CS3237

FINAL PROJECT

Group 9

JOSEPHINE TANADI (A0170123X)

LIM KIAN HONG (A0168695H)

PANG JIA JUN VERNON (A0185646R)

PHAM QUANG MINH (A0170723L)

Table of Contents

1. Introduction

Sudden cardiac arrest (SCA) is a condition in which the heart suddenly and unexpectedly stops beating. When this happens, blood stops flowing to the brain and other vital organs. Time is of the essence when treating a victim of SCA; irreversible damage to the brain cells occurs just after several minutes of oxygen deprivation.

According to the Singapore Heart Foundation (Singapore Heart Foundation, n.d.), the immediate cause of SCA is often a sudden development of an abnormal rhythm of the heart (known as ventricular fibrillation (VF)) in which there is chaotic electrical activity without mechanical contraction. This results in a rapid cessation of blood circulation. VF may be preceded by ventricular tachycardia (VT) where there is a rapid heart rate e.g. 200 BPM but blood pressure may be very low.

SCA is one of the leading causes of mortality in Singapore. In 2016, the survival rate for SCA is a low 23.4%. Besides that, more than 2,000 out-of-hospital SCA's occur nationwide every year. This meant that 70-80% of SCA cases occur either at home or in public places.

Our proposed solution, hence, aims to maximise the survival rate of SCA through the development of a comprehensive IoT healthcare system. This system comprises 3 main components, namely a wrist-worn edge device, an Android smartphone (gateway), and a laptop (cloud). The overall operation of the system (*see Appendix A*) employs two distinct strategies to achieve its main goal: the instant detection and early prediction of SCA in users.

The instant action aspect of the IoT system involves the prompt and accurate identification of SCA in users. This is accomplished by computing and tracking the heart rate of users in real-time where the user will be wearing the edge device with the help of the 3D printed case (*see Appendix B*). This aspect is isolated in the edge device and gateway platform without any involvement from the cloud.

The long-term aspect of the IoT system, on the other hand, involves the early prediction of SCA in users. This is accomplished by monitoring the periods at which users experience high <u>physical</u> or <u>psychological</u> stress. Physical stress typically derives from exercise and, hence, can be classified as a positive form of stress. Psychological stress originates, however, from unpleasant and stressful situations such as work pressure, which can then be classified as a negative form of stress. According to the American Heart Association (American Health Association, 2014), negative stress is often associated with the increased risk of heart disease and, ultimately, cardiac arrest. Therefore, it is, without a doubt, a good thing to have high positive and low negative stress, and our system strives to assist users to achieve that goal. This analysis takes place entirely and independently in the cloud platform, where trends derived will eventually be sent to the gateway device to empower users to make more informed decisions with regards to their current lifestyle habits.

2. Techniques Employed

Given below are some of the main techniques we have employed to achieve the aforementioned goal of maximising the survival rate of SCA in Singapore. Implementation details can be found in Section 3 of the report.

Communication

- Bluetooth Low Energy (BLE) to transmit sensor data from edge device to gateway platform
- Message Queuing Telemetry Transport (MQTT) to transmit sensor' data from gateway to the cloud platform

Software Development

- Android application as a gateway platform to monitor vitals (for instant action) and acts as a communication bridge between the edge device and cloud platform

Signal Processing

- Raw photoplethysmogram (PPG) signal data processed and cleaned for further analysis
- Fast Fourier Transform (FFT) algorithm employed at the gateway platform to compute heart rate
- Heart Rate Variability (HRV) analysis performed at the cloud platform to analyse psychological stress

Machine Learning

- Human Activity Recognition (HAR) performed at the cloud with a Convolutional Neural Network (CNN) model to identify physical stress
- Transfer learning employed to develop a pre-trained Multilayer Perceptron (MLP) model to identify psychological stress

Miscellaneous

- 3D-printed wearable case created for the edge device (*see Appendix C*)
- Power management techniques implemented on the edge device

3. Implementation Details

3.1. Transmission of Sensor Data from Edge Device to Gateway Platform

Data from the following sensors were collected and transmitted via BLE to the gateway platform.

3.1.1 Pulse Oximeter (MAX30101)

Data from the pulse oximeter was collected to monitor the heart rate of the user. Following Maxim's official recommendation (Maxim, 2018) for wrist heart rate monitoring, we have targeted data collection from the green LED and disabled the red and infrared LEDs. This is done by initializing the Pulse Oximeter to Multi Mode, instead of the HR mode or SPO2 mode, which allows us to customize which LED we wish to enable or disable.

The code below shows how we have initialized the pulse oximeter at main.cpp *(Refer to /iot-device/HSP/main.cpp)*.

```
max30101.Multimode_init(0x0F, 0x04, 0x03, 0x03, 0x00, 0x00, 0xFF, 0x03,
0x00, 0x00, 0x00);
```

We defined a ticker within HspBLE.cpp *(Refer to /iot-device/HSP/Hsp_BLE/HspBLE.cpp)* to send PPG signals collected from the pulse oximeter at an interval of 4 ms by invoking the ppgHandler function.

```
ticker.attach(callback(this, &HspBLE::ppgHandler), 0.04);
```

This ticker interval is set at an interval of 4 ms to precisely synchronise the overall sampling rate (25 Hz) of the MAX30101 sensor.

3.1.2 Accelerometer (LIS2DH)

Data from the accelerometer was collected for Human Activity Recognition (HAR) (e.g. if the user is in an idle state, jumping, or running).

