Smart Team Final Report

Project #4 - Heartbeat-based authentication in wearable devices

Nathan Driscoll, Nicholas Stevens, Solomon Chong

Topic Definition

During the COVID-19 pandemic, quarantining of possible COVID-19 cases (e.g. travellers from overseas countries which have a high number of COVID-19 cases) has been used to prevent the spread of the virus. For example in South Korea, all COVID-19 patients must undergo 10 days of compulsory hotel quarantine [10]. Home quarantine is an important component of this and will be the focus of this project. However, up to 25% of people in home quarantine violate home quarantine rules [11]. To ensure people stay at home during their home quarantine and as an alternative to door knocking by authorities, wearable devices can be used. A wearable device can be worn by the person in home quarantine that can track their location and alert the relevant authorities if a person has broken home quarantine rules. But the problem with this is, how do you know who is wearing the device?

The solution to this problem provided by this project is heartbeat-based authentication.
Our team will aim to develop a heartbeat based authentication, which is to be incorporated inside a wearable device. This project will be based on the paper “Unlock with Your Heart: Heartbeat-based Authentication on Commercial Mobile Phones” [6]. The wearable device will be worn by people who are in home quarantine. For this project, the wearable device will be a sensortag.

Background

Remote Electronic Heart rate monitoring has quite a history over the last 100 years. Although taking a pulse as a diagnostic tool has always been part of medicine, the first device that was marketed wireless was able to electronically measure the heartbeat as a continuous function from electrical signals was marketed in 1977 using a known technology EKG (Electrocardiography) [1] and making it portable. With the EKG, study of the heartbeat was able to move past beats per minute (BPM) and pressure into complex time and frequency analysis [2]. Since then, several ways to measure the heart have been developed; the focus has diverged between the use in specialised machines for modern medicine and portable devices that provide quick insights into health.

Popular for portable devices is the use of optical light to measure skin capillaries which allows low contact measurement. There are also several attempts co opting hardware developed for Wi-Fi [3]and smartphone accelerometers both with success. With this increased accuracy of portable devices experimentation in heart monitoring in use for biometrics-based authentication has been explored.

The use of accelerometers to measure the vibration of your chest in response to a heartbeat is called a seismocardiogram (SCG) [6]. This vibration can be used as a biometric feature for user

authentication. This is because the heartbeat pattern depends on the biological features and geometric structure of the heart, which is unique for each person.This means that SCG provides strong protection against spoofing attacks.

Heart rate provides unique advantages over other biometric security such as fingerprints in that it is much harder to fake. Devices to mimic a heart while possible are not widely available in consumer space [4]. Early attempts with authentication using large EKG machines showed a false acceptance ratio of 2% and false rejection of 0.3% possible [5].

The paper “Unlock with Your Heart: Heartbeat-based Authentication on Commercial Mobile Phones” [6] which our project will be derived from uses the accelerometer in smartphones to measure the heart and works to uniquely identify an individual with 95% accuracy (from a sample of 35 people).

The paper discusses the changes of finding unique features of a person's heartbeat to differentiate from others while accounting for normal variance in an individual. The use of Dynamic time warping and time domain matching to normalise a signal then using the machine learning technique Support Vector Machine to compare an input to a user against a global benchmark is discussed in depth.

Commercial applications for heart-based authentication are already on the market. In 2014 a small start-up tried to market a device that could be worn as a small ring and provide authentication to other devices such as a phone based on the wearer’s heartbeat [7]. Nymi’s current device is wristband based and requires a fingerprint to be activated once upon wearing it. The current solution is aimed at high security environments, in essence a key card with extra security. This is much different to the consumer focus the company started with but shows that there is a market space for heart rate security. [8]

Other papers show that improvements on the 95% with a smartphone to 99% [9] as well as the practical implementation of cloud processes to register a user and device to a system.

Project Outline

In this project we will create a heartbeat-based authentication system that will be used in a wearable device. The system will attempt to accurately authenticate a person based on their heartbeat pattern.

To authenticate a person, their heartbeat pattern will be collected by the sensortag using the accelerometer on the tag. The collected pattern will then be analysed by a support vector machine (SVM) model to compare it against the person’s heartbeat pattern as well as other heartbeats. The SVM model will give the likelihood that the given heartbeat belongs to the owner or not and if this likelihood is above a certain threshold, then the person is successfully authenticated.

For the project, there are 2 use cases of the system: training and authentication.

