CLOUD-BASED ECG CLASSIFICATION WITH MOBILE INTERFACE

Study Group 1

Epperson, Joseph David (jde160530@utdallas.edu)
Hoff, Jason (jxj143230@utdallas.edu)
Noor Shoudha, Shamman (sxn170028@utdallas.edu)

***Abstract*— Cloud computing can be a useful tool to analyse biosignals; when that methodology is combined with over-the-air transfer of data, it allows healthcare providers to access relevant information on a lightweight mobile interface. Here, we develop an Android application that receives an ECG signal over Bluetooth, plots the data stream, and allows a user to send a biosignal analysis request to determine if any abnormality is present in the ECG signal. The application can receive the result of the analysis within seconds of time, allowing healthcare providers to efficiently analyse incoming data.**

***Keywords*— ECG, Pan-Tomkins, SVM Classifier, Android, Bluetooth**

I. INTRODUCTION

In this project, we used previously described processes with a new interface to analyze ECG signals from MIT-BIH Arrhythmia Database in MATLAB. This was done by connected a user via Android application to a server capable of applying the Pan-Tompkin algorithm to find the P, Q, R, S, and T waves of an ECG signal. The server can extract the features of the processed waves and use them as inputs of a supervised machine learning algorithm. We implemented this Support Vector Machine (SVM) to learn the characteristics of normal and abnormal heartbeats from the morphological features of the extracted waves.

The trained model that was obtained from our SVM can be used as a tool during ECG analysis. We have created an Android application that can be used to interface with an ECG signal access point, such as a server with stored data. By inputting a new ECG signal into the classifier, we can compare the feature vector of the new signal against our SVM model to determine the number of beats that were analyzed, in addition to the number of abnormal beats that were detected. Figure 1 illustrates the flow of data.