The code below shows how we have initialized the pulse oximeter at main.cpp *(Refer to /iot-device/HSP/main.cpp)*.

```
lis2dh.initStart(0x03, 0x00); // data_rate = 25 Hz, fifo_threshold = 0
```

Unlike the PPG signal data where data is streamed out every ticker interval, accelerometer data is sent after every interrupt defined by the data rate of 25Hz. The data rate of 25Hz ensures that transmission of accelerometer data is synchronized with the transmission of PPG signals.

3.1.3 Design Consideration

BLE was favoured over serial transmission to achieve the primary objective of a wearable IoT device. Using BLE, data could be transmitted wirelessly to the gateway platform and the edge device need not be attached to a gateway platform to stream data.

Similarly, PPG signals were chosen instead of ECG signals to improve the usability of the edge device as the user need not wear ECG cables while using the system. As PPG and ECG signals are highly correlated, collecting PPG signals would suffice in our use case to predict the heart rate of the user.

Unused sensors were switched off to improve the power consumption of the edge device. The sensors include:

- MAX30003 (ECG)
- BMP280 (Barometric Pressure)
- MAX30205 (Temperature Sensor)

Furthermore, as mentioned previously, for the pulse oximeter sensor, only the green LED was enabled and the red and infrared LED were disabled to further conserve power.

3.2. Handling of Sensor Data at the Gateway Platform

An Android device was used as the gateway instead of the Odroid platform. The Android device was chosen as the user will be wearing the edge device and having a portable gateway for sending and receiving information from the cloud would be extremely convenient and can achieve our intended objective of making the edge device a wearable and highly portable device.

The gateway device is mainly responsible for receiving sensor data from the edge device via BLE, process the data received for instant action, and forward the data received to the cloud platform via MQTT for processing.

The gateway device is also responsible for receiving processed data from the cloud via MQTT and displaying the received information in the application.

3.3. Cleaning the PPG Data

The PPG signal transmitted from the edge device to the gateway platform is tampered by sudden and unnatural “jumps” of the measured data as seen in Figure 3.3.1.

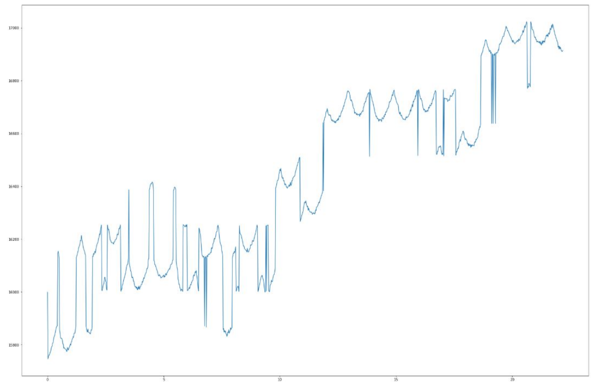


Figure 3.3.1: PPG signal received at the gateway from the edge device

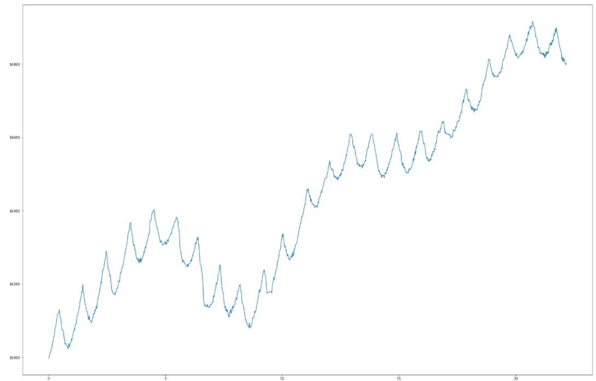


Figure 3.3.2: PPG signal data after being cleaned at the gateway

It can be seen from Figure 3.3.1 that the PPG signal is made discontinuous by jumps in values of similar size. To clean the PPG data, we make use of the fact that the jump is much larger than natural changes in the PPG data, so we categorise all changes that exceed a certain threshold as unnatural jumps. By keeping track of the total unnatural jumping distance, we can pull all signal data down to the same level as the original recorded data by simply subtracting the accumulated unnatural jump distance so far at that data point.

A problem with this solution is that the magnitude between the increasing jumps and the decreasing jumps are different, which causes the tracked total unnatural jump distance to accumulate over time. This makes the value of the cleaned signal slowly increases to infinity. To prevent this from happening, we introduce a decay value to the tracked total unnatural jump distance so that it can slowly converge back to zero without making a noticeable impact on the shape of the cleaned data.

The cleaned PPG signal data now shows clear peaks of the heartbeats as seen in Figure 3.3.2. This data is used at the gateway platform to find the heart rate of the wearer. It is also sent to the cloud for further analysis for more data extraction (refer to Section 3.6).

3.4 Calculating Heart Rate at Gateway Platform

The cleaned PPG data goes through multiple steps of signal processing before its frequency spectrum can be analysed with Fast Fourier Transform (FFT) to calculate the heart rate.