Training

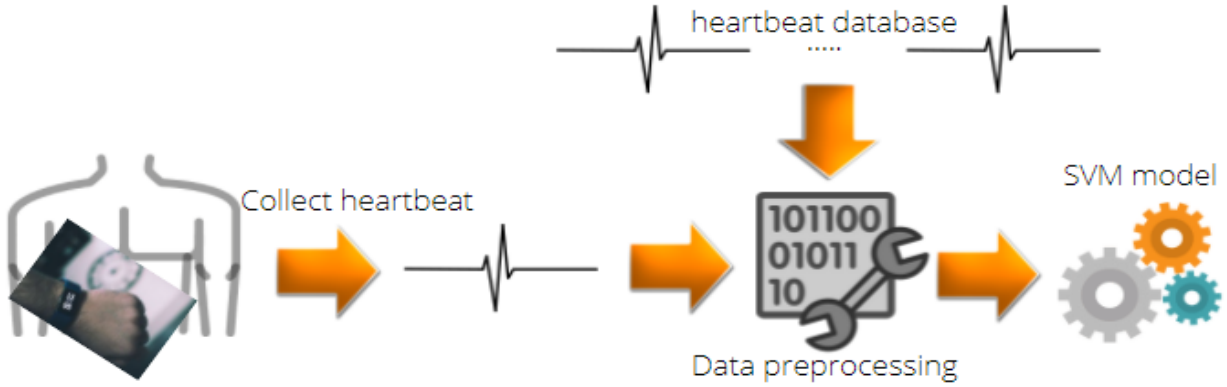


Figure 1 - Training

The first step is the training process. This step involves collecting the person's heartbeat pattern using the accelerometer on the sensortag. The heartbeat pattern collected is the training heartbeats. This heartbeat pattern then undergoes steps called heartbeat segmentation and feature extraction. These steps reduce the noise of the heartbeat pattern. Once the heartbeat pattern is collected and features are extracted,they are used in a support vector machine (SVM) model. The SVM model is a 2-class classifier that is trained using the training heartbeats from the owner (as the positive samples) and the benchmark heartbeats from a heartbeat database (as the negative samples) [6].

Authentication

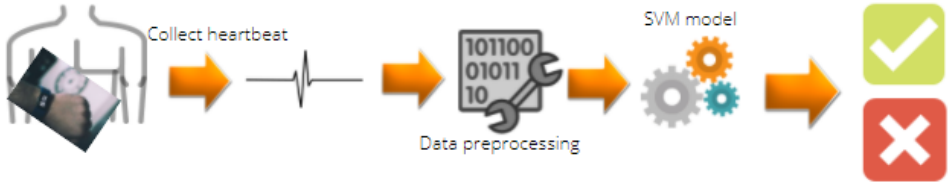


Figure 2 - Authentication

The second step is the authentication step. A person will wear the device and samples of their heartbeat will be taken just as in training. These heartbeats will have features extracted and then be inputted into the SVM model. The SVM model will then give the likelihood of whether an unknown heartbeat signal belongs to the owner or not. If this prediction is above a certain threshold (to be determined during testing), then the person is successfully authenticated.

Implementation

We have divided the project into 2 main implementation stages - collecting and training. Collecting refers to using the sensortag to collect heartbeat patterns. Training refers to the training of the support vector machine (SVM) with processed data. Each stage corresponds to what we believe to be a major part of the project.

In the collecting stage we attempt to collect heartbeat data from the sensor tag. We will be using the accelerometer on the sensortag. As we are using onboard sensors to measure the heartbeat, we estimate that this can be done within a week with knowledge gained from lab exercises.

We estimate that the training stage will take up the most amount of time. As a lot of preprocessing needs to be done to the collected data in order to start training our SVM model.

Milestones:

- **Collection** - capture the heartbeat motion of the user using the accelerometer on the sensor tag to collect the data needed for training as well as authentication
- **Segmentation** - divide the continuous heartbeat patterns into individual heartbeats. This step is required for feature extraction.
- **Feature Extraction** - each heartbeat cycle is decomposed into multiple levels of wavelet coefficients, and we choose the wavelet coefficients that are most closely related to the heartbeat patterns. This way, we reduce noises that come from different sources, including the respiration movements, small limb movements, and small variations in accelerometer readings [6].
- **Training the SVM model** - the model must be trained and set up correctly to ensure correct authentication. The SVM will be fine tuned to maximise its accuracy
- **Testing the SVM model** - test how well the model performs. Testing of the model must be conducted to evaluate its effectiveness. Testing and training will happen simultaneously to maximise the accuracy of the model.

The following gantt chart for the project has been developed showing the timeline for the implementation of each milestone.

