the TCP/IP protocol was another case implemented in this paper. In order to upgrade or commercialize the project in the future, it can be possible to evaluate more features and arrhythmia by upgrading the algorithm used in the project and also using other algorithms. Vital signals from multiple patients can also be transmitted and processed.

ACKNOWLEDGMENT

We would like to thank Dr. Hossein Rabbani for allowing access to the laboratory facilities of the Isfahan University of medical science, and Ms. Karimi for making the ECG Simulator available. Finally, we would like to thank Ms. Maryam Hosseini and Asal Mansoori for helping us to design the system model, Shiva Mansouri, an obstetrician, and Dr. Sima Mohammadi for their valuable feedback, which was widely used in the implementation of this paper.

REFERENCES

[1] A. M. Borujeni, M. Fathy, and N. Mozayani, “Developing and Evaluating A Real time and Energy Efficient Architecture for An Internet of Health Things,” in *Proceeding of 4th International Conference on Smart Cities, Internet of Things and Applications, SCIoT 2020*, pp. 106–111, Sep. 2020.

[2] J. E. S. Simanjuntak, M. L. Khodra, and M. C. T. Manullang, “Design Methods of Detecting Atrial Fibrillation Using the Recurrent Neural Network Algorithm on the Arduino AD8232 ECG Module,” *IOP Conf. Ser. Earth Environ. Sci.*, vol. 537, no. 1, 2020.

[3] J. A. Nasiri, M. Naghibzadeh, H. S. Yazdi, and B. Naghibzadeh, “ECG arrhythmia classification with support vector machines and genetic algorithm,” in *EMS 2009 - UKSim 3rd European Modelling Symposium on Computer Modelling and Simulation*, pp. 187–192, 2009.

[4] A. S. Prasad and N. Kavanashree, “ECG monitoring system using AD8232 sensor,” in *Proceedings of the 4th International Conference on Communication and Electronics Systems, ICCES 2019*, pp. 976–980, Jul. 2019.

[5] V. K. Bhosale and K. R. Bhosale, “Healthcare based on IoT using Arduino and AD8232 heart rate monitoring chip,” *Asian J. Converg. Technol.*, vol. II, no. 2350, pp. 3–5, 2016.

[6] A. Mishra, B. Chakraborty, D. Das, P. Bose, and U. G. Students, “AD8232 based smart healthcare system using Internet of Things (IoT),” *Int. J. Eng. Res. Technol.*, vol. 7, no. 4, April. 2018.

[7] M. Chhabra and M. Kalsi, “Real time ECG monitoring system based on Internet of Things (IoT),” *Int. J. Sci. Res. Publ.*, vol. 7, no. 8, p. 547, 2017.

[8] A. Mishra, A. Kumari, P. Sajit, and P. Pandey, “Remote web based ECG monitoring using MQTT protocol for IoT in healthcare,” *Int. J. Adv. Eng. Res. Dev.*, vol. 5, 2018.

[9] A. Mishra and M. Mohapatro, “An IoT framework for bio-medical sensor data acquisition and machine learning for early detection,”, *Artic. Int. J. Adv. Technol. Eng. Explor.*, 2019.

[10] Ruisi He and Zhiguo Ding, *Applications of Machine Learning in Wireless Communications*. pp. 4-7, 2019.