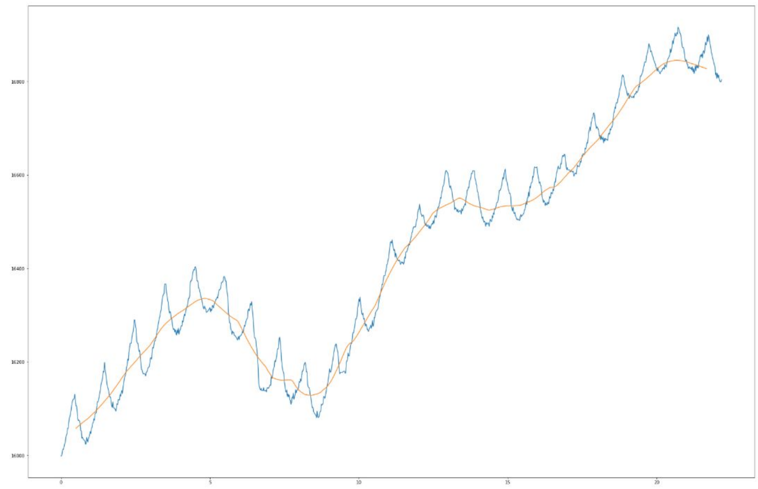


Figure 3.4.1: PPG signal and its moving average (large window)

Firstly, we subtract the cleaned PPG signal with its moving average line. The window size of the moving average line was chosen to be at least larger than 1 heart rate cycle, which eliminates the fluctuation caused by the heart rate in the moving average line as seen in Figure 3.4.1. The result of the subtraction (Figure 3.4.2) is a signal around the zero line which frequency spectrum is dominated by the heart rate frequency instead of the DC value

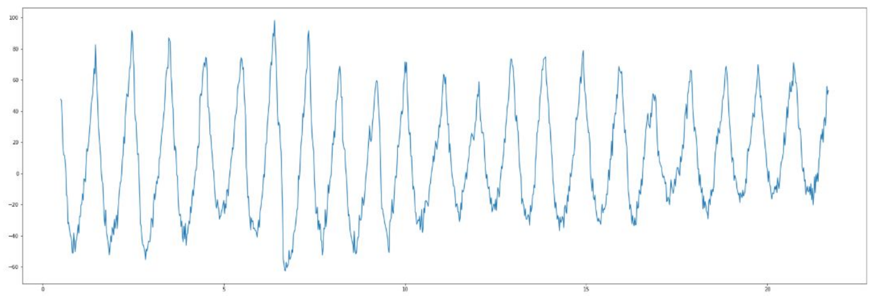


Figure 3.4.2: PPG signal after subtracting the moving average (large window)

After that, we extract the moving average line of the subtracted line above to eliminate the noise in the signal. This time, we chose the window size of the moving average to be much smaller than a heartbeat cycle (about 10 times smaller) so that it can follow the fluctuation of the heartbeat and smooth it along the way. The result of the extraction is a very clear fluctuation of the heartbeats without the noise found in the previous line (Figure 3.4.3).

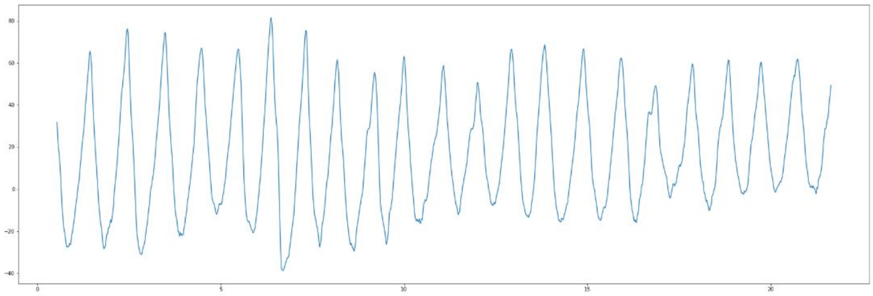


Figure 3.4.3: Extracted moving average (small window) from previous line

With the smoothed PPG heartbeats data available, we can now use Fast Fourier Transform (FFT) to analyse the frequency spectrum of the signal. FFT was implemented in the Android application of the Gateway using the Noise library, which works kuje the Android wrapper for the kissfft library for C. After calculating the magnitude spectrum of the signal, the highest peak of the signal will be found, and the heart rate is then estimated between the frequency of that highest peak and the previous/next discrete frequency point (depending on which magnitude is larger). The estimation is based on the magnitude ratio between the frequency of the highest peak and the chosen neighbour frequency point:

$$est\_heart\_rate = f_{max\_peak} \pm f_{resolution} \times \frac{mag_{next/prev\_peak} - mag_{prev/next\_peak}}{(mag_{max\_peak} - mag_{prev/next\_peak}) + (mag_{next/prev\_peak} - mag_{prev/next\_peak})}$$

The larger this ratio is, the closer the estimation of the heart rate is to the frequency of the highest peak in the magnitude spectrum.

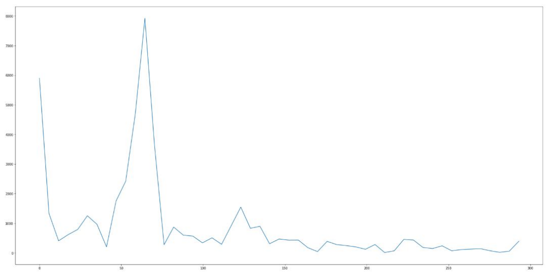


Figure 3.4.4: Magnitude spectrum of the processed PPG signal

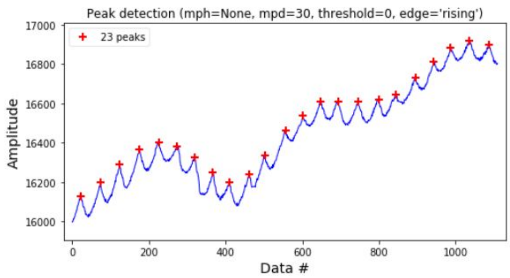


Figure 3.4.5: Peak count analysis on time-domain signal (Verification)

To verify that the analysis in the frequency domain is correct, we perform a separate time-domain analysis and compare the results of both methods of heart rate analysis. The time-domain analysis counts the number of peaks in the signal and uses it to estimate the heart rate. In this example, the frequency-domain analysis (Figure 3.4.4)

return the heart rate value of 63.29 bpm, while the time-domain analysis (Figure 3.4.5) found 23 peaks within 22.18 seconds, which is equivalent to a heart rate of 62.22 bpm. This shows that the frequency-domain analysis is correct and can be used to estimate the heart rate well.