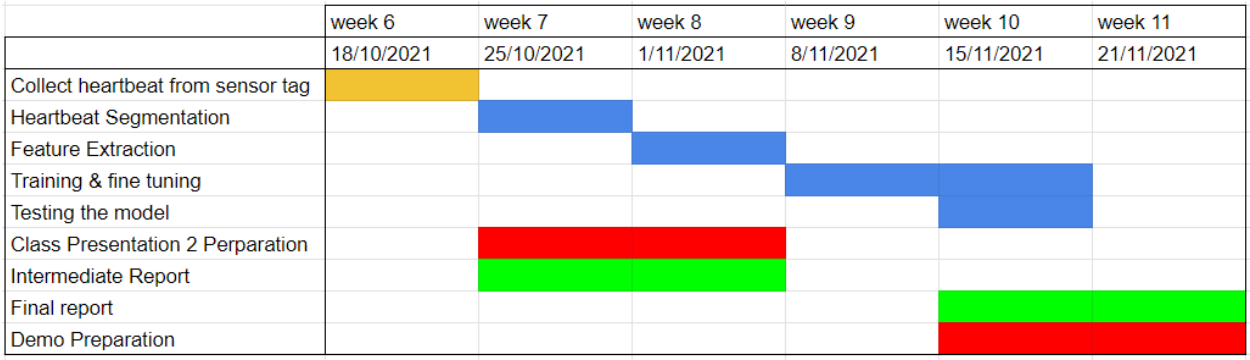


Figure 3 - Gantt Chart

Collection

With the sensor tag we have been able to collect a 7 sec sample at 250hz of the z-axis of a heart beat and transfer the data to a computer to process with python. There were some changes required to the MPU-9250 contiki driver as the device casted the reading from unsigned integer to a floating point back to a signed int losing precision as the least significant bit is quite important for our use.

The sample collected was precise enough to detect heartbeat with enough precision which we believe to do the heartbeat authenticatication. Details of what we have done with the current data is explained in the segmentation section.

The project required taking readings at 250Hz. This required the use of rtimers, as c,s and e timers only give at most 100Hz sampling. After implementation of an r timer the contiki was able to meet the objective of 250 Hz. Unbeknownst to us at the time the use of rimter catastrophic affected the RF components of contiki. 6Lopan stopped working. It was found that shutting the RF off then on restored the 6Lopan Network.

Networking

Physical and Network layer

We had originally planned on using Bluetooth Low Energy (BLE) to connect the sensor to a computer to process the data. BLE has an advantage in that many computers have BLE in built, not requiring the use of additional hardware such as an edge router. While the original contiki version does not provide BLE for the sensor tag, a maintained fork called contiki-ng does. After some code changes we were able to build and flash our project using the new contiki. We were not able to get BLE working. At this point we had three option

- Convert the logic of our code to Texas Instruments RTOS (TI-RTOS) which had functioning BLE drivers
- Continue debugging the BLE on the contiki system.
- Move to the 6Lowpan, which has working drivers.

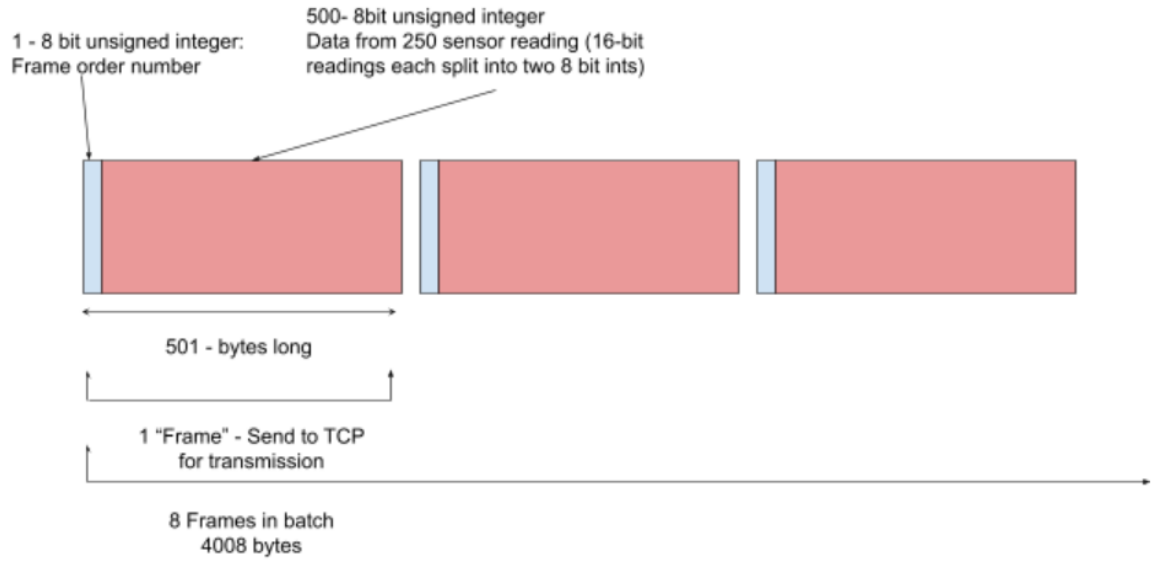
We spent time on all three options. When no progress was made with BLE we moved to using the 6lowpan.