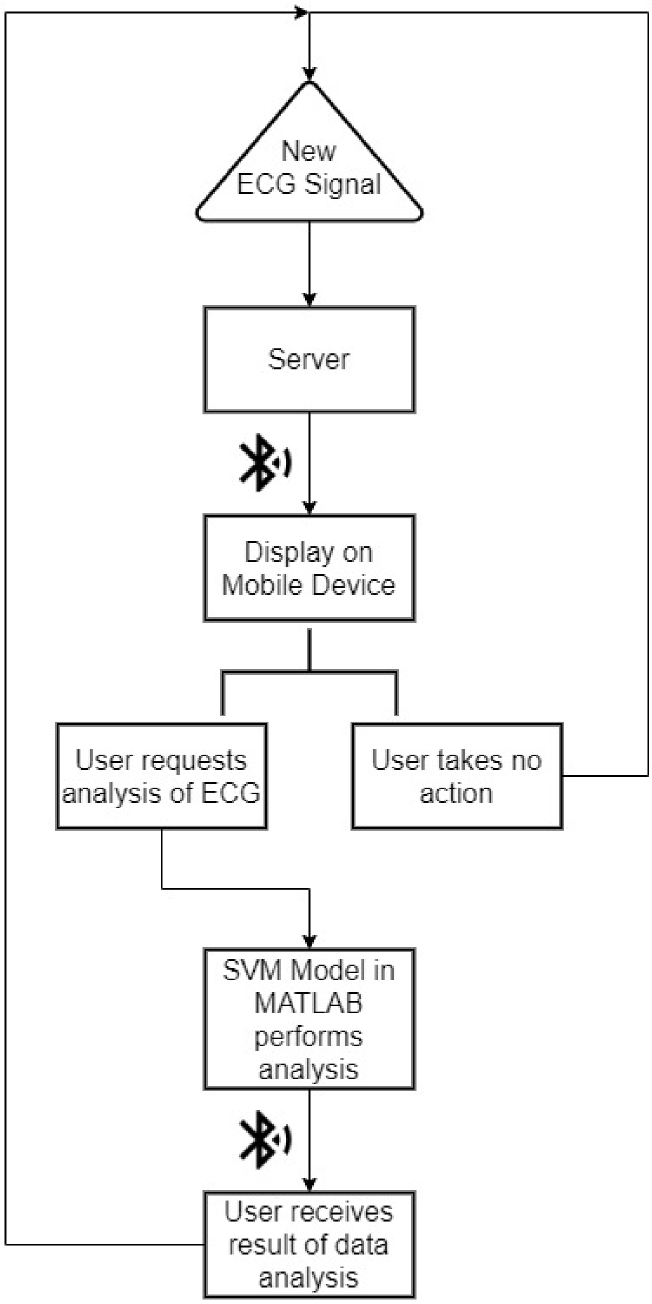

FIGURE 1: FLOW OF DATA

II. APPLICATION INTERFACE

The application was designed with the user in mind, providing immediate access to plotting of data when the application is opened. The main screen of the application contains a button to access a list of Bluetooth access points, a graph upon which to plot an ECG signal, buttons to send a request for data analysis and to clear the current data stream, and a text field that displays the results of analysis. See Figure 2 for a screenshot of the main screen.

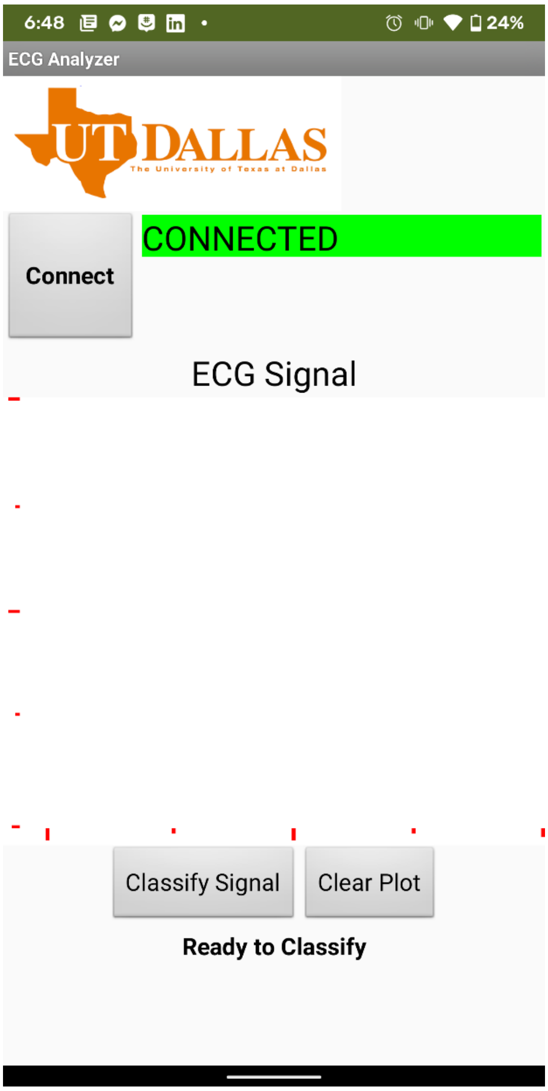

FIGURE 2: MAIN SCREEN

III. USER EXPERIENCE

The app must be connected to the appropriate server via Bluetooth for maximum functionality; when a user launches the application, they must access the list of Bluetooth access points to receive an ECG signal. Once a signal has been acquired, the server begins to send data. The application has been developed to plot any data it receives when no other user inputs have been detected. Figure 3 illustrates an example of a plotted ECG signal on the Android application.

