Watch the InFLUence: A Flu Diagnostic Model

Avik Banerjee, *Student, Data Modeling and Pi*, Ivan Madrigal, *Student, Sensors and Hardware*, Quyen B. To, *Student, User Interface* and Rahim Esfandyarpour, *Professor, Advisor*

***Abstract*—The influenza is a respiratory disease that warps the world when its annual outbreak comes around. Containment has become essential to reduce the intensity of each outbreak and thus a need for early and rapid flu detection. The goal of *Watch the InFLUence* is to analyze and compare multiple body parameters using our data model to determine whether a subject is infected or is healthy. Although we experienced many limitations due to the Collaborative Institutional Training Initiative standards and the recent coronavirus, we were limited on data, but through online sources we developed a rudimentary model. This model was developed by gathering data from volunteers on the UCI campus and using data found online. To diagnose a subject, we would take their data and send this to our model and user interface as .CSV files. Afterwards, the model compares the data and produces a diagnosis. Meanwhile, the user interface presents the plots for ECG and PPG while displaying the numerical values for body temperature and heart rate. The data is locked behind a user specific code ensuring that each subject's data is private. Together, this forms our flu diagnostic model that is capable of flagging participants based on nuances in the four body parameters that were listed above.**

***Index Terms*—Body Temperature, ECG, Electrocardiogram, Heart Rate, Influenza, Photoplethysmography, PPG, User Interface**

I. Introduction

INFLUENZA, also known as the flu, is a respiratory disease that has large-scale outbreaks every year that typically occurs in the winter, known as flu season. These outbreaks occur due to the constantly evolving strain of the virus which makes developing resistance difficult. The influenza has a known communicable period spanning approximately 6 to 8 days from first contracting symptoms and travels through the air to people around the carrier [1]. With approximately one billion people infected and up to 500,000 casualties per year [2], the influenza virus has proven to be a topic that requires early and accurate diagnosis. As such, our goal is to design a model that can detect the influenza based on nuances within the subject's body. These nuances begin arising within a few hours of contracting the virus and thus can be detected with an accurate enough model. Having the ability to detect the flu early will enable medical professionals to give advice to the subjects such as prescribing them the appropriate treatments. As a result of early detection, we believe that this would help reduce traffic in medical facilities during flu season as carriers can be identified and treated early on before they infect too many other people.

Presently, there are numerous methods to diagnose the influenza virus; however, these methods require specialized equipment and lap space, intimate methods or large periods of time. The two premier methods are rapid influenza diagnostic tests (RIDTs) and rapid molecular assays (RMAs)[3]. These two approaches can provide a result within a 10-20-minute timeframe, but they are limited by three factors: they require non-reusable cotton swabs, have limited reliability and require intimate methods. Originally our goal was to create a wearable device alongside the model that can measure certain body parameters and diagnose the flu, but due to limited time and lack of large sample sizes we reduced our scope. Consequently, by focusing on the model we deviated from RIDTs and RMAs invasive methods and instead used simple contact methods. These simple contact methods enabled us to maintain a small timeframe of data collection of approximately 2 minutes. Compared to our original estimated timeframe of 8 minutes, we reduced our timeframe by 75%. As a result, our scope change made us much more time effective than before at the cost of real time diagnosis.

Originally, our model was designed to diagnose based on five different parameters: however, due to extremely limited available data on EDA and the difficulty of accurately getting a respiration rate from our sensors. Similarly, we opted to avoid using respiration rate due to the difficulty we had accurately analyzing this parameter, thus we swapped to PPG. We made this change because its envelope is respiration rate, thus removing the error by having the respiration rate directly. As a result, we now analyze four body parameters that are most likely to change when contracting the influenza. These parameters are body temperature (BT), heart rate (HR), electrocardiogram (ECG), photoplethysmography (PPG). By comparing the subject's values to our modeled data, we can detect subtle fluctuations attributed to the influenza and thus advise our subjects to seek medical advice and limit the amount of time they spend around others.

Our approach to designing an influenza diagnostic model was to collect samples from students on campus and

This paper was submitted on Dec. 11, 2019 for review. This work was supported in part by the Undergraduate Research Opportunity Program.

A. Banerjee is an undergraduate at the Department of Electrical Engineering and Computer Science, University of California, Irvine, Irvine, CA 92697 (email: avikb@uci.edu).

I. Madrigal is an undergraduate at the Department of Electrical Engineering and Computer Science, University of California, Irvine, Irvine, CA 92697 (email: imadriga@uci.edu).

Q. B. To is an undergraduate at the Department of Electrical Engineering and Computer Science, University of California, Irvine, Irvine, CA 92697 (email: qbto@uci.edu).

R. Esfandyarpour is an assistant professor for the Departments of Electrical Engineering and Computer Science and Biomedical Engineering at the University of California, Irvine, Irvine, CA 92697 (email: rahimes@uci.edu).

compounding it with data found from the internet. Unfortunately, due to the coronavirus and inability to gather data from clinics or hospitals without prior HIPAA approval, we were limited with the amount of data we could gather. Consequently, we could only produce a rudimentary model that can detect nuances but is not at the current standards for influenza detection accuracy.

II. Materials Used

To develop our influenza diagnostic model, we used both hardware and software that work in unison. As such, we have an extensive comprised of continuous measuring hardware, one-shot hardware and software. Alongside our hardware, we also used some specific sanitary materials and logistical documentation that ensure that we are abiding by Collaborative Institutional Training Initiative (CITI) regulations.