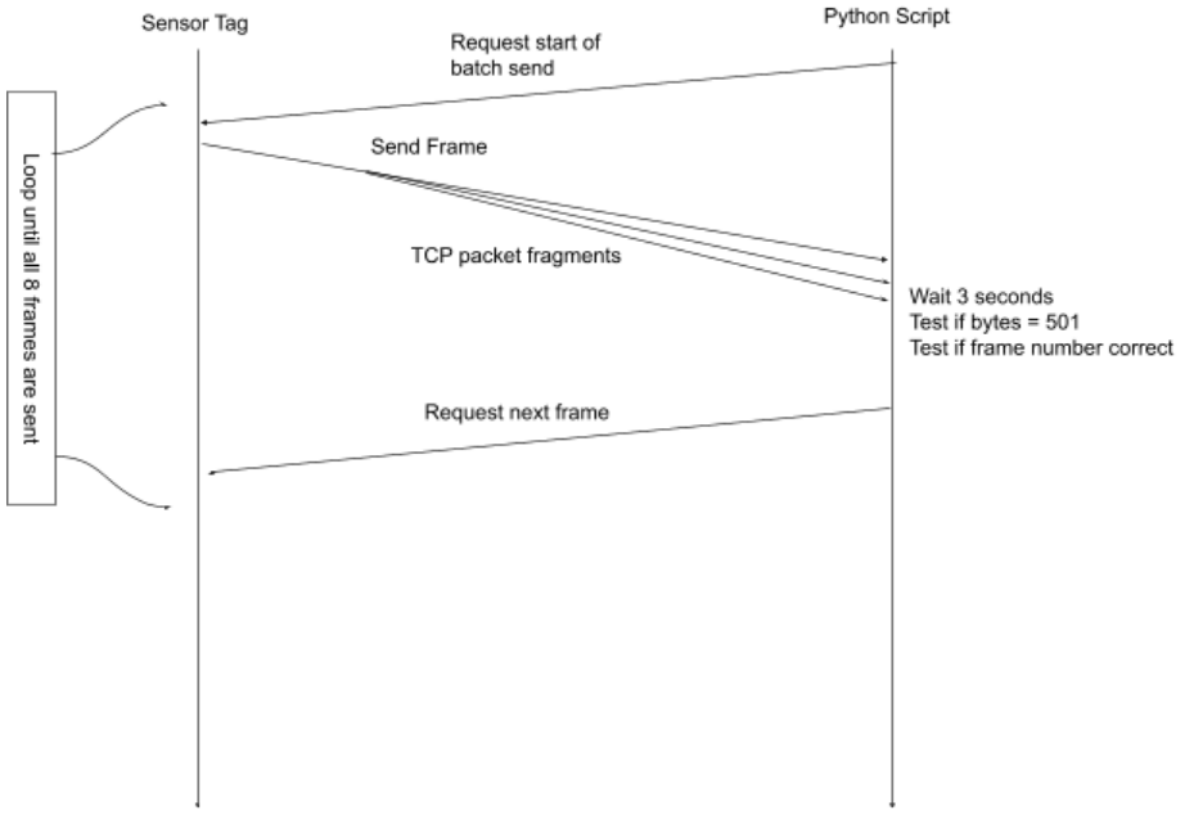
Application Layer

The project uses the TCP socket library in contiki to send a fixed length frame to any client on the network that requests it. A simple python script using the socket library accepts the frame and does testing to ensure that the expected amount of data has been sent and that the frames are in the correct order.

The structure of the Application layer is:

- The 2000 reading 16-bit readings are split into 4000 8-bit reading
- The 4000 bytes are broken into 8 groups of 500 bytes; these are referred to as frames.
- Each 500 bytes are appended with a single 8bit unsigned number which corresponds to the order of the readings Frame 1 was taken before frame 2.
- Frames are only processed one at a time after a request has been made by a client.
- When the contiki sensor receives a packet with the first for bytes ‘send’ it will process the first frame and send to the client that sent the ‘send’ bytes.
- The client waits for a 3 second delay; checks that 501 bytes have been received; checks that the first bytes corresponds to the expected frame.
- The client will send a new packet back to the with ‘send ‘ as the payload to receive the next frame.
- When all 8 frames have been received by the client, the client converts the 4000 bytes back into 2000 readings and outputs to a csv with a name inputted by the user.





Networking Problems encountered:

1. Not true TCP - contiki does not check the acknowledgments sent by the requesting client against the expected packet. When TCP segmentation happens contiki uses the acknowledgment for a packet that it has not yet been sent and does not end up sending the next packet at all. The exact cause of this was not found. The solution was for the client receiving the data to not ask for anymore packets until the expected amount of bytes have arrived.
2. The heart reading stops the RF functionality. The work around was to restart the RF portion after a heart reading. The assumption is that the use of rtimer causes timing issues in the RF code but the cause was not investigated.

Heartbeat segmentation

As the sensor tag collects heartbeat in a continuous manner, we have to separate it into individual heartbeats. Since an average person has a heartbeat ranging from 50bpm to 120bpm, we divide the collected heartbeat into sections of 2 seconds to guarantee that at least 1 full heartbeat cycle can be collected in each section. Figure 5 below shows a heartbeat pattern being separated into 2 second intervals.