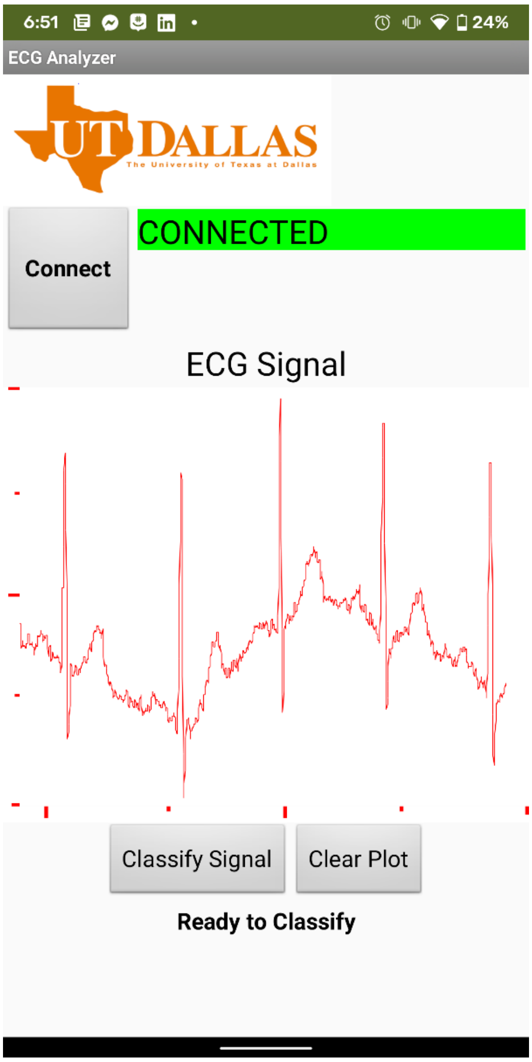

FIGURE 3: PLOTTING ECG DATA

After viewing the ECG, the user can click a button labelled "Classify Signal" to send a request to the server. The server receives the classification request and sends the displayed portion of the signal to the SVM model for analysis. Once the analysis has been performed, the user receives a short report, showing the number of beats that have been analysed and the number of beats that were detected as abnormal. See Figure 4 for an example of the short report. Once the short report has been received, the user can clear the window and results with a "Clear Plot" button near the bottom of the screen. If no action is taken, the application begins to request data again after 30 seconds from the end of plotting.