Using the frequency analysis, sudden cardiac arrest (SCA) can be detected by finding the lack of heartbeat using these two conditions:

- Either if the calculated heart rate is smaller than 30 Hz.
- Or if the magnitude value of the highest peak does not exceed a certain threshold.

After detecting the absence of a heartbeat, the gateway will take appropriate instant actions to react to the situation.

To continuously calculate the heart rate over time, we define a constant window size of data to be analysed each time FFT is used (window size should preferably be a power of 2 to make FFT more efficient). At the start, there is a period where the window is still waiting for more data to fill it up and it is not ready to be analysed. After the window is filled, the heart rate can then be calculated continuously after every second. Data enter the filled window in the first in - first out order so that the latest signal data can be analysed.

One thing to note is that the resolution of the FFT (the distance between two nearest peaks in the distinct Fourier Transform) only depends on the time frame of the data fed to FFT:

$$f_{resolution} = \frac{60}{window\_time}\ (bpm)$$

This means that if FFT analyses a longer time frame of the signal, the accuracy of the calculation of the heart rate will increase. At the same time, a longer time frame for FFT analysis also means that it will take longer to detect a sudden change in heart rate, which means it would be slower in detecting SCA. This means that there is a tradeoff between accuracy and reaction speed of the heart rate calculation in this implementation.

3.5. Android application

The Android application installed on the Android gateway device is based on the existing MAXREFDES100 Android Application provided on the Maxim Integrated website. Modifications have been made to the “SensorActivity” page, not only that, “DetectionActivity”, “PredictionActivity” and “LongTermActivity” pages have been added that can be accessed from the “SensorActivity” page when the user first selects a device.

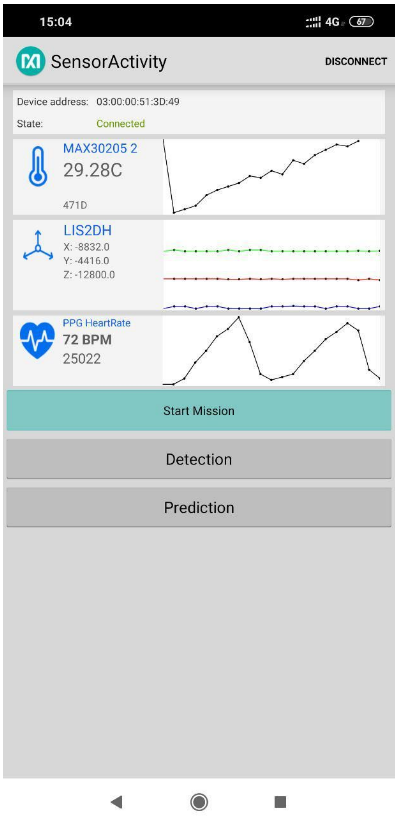


Figure 3.5.1: SensorActivity page

In the “SensorActivity” page, we are able to view sensor data received from the edge device using graphs for “Temperature”, “Accelerometer” and “PPG Heart Rate” data. From this page, we are able to access both “DetectionActivity” and “PredictionActivity” pages.

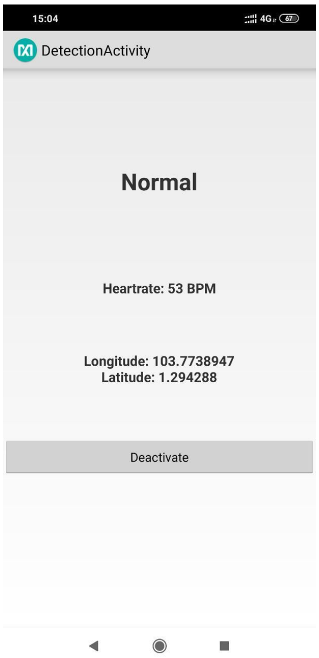


Figure 3.5.2: DetectionActivity page when user has "Normal" status

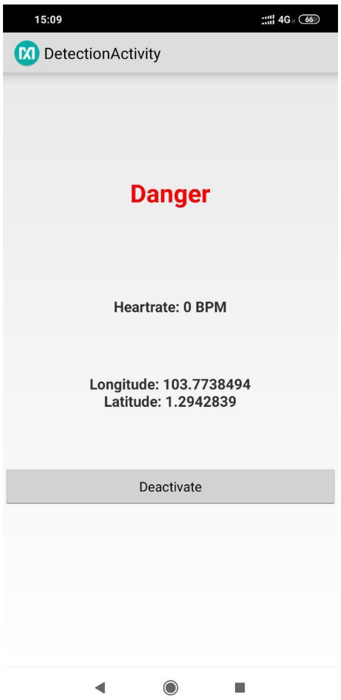


Figure 3.5.3: DetectionActivity page when user has "Danger" status

In the "DetectionActivity" page, the user is able to view their current status as determined by the application, heart rate, and location coordinates. Should the device start to vibrate and play buzzer sound from determining that the user is in danger, the user is able to deactivate them from this page should the alert be a false positive.