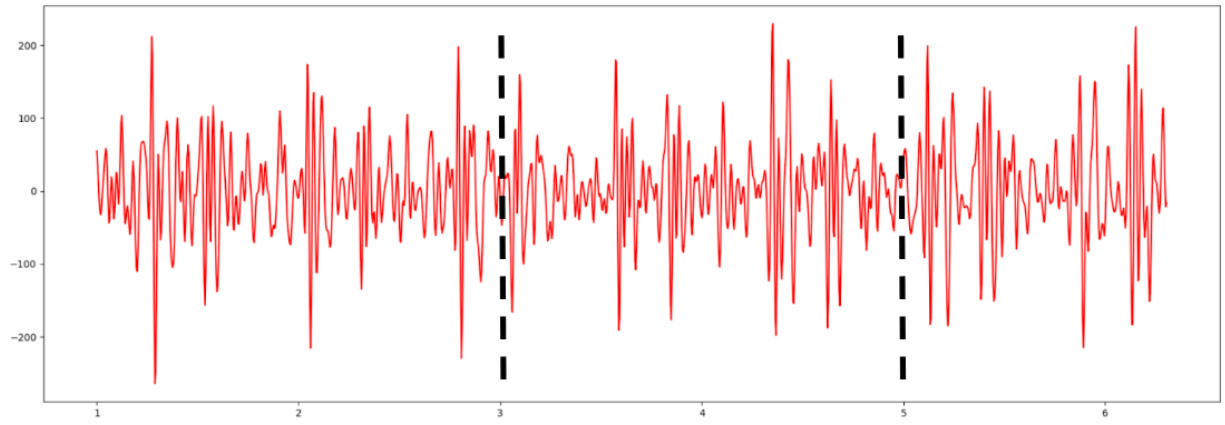


Figure 5 - Segmentation

We then try to segmentate out a heartbeat in a section. The heartbeat cycle contains features such as the AO and RF. These features can be found to correctly segment the heartbeat pattern. In each 2 second interval - we will extract a single heartbeat. We do this by identifying the AO and RF using the shortest distance between 2 peaks that's greater than 200ms starting from the highest peak. This performs well in trying to find the AO and RF as AO to RF intervals are larger than 200ms when the heart rate is slower than 120bpm. Now that we have obtained the AO to RF distance, we simply multiply 0.5 to it to obtain the ATC to AO distance and from that we can obtain a full heartbeat cycle.

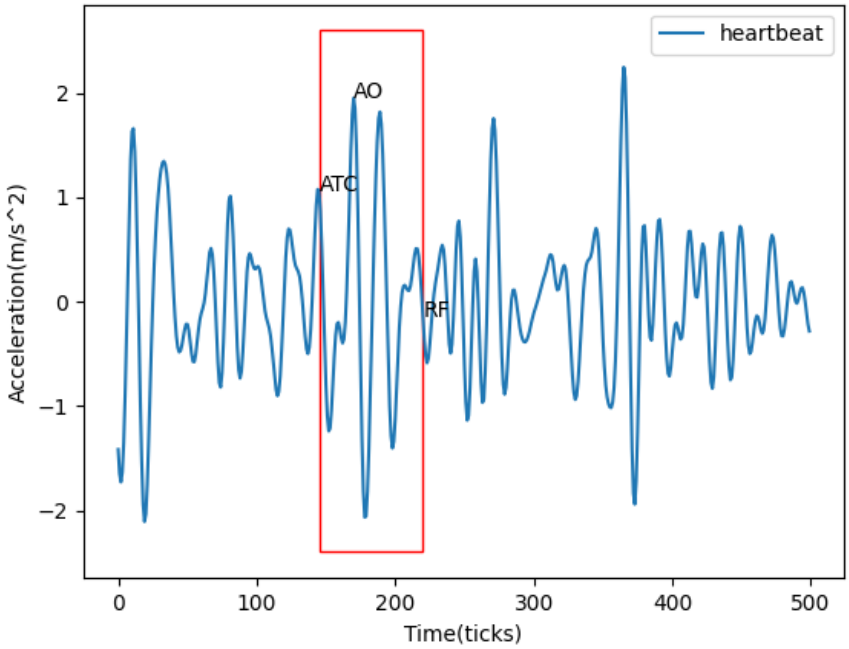


Figure 6 - Segmented heartbeat (the ATC-RF interval forms a heartbeat

We are confident that the data should perform well with the SVM as the segmented section resembles a full heartbeat fairly accurately as shown in the graph below. (left - heartbeat from the paper, right - our segmented heartbeat)