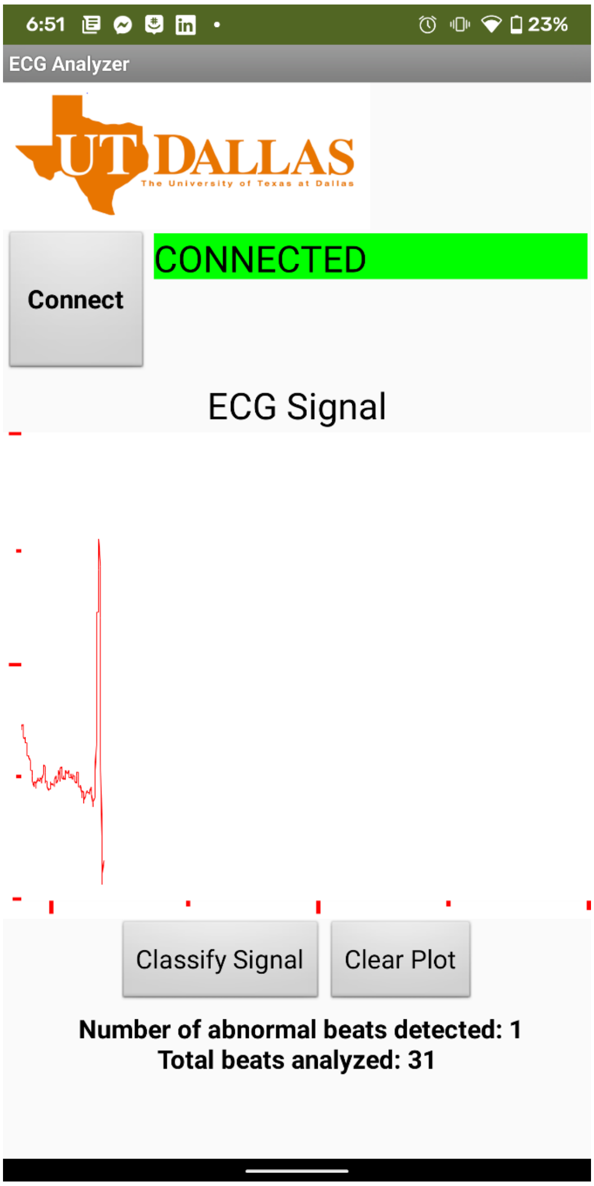

FIGURE 4: SHORT REPORT EXAMPLE

IV. HARDWARE

The prototype of the device created for this project used external, off-the-shelf hardware for wireless communication. A SparkFun BlueSMiRF Silver was integrated with an Arduino Mega 2560 and an LG gram laptop for the server to mobile device communication. A Google Pixel 2 XL was used as the mobile device for testing. Figure 5 illustrates these components (LG gram excluded).

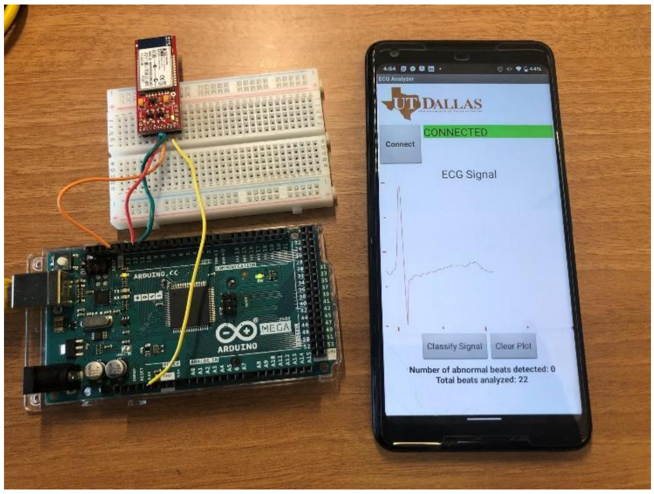

FIGURE 5: HARDWARE ILLUSTRATION

V. SOFTWARE

Serial communication for data transfer via Bluetooth was used between the MATLAB application and the Pixel 2 XL device. The EGC signal was transferred in 12-byte packages to stay below the limit of the serial buffer. The Arduino sketch was written using the Software Serial library to send incoming serial data to the BlueSMiRF for outgoing serial communication, and incoming data from the BlueSMiRF to the outgoing serial port of the Arduino.

The Android application was developed with MIT App Inventor 2, which provided a cloud-based GUI for Android development and live testing. The GUI is illustrated in Figure 6.

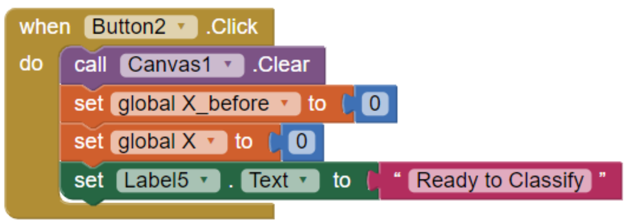

FIGURE 6: FLOW OF DATA

VI. CONCLUSION

The development of this application showed that challenges in combining cloud computing with biosignal analysis can be done with elementary hardware and software. Additional work could be done on this project to implement a cloud database for integration at any internet access point, removing the need for a local server. The development of the application could be progressed to meet additional user needs, such as identification and plotting of specific abnormal waveforms for analysis.

REFERENCES

[1] Y. Zhu, “SVM Classification Algorithm in ECG Classification,” Commun. Comput. Inf. Sci., vol. 308 CCIS, no. PART 2, pp. 797–803, 2012.

[2] W. J. Tompkins, “A Real-Time QRS Detection Algorithm,” IEEE Trans. Biomed. Eng., vol. 32, no. 3, 1985.

[3] T. Munasinghe, E. W. Patton and O. Seneviratne, "IoT Application Development Using MIT App Inventor to Collect and Analyze Sensor Data," 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 6157-6159, doi: 10.1109/BigData47090.2019.9006203.