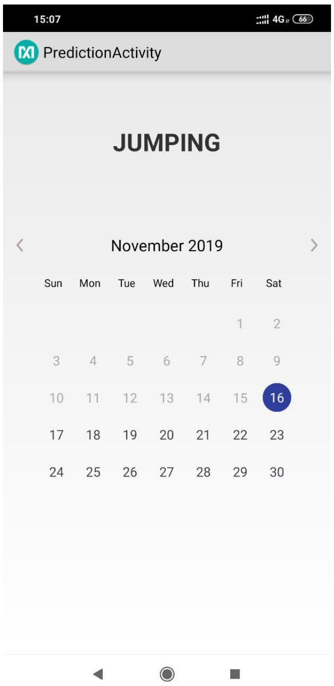


Figure 3.5.4: PredictionActivity page

In the “PredictionActivity” page, we are able to observe the current status of the user, whether they are “Idling”, “Running” or “Jumping”. This is so that we are able to co-relate any rise in the user’s heart rate to the activity that they are currently engaged in. Should there be a high heart rate, if the user is currently engaged in any strenuous activity we can determine that the high heart rate is not uncalled for. The user is then able to select a day from the calendar where he can view his vital statistics for the particular day.

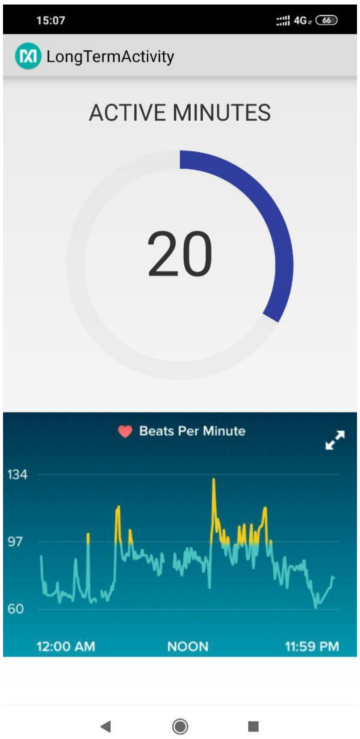


Figure 3.5.5: LongTermActivity

From the calendar in “PredictionActivity” page, the user will be directed to the “LongTermActivity” page where they can view the amount of time they have been exercising and a chart of their resting heart rate for the day. With the use of the algorithm that will be mentioned in the next section, we are able to cleanly separate the resting heart rate from the heart rates while doing strenuous activity. This information can be used to assist in the early prediction of SCA by observing the trends in the user’s resting heart rate.

3.6. Analysis of Heart Rate Variability (HRV)

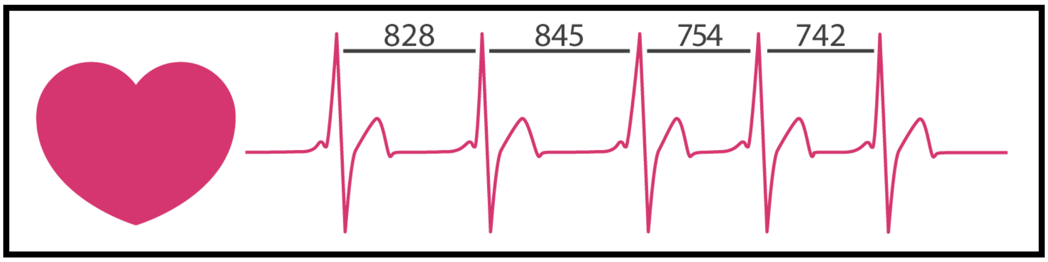


Figure 3.6.1: ECG graph showcasing heart rate variation

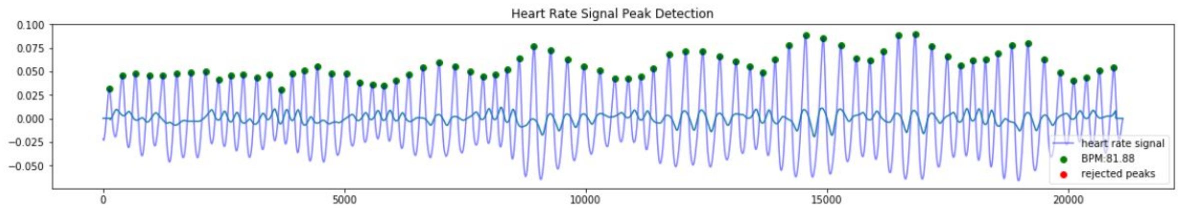
Heart rate variability (HRV) is the physiological phenomenon of the variation in the time interval between consecutive heartbeats in milliseconds. A normal, healthy heart does not tick evenly like a metronome, but instead, when looking at the milliseconds between heartbeats, there is constant variation (Figure 3.6.1).

In general, a low HRV (or less variability in the heartbeats) indicates that the body is under stress from exercise, psychological events, or other internal or external stressors. Higher HRV (or greater variability between heartbeats) usually means that the body has a strong ability to tolerate stress or is strongly recovering from prior accumulated stress. At rest state, a higher HRV is generally favourable.

HRV is a good indicator of one's general well-being and health. Over the past few decades, research has shown a relationship between low HRV and worsening depression or anxiety. A low HRV is also associated with an increased risk of death and cardiovascular disease.