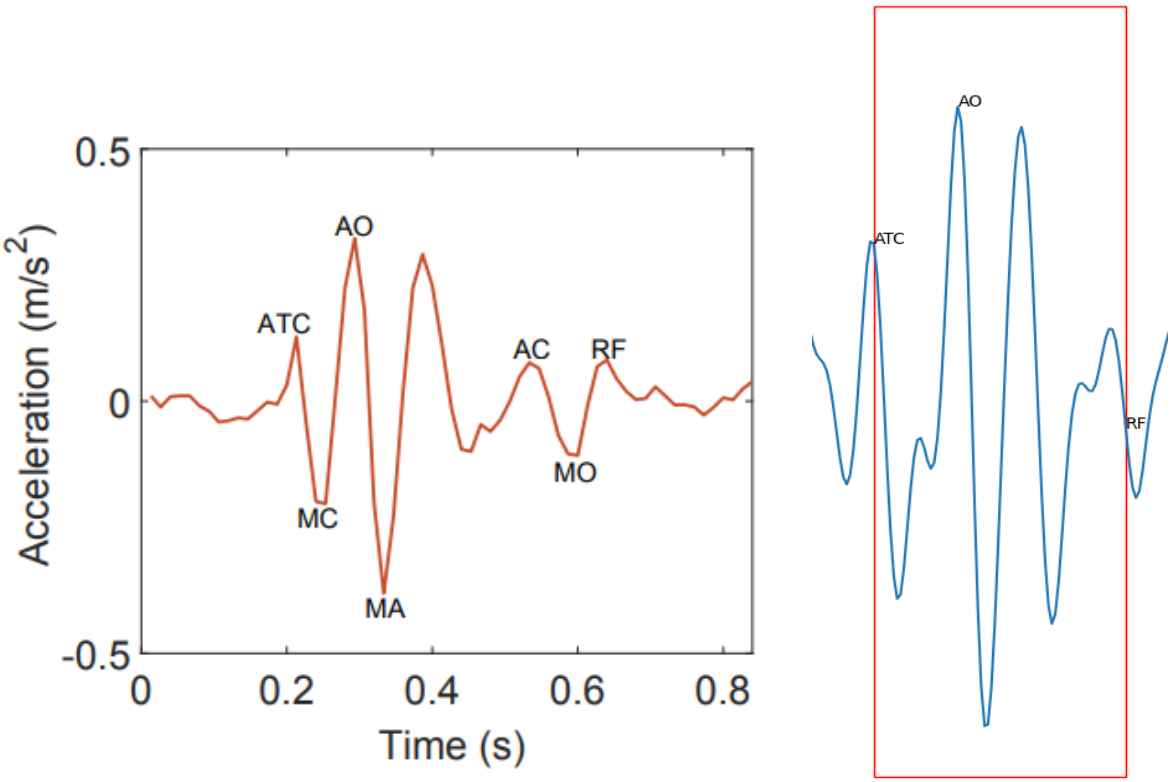


Figure 7 - Heart comparison

Feature extraction

After heartbeat segmentation, the next task was to convert the segmented heartbeat into features appropriate to be used in training the SVM. The feature extraction process needs to retain the characteristics of the user's heartbeats and remove irrelevant noises.

The paper outlined numerous methods of feature extraction. Our team tried variations of these methods. The first method that was implemented was taking the average amplitude of all heartbeats during the different stages of a heartbeat cycle. Namely the average ATC, average AO and average RF. The resulting SVM was not performing as well as expected. (see evaluation for more details)

Clearly traditional methods of extracting features based on the interval between different heartbeat stages is not applicable in our situation. This is due to the variations in the amplitude of SCG leading to unreliable heartbeat stage identifications. Therefore, we switched to using Discrete Wavelet Transform (DWT) to extract features from the SCG signals.

The DWT decomposes SCG signals into five levels with level 1 to 5 represent signal components in the frequency range of 25 ~ 50 Hz, 12.5 ~ 25 Hz, 6.25 ~ 12.5 Hz, 3.13 ~ 6.25 Hz and 1.56 ~ 3.13 Hz, respectively. To reduce noise in the SCG signal, we only use the detailed coefficients from the second level to the fourth level as the feature vector for heartbeat authentication.