A. Hardware

All hardware involved within the project can work independently of the other, meaning they only require a power source or a laptop to function. As such, to distinguish our hardware, we decided to design a hierarchy amongst the hardware to establish their level of significance and reliability. The hierarchy we constructed involves continuous measuring hardware and one-shot hardware. We chose these as our distinguishing characteristics because we get more data from continuous hardware, but are limited by our one-shot hardware, thus we prioritize the prior. All our sensors can be seen in Fig. 1.

1) Continuous Measuring Hardware: When designing our project, we figured we would need to limit ourselves in terms of cost due to large costs for each sensor that would be needed. As a result, we focused our search on sensors that were accurate, economic and could have continuous data collection. First off, we have our continuous measuring hardware, the MAX86150 and MAX30205, which are seen below in Fig. 1. We defined continuous measuring hardware as sensors that can record data continuously and logs it onto .CSV files. These sensors were essential for the project because they streamlined the process of having to manually log results while also giving us continuous data.

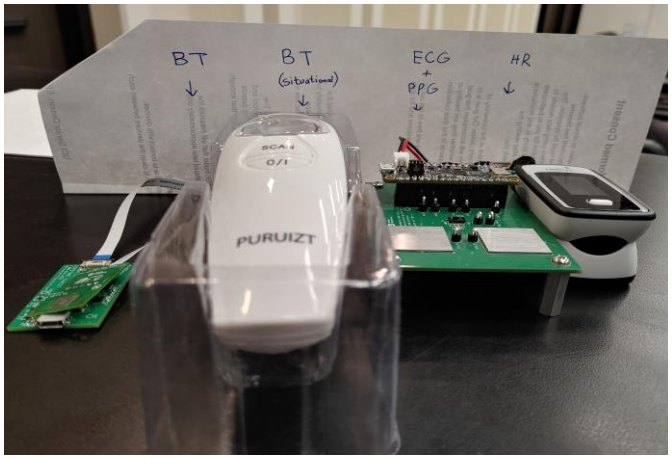


Fig. 1. The collection of all hardware involved. The MAX30205 (left), Puruizt Thermometer (middle left), MAX86150 (middle right), and Innovo Deluxe Fingertip Pulse Oximeter. Continuous sensors are the left and middle right sensors and one-shot sensors are middle left and right sensors.

Similarly, these sensors also minimize the odds of bad data because we gather continuous data as opposed to singular points of data.

a) MAX86150: The MAX86150 functions as two sensors integrated onto a singular evaluation board, being a PPG and ECG sensor. We elected to use this sensor because of its real-time monitoring and data logging capabilities found in this sensor. Additionally, we can interface the evaluation board directly with a laptop by using a, a 3.7V battery and Bluetooth [5]. Fortunately, the battery is rechargeable, and thus reduces costs by not having to purchase multiple of them and reduces waste from the data collection process. From Fig. 2, we see the fully assembled circuit in its on-state, meaning that it is connected via Bluetooth to a computer. The ECG sensor operates by firmly placing one thumb on each of the two metal electrodes seen in Fig. 2 and waiting for data to be collected and stored. Similarly, the PPG sensor operates by placing your index finger on top of the sensor above the red pin in the center of the board. The PPG sensor operates by emitting a red light that is then used to analyze the volume of the blood circulating through the finger on the sensor in terms of IR count. This means that the subject can tell when data is being collected by noting the red light on their index finger.

b) MAX30205: The MAX30205 is our body temperature sensor that was chosen because of its 0.1°C accuracy around the average human body temperature range, 37°C to 39°C [6]. The sensor is integrated onto an evaluation board like the MAX86150 and can run directly onto the user's laptop which is useful for simple data collection. The evaluation board comes with two distinct parts, the

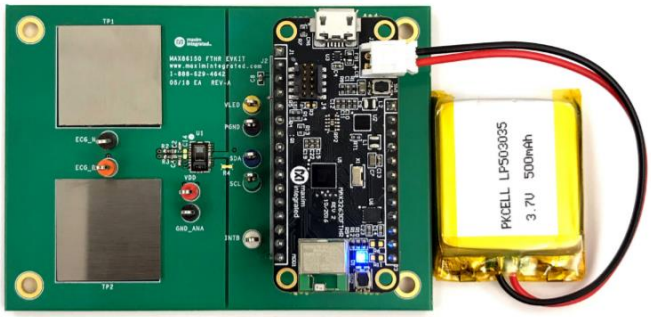


Fig. 2. The MAX86150 evaluation kit in its on state, connected to a source. The board is currently operating on Bluetooth and is using a dedicated 3.7V source to power it. Dedicated pin holes on the board allow us to interact with the sensor embedded into it. This image is property of Maxim Integrated.

Fig. 3. The MAX30205 evaluation kit in its off state, connected to its controller PCB. The MAX30205 has a metal electrode underneath the image shown which is used to take data. The 10-pin flex cable interfaces the controller to the sensor and the controller provides a I2C to USB interface. This image is property of Maxim Integrated.

sensor and electrode where all the data is taken, and the controller PCB which is done for an I2C to USB interface seen in Fig. 3. The sensor operates by placing a finger on the electrode under the MAX30205 and allowing the electrode to warm up. After some delay, the sensor begins to provide results. Unfortunately, due to how the board was constructed, it takes approximately 90 seconds to reach actual body temperature; however, depending on how the subject holds the sensor it has a notable error because of its shape. As such, we used this sensor to measure our body temperatures because we had the time to continuously record our data and experience to manually use it.

2) One-Shot Hardware: One-shot hardware are the sensors we designated as the sensors that can only produce singular points of data. This hardware requires us to manually input data and log them onto .CSV files ourselves. Although these sensors have the drawback of only producing singular points of data, they do have some benefits. The benefits include being more time effective, being simple to administer to multiple subjects and being easier to sterilize between uses.

a) Puruizt