There are numerous methodologies to calculate and analyse HRV. Luckily for us, there exists a python library known as HeartPy that is able to abstract the complex calculations involved in computing HRV. Given a time series of PPG data, this Python Heart Rate Analysis Toolkit will perform the following operations (refer to */iot_cloud/hrv/hrv_analysis (calm).ipynb* for the source code):

1. Scale the raw PPG signal received from the gateway
2. Enhance peaks in PPG signal (optional)
3. Filter noise in PPG signal using a bandpass filter [0.75 Hz to 3.5 Hz]
4. Compute HRV

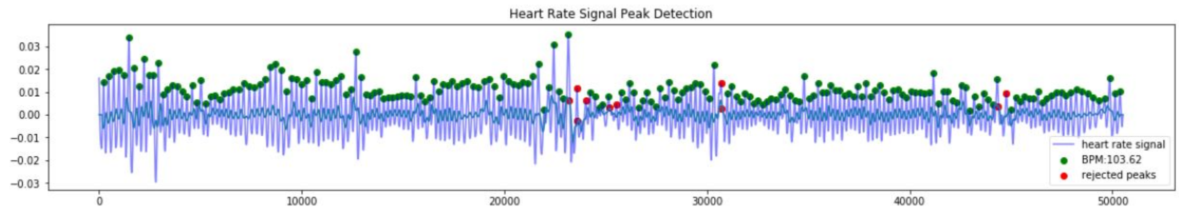



```
bpm: 81.879775
ibi: 732.781690
sdnn: 88.774312
sdsd: 62.546297
rmssd: 62.554177
pnn20: 0.328571
pnn50: 0.214286
hr_mad: 66.000000
breathingrate: 0.170374
```

Figure 3.6.2: HRV analysis for calm test subject

After these operations, we will possess several prominent features used to score HRV (Figure 3.6.2). Among these, RMSSD, short for root mean square of successive differences between adjacent NN intervals, is considered the most relevant and accurate measure of HRV. Hence, it is one of the more heavily weighted features in our neural network model used to identify psychological stress, which will be discussed further in Section 3.8.

To prove that high psychological stress has a strong correlation with low HRV score, we put ourselves through an artificial stress test where the test subject is told to watch several compilations of scenes of jump scares. True enough, the results obtained agreed with our hypothesis as seen in the diagram below.




```
bpm: 103.616312
ibi: 579.059406
sdnn: 40.956919
sdsd: 38.900578
rmssd: 38.900578
pnn20: 0.268657
pnn50: 0.079602
hr_mad: 24.500000
breathingrate: 0.308770
```

Figure 3.6.3: HRV analysis for stressed test subject

Now that we are able to obtain HRV features from a window of PPG data, we are finally ready to move on to the next section, where our system will identify periods of negative psychological stress in users using an MLP model.

3.7. Identification of psychological stress

As explained in the previous section, psychological stress can have a negative impact on one's HRV score. Greater HRV at rest is generally indicative of better health, a younger biological age, and better aerobic fitness. Therefore, our IoT system seeks to improve one's

HRV score (which reduces the overall risk of SCA) by identifying periods where users experience high psychological stress and, thereby, notifying users to manage their stress accordingly. In essence, our system will act as a tool for preventive stress management.

Transfer learning is employed here to improve our system's ability to accurately perform the aforementioned tasks. Transfer learning is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different yet related problem. The intuition behind transfer learning is that if a model is trained on a large and general enough dataset, this model will effectively serve as a generic model that we can then take advantage of without having to start from scratch training a large model on a relatively small dataset we collect on our own.