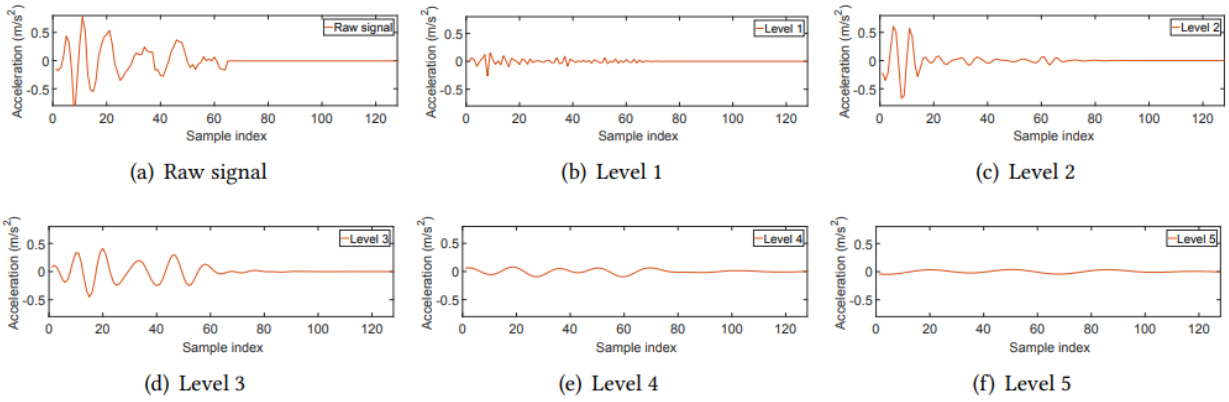


Figure 8 - Five levels of DWT decompositions

From here, we tried using the average of the detailed coefficients across those three levels but resulted in an even lower accuracy than the first method. It came to the team's attention that more features are required to improve the model. Hence, we ended up using all values across those three levels to obtain a 56 dimensional feature vector and finally we see a slight improvement in the accuracy. (see evaluation for more details)

	feature0	feature1	feature2	feature3	feature4	feature5	feature6	feature7	feature8	...	feature47	feature48	feature49	feature50	feature51	feature52	feature53	feature54	feature55
0	0.000095	0.000375	0.000448	0.000239	-0.000943	-0.000397	-0.001005	0.000994	-0.000465	...	-0.000289	0.001554	-0.002450	0.003799	-0.007877	0.015073	-0.011075	-0.000753	-0.000622
1	-0.000607	0.000481	-0.000014	-0.000042	-0.000196	-0.000187	-0.000242	0.000727	-0.000166	...	-0.001396	0.000836	-0.001620	0.001495	-0.000598	0.000789	-0.020161	0.030542	-0.014000
2	0.000445	0.001325	-0.000401	0.000102	-0.001352	-0.000233	-0.001019	0.001529	-0.000243	...	-0.001061	0.002134	-0.006691	0.009895	-0.020644	0.040661	-0.029300	0.000101	-0.002564
3	0.000071	-0.000174	0.000166	-0.000506	-0.000404	-0.000140	0.000599	0.001019	0.000811	...	0.000497	-0.000065	0.000698	-0.000222	0.000699	-0.000326	-0.000120	-0.000373	-0.001038
4	0.000646	-0.000736	0.000551	-0.000316	0.000070	-0.000280	-0.000110	0.000342	-0.000266	...	0.001745	-0.001414	0.000792	-0.000491	-0.000832	0.000826	-0.001429	0.001365	-0.001661
5	0.000327	-0.000668	0.001214	-0.000363	0.001115	-0.000617	0.000332	-0.000284	0.001181	...	-0.000789	0.000439	-0.000466	0.000864	0.000002	-0.000888	0.000611	-0.000866	0.002597
6	0.000146	-0.000046	-0.000041	-0.000345	0.000036	0.000079	-0.000059	-0.000364	0.001684	...	-0.001044	0.002160	-0.002499	0.004447	-0.009308	0.005821	0.000226	-0.000015	0.000277
7	0.000723	-0.000673	0.000496	-0.001585	-0.000037	-0.001651	0.000819	0.000142	0.001157	...	-0.000631	0.000659	0.000171	0.000617	0.000200	-0.000211	0.000119	-0.000354	-0.000578
8	-0.002564	-0.000186	-0.001516	0.002026	0.000770	0.002016	0.000116	-0.000696	0.000748	...	0.000278	-0.000041	0.000438	-0.000705	0.000716	-0.000940	0.000874	-0.000506	0.000320
9	0.000321	-0.000149	0.000299	-0.000347	-0.001240	-0.000534	-0.000725	0.000720	-0.002067	...	0.000059	0.000578	-0.000842	0.002329	-0.003714	0.008871	-0.005578	0.000654	-0.000169
10	-0.001049	0.000049	0.000199	0.000583	0.000568	0.000105	-0.000176	0.000215	-0.002344	...	0.000184	-0.000086	0.000182	-0.000652	-0.000119	-0.000843	-0.000205	-0.000274	-0.000074
11	-0.000417	0.000211	0.000206	-0.000477	0.000165	-0.000333	0.000566	0.000275	0.000334	...	-0.000001	-0.000494	0.001012	-0.000147	0.000739	-0.001479	-0.000401	-0.000986	0.000205
0	-0.041742	0.019167	0.007863	-0.026303	0.063219	-0.109564	0.280887	-0.013170	-0.059211	...	0.080198	-0.137461	0.110905	-0.043037	0.038602	-0.034982	0.027481	-0.015947	-0.003860

[13 rows x 56 columns]

Figure 9 - Feature vector

SVM training and testing

We used the sklearn implementation of the SVM. The SVM was created with the kernel parameter "linear" and all other parameters default. Each 56 dimensional vector created from each heartbeat formed a row inside a Pandas dataframe. For both training and authentication, all feature columns were passed to the SVM as data and the last column called "target" was passed as the label column.