We made very good use of the SWELL Knowledge Work Dataset (http://cs.ru.nl/~skoldijk/SWELL-KW/Dataset.html) for the purpose of transfer learning. The dataset was collected in an experiment, in which 25 people performed typical knowledge work (writing reports, making presentations, reading emails, searching for information). The experimenters manipulated the participants' working conditions with the stressors: email interruptions and time pressure. A varied set of data was recorded: computer logging, facial expression from camera recordings, body postures from a Kinect 3D sensor and heart rate (variability) and skin conductance from body sensors. The participants' subjective experience on task load, mental effort, emotion, and perceived stress was assessed with validated questionnaires as a ground truth.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 6)                 42
_________________________________________________________________
dense_2 (Dense)              (None, 100)               700
_________________________________________________________________
dense_3 (Dense)              (None, 100)               10100
_________________________________________________________________
dense_4 (Dense)              (None, 100)               10100
_________________________________________________________________
dense_5 (Dense)              (None, 100)               10100
_________________________________________________________________
dense_6 (Dense)              (None, 100)               10100
_________________________________________________________________
dense_7 (Dense)              (None, 3)                 303
=================================================================
Total params: 41,445
Trainable params: 41,445
Non-trainable params: 0
```

Figure 3.7.1. Structure of MLP model

Listed below is a summary of the procedures we have taken to train an MLP model based on the above mentioned dataset (refer to */iot_cloud/hrv/transfer learning/SWELL-Knowledge Worker Dataset for Stress and User Modeling Research.ipynb* for the source code).

1. Load the dataset into a Pandas Dataframe and extract only relevant features from the dataset (e.g. heart rate, mean R-R intervals, SDRR, pNN50, etc.)
2. Normalise the features using a MinMaxScaler(feature_range=(0,1)). Doing so improves the performance of the model.

3. Perform one-hot encoding on the labels (STRESS, INTERRUPTION, NO-STRESS)
4. Perform train-test-val split.
5. Build and train the MLP model using the train and validation set.
6. Evaluate the model using the test set.

Refer to Figure 3.7.1. for the structure of the MLP model. It comprises several dense layers with ReLU activation function and an output dense layer with softmax activation function.

The learning curves are also shown below, which are good evaluators of the overall learning done by the MLP model.

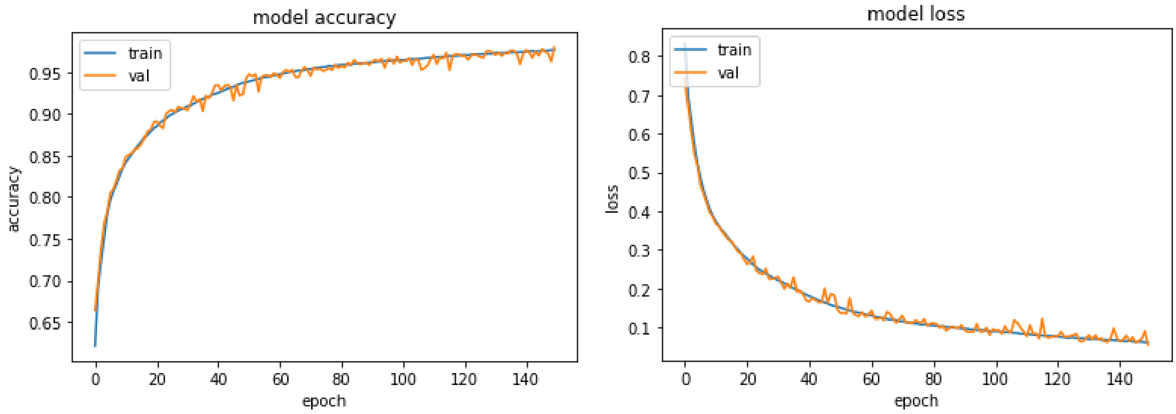


Figure 3.7.2. Learning curves

3.8. Human Activity Recognition (HAR)

HAR is a broad field of study concerned with classifying sequences of accelerometer data recorded by specialized harnesses or smartphones into known well-defined movements. These movements are often typical activities performed such as walking, standing, and sitting. In our case, we will be classifying 3 main categories of motions (and non-motions), namely IDLE, JUMPING, and RUNNING to showcase the HAR capabilities of our trained NN model. Performing HAR allows the system to monitor the periods where each individual user exercises and hence experiences positive physical stress.

A summary of the steps we have taken to train the CNN model using our own collected dataset is given below (refer to */iot_cloud/har/acc_train.ipynb* for the source code). The dataset comprises the x-, y-, and z-axis signals of the accelerometer data, as well as the label consisting of the ground truth of the motion involved.

Using the sliding window algorithm, we first segmented the dataset into appropriately-sized windows for analysis. In our case, each window size is 30 samples wide (3 seconds worth of data). Hence, the input shape of the ML model is (30, 3). We then perform train-test-validation split of the dataset.

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_1 (Conv1D)            (None, 23, 64)            640
_________________________________________________________________
dropout_5 (Dropout)          (None, 23, 64)            0
_________________________________________________________________
dense_8 (Dense)              (None, 23, 100)           6500
_________________________________________________________________
dropout_6 (Dropout)          (None, 23, 100)           0
_________________________________________________________________
dense_9 (Dense)              (None, 23, 100)           10100
_________________________________________________________________
flatten_3 (Flatten)          (None, 2300)              0
_________________________________________________________________
dense_10 (Dense)             (None, 3)                 6903
=================================================================
Total params: 24,143
Trainable params: 24,143
Non-trainable params: 0
```

Figure 3.8.1 Structure of CNN model

A CNN model is employed for this task. Refer to Figure 3.8.1. for the structure of the model. It comprises of a Conv1D input layer with ReLU activation function, several hidden dense and dropout layers, and an output dense layer with softmax activation function. The purpose of the dropout layers is to reduce the possibility of the occurrence of overfitting, which is extremely common in small datasets like the one we will be using for the purpose of HAR.

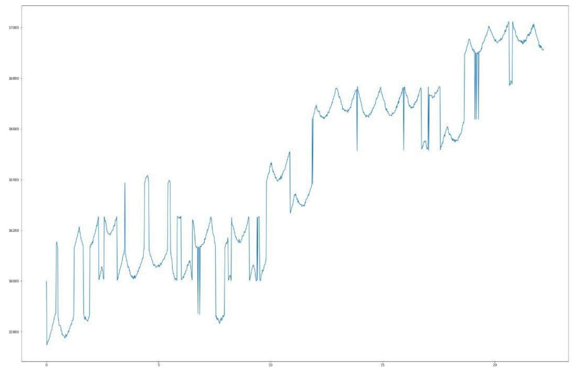
4. Experimental Evaluation

After processing signals and calculating heart rate using FFT, we were able to successfully detect SCA at the gateway device and sound the buzzer at the gateway device which would alert people nearby to assist the user suffering from SCA.

We were also able to accurately classify human activity into idle state, jumping state or running state, allowing us to only target heart rate data at idle state to eliminate false positives.

Lastly, with heart rate data at idle state, we were able to successfully predict when the user is under psychological stress. This experiment was carried out by letting the user watch jump scare videos and observing if our MLP model was able to pick up high psychological stress situations.

5. Challenges Faced



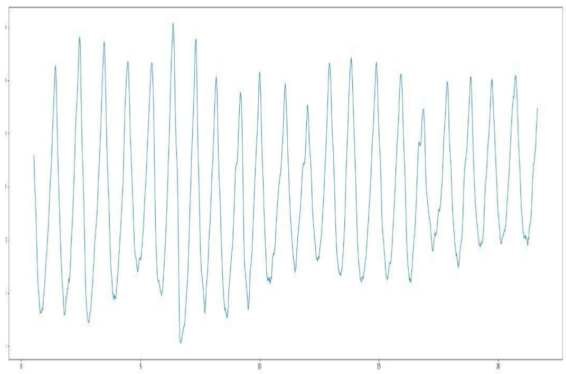


Figure 5.1: Signal before processing *Figure 5.2: Signal after processing*

One of the challenges that we faced would be the processing of PPG signals collected from the edge device. As shown in Figure 5.1, we could see that raw signal data contains a lot of noise making it very difficult for us to calculate the hear trate of the user accurately. Our solution to this was to use signal processing (refer to Section 3.3) to reduce noise collected in our PPG signals. Figure 5.2 shows how the PPG signals look like after signal processing.

Another challenge that we faced would be the lack of PPG data available online for training our model. Although there was a significant amount of ECG training data, we were not able to find PPG data large enough for training our model. Our solution to this was to supplement training data available with data that we have collected ourselves.

Our original intention was to implement FFT calculation at the edge device so as to transmit heart rate data only instead of raw PPG signals via BLE. However, after implementing FFT at the edge device, we realize that it cannot run correctly. The reason is that the jump noise elimination logic does not work correctly at the edge device (might have something to do with how the data is stored and encoded in the edge device) and that the implementation of the circular buffers used for FFT calculation seems to be too intensive for the edge device. Our solution to this problem was to move the FFT calculation from the edge device to the gateway device where the sensor data can be read properly, and replace all circular buffer implementation into a first-in-first-out queues.

6. Future Extension

A possible extension for our application would be to integrate the cardiac alert functionality with SCDF's myResponder mobile application. As our current implementation is already collecting the user's current coordinates in our application, should a cardiac arrest be detected in our application, we can pipe this information to the myResponder application so that any responder available nearby can assist the user and relevant medical personnel can be alerted as well.

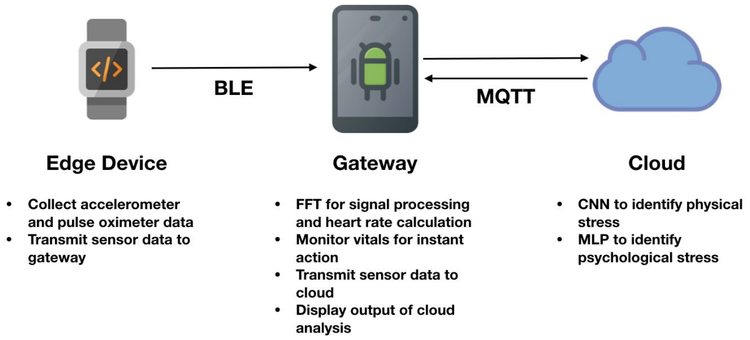
7. References

American Health Association. (2014, June 17). Stress and Heart Health. Retrieved November 16, 2019, from https://www.heart.org/en/healthy-living/healthy-lifestyle/stress-management/stress-and-heart-health.

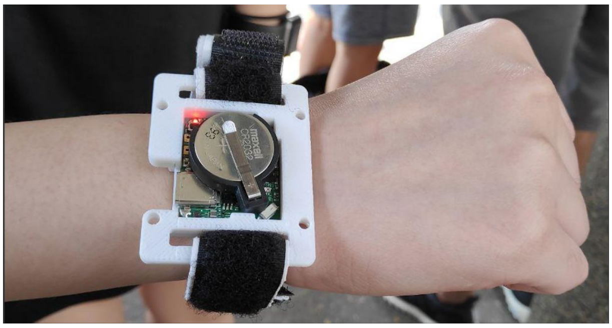
Recommended Configurations and Operating Profiles for MAX30101/MAX30102 EV Kits. (2018). Retrieved November 16, 2019, from https://pdfserv.maximintegrated.com/en/an/AN6409.pdf.

Sudden Cardiac Arrest. (n.d.). Retrieved November 5, 2019, from https://www.myheart.org.sg/my-heart/heart-conditions/sudden-cardiac-arrest/.

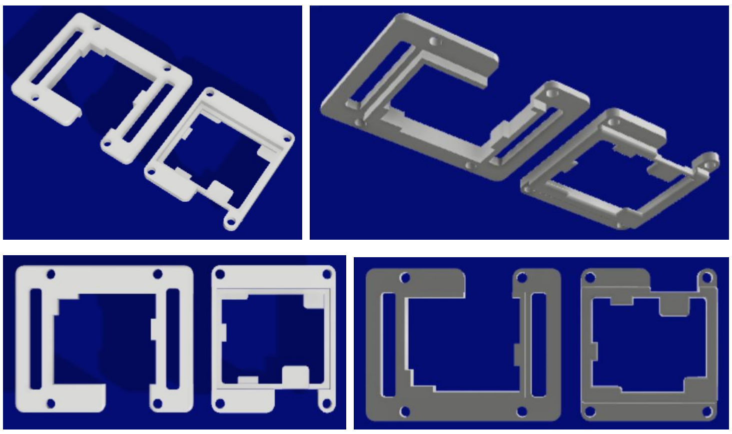
Appendix A: Overall Operation of IoT Healthcare System



Appendix B: Wrist Worn Device



Appendix C: 3D Model for Watch Case



Appendix D: GitHub Repository of Project

[Link to GitHub Repository]