Our team collected many heartbeat patterns to be used as a heartbeat database when training the SVM. These heartbeats form the negative case and allow the SVM something to compare the given heartbeat sample against to make a prediction. When training, the users inputs are

concatenated with the heartbeat database to form a single Pandas dataframe that can be inserted into the SVM for training.
For authentication, the user passes in a new heartbeat pattern and this data is passed to the SVM predict attribute which will return a value of 0 (meaning this is the wrong person) or 1 (meaning this is the correct person and they are successfully authenticated).

Evaluation

5-fold cross validation was used to evaluate the accuracy of the SVM model on each of the feature extraction methods. The method cross_val_score was used from sklearn to calculate the 5-fold cross validation score.

The accuracies calculated are shown in the table below:

Method	Accuracy
1. Average AO, RF, ATC scores	71%
2. Average DWT values across 3 levels	65%
3. Full DWT values across 3 levels	76%

The highest accuracy was calculated for 76% for option 3 - using the full array of DWT values across 3 levels and therefore this method was chosen as the final method. With an accuracy of 76% this system needs further improvements before it can be used for authentication as this accuracy will not protect the system against attackers and will also deny legitimate users at a rate too high. An accuracy closer to 95% would be appropriate before this system can be used for authentication as attackers would be denied and legitimate users blocked

Reasons for low accuracy:

- **Not enough data** - due to university being online and the group being unable to meet up in person due to the COVID-19 pandemic, the collection of heartbeats was difficult. With more heartbeat data, the SVM will have more training data which will enable it to make a correct prediction more of the time - increasing the accuracy of the system.
- **Inconsistent data** - sensortag can give poor readings at times. These poor readings which the SVM uses to train or predict makes it harder for the SVM to create a correct decision boundary leading to a decrease in accuracy. A different device with a better accelerometer may be able to give more consistent data.
- **More SVM fine tuning required** - To increase the accuracy of the SVM more time was required in fine tuning. There was not enough time to check the effect that different parameters of the SVM had on the accuracy of the SVM. Also, while the method of using all values from 3 levels of the DWT was chosen in the end, there may have been another method that gave even higher SVM accuracy but unfortunately there was not enough time to investigate other feature extraction methods.

Conclusion

In this report we have demonstrated our implementation of using heartbeat patterns for authentication. Our results have demonstrated that heartbeat based authentication is possible and with some improvements to our system such as more heartbeat data and a more fine tuned SVM model,, we believe that heartbeat based authentication can be used for any authentication applications such as tracking and authenticating that a person remains in home quarantine during the COVID-19 pandemic.

Bibliography

[1] G. Ernst, "Hidden Signals—The History and Methods of Heart Rate Variability," *Frontiers Public Health,* vol. 5, no. 265, 2017.

[2] "Standards of Measurement, Physiological Interpretation, and Clinical Use," *Task Force of the European Society of Cardiology the North American Society of Pacing Electrophysiology,* vol. 96, no. 5, 1996.

[3] X. Z. Z. L. a. F. R. Yu Gu, "WiFi-based Real-time Breathing and Heart Rate Monitoring during Sleep," in *IEEE Global Communications Conference*, Waikoloa, 2019.

[4] M. H. K. C. D.A. Ramli, "Development of Heartbeat Detection Kit for Biometric Authentication System," *Procedia Computer Science,* vol. 96, pp. 305-314, 2016.

[5] C. Hegde, H. R. Prabhu and D. S. e. a. Sagar, "Heartbeat biometrics for human authentication," *Signal, Image and Video Processing,* vol. 5, no. 4, p. 485, 2010.

[6] L. Wang, K. Huang, K. Sun, C. Tian, L. Xie and Q. Gu, "Unlock with Your Heart: Heartbeat-based Authentication on Commercial Mobile Phones," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies,* vol. 2, no. 3, 2018.

[7] C. Yury, "Your Heartbeat May Soon Be Your Only Password (Interview with Andrew D'Souza, President of Bionym)," Wired, June 2014. [Online]. Available: https://www.wired.com/insights/2014/06/heartbeat-may-soon-password/. [Accessed 10 October 2021].

[8] Nymi, "Nymi Workplace Wearables," Nymi, [Online]. Available: https://www.nymi.com/nymi-band. [Accessed 10 October 2021].

[9] S. Islam, "Heartbeat Biometrics for Remote Authentication Using Sensor Embedded Computing Devices," *International Journal of Distributed Sensor Networks,* vol. 11, no. 6, 2015.

[10] Https://www.abc.net.au/news/2021-10-10/south-korea-mandatory-quarantine-for-the-covid-positive/100514744

[11]https://www.sbs.com.au/news/simply-unacceptable-one-in-four-victorians-with-coronavirus-not-at-home-when-door-knocked/ced16679-fadc-4121-95e4-28e7848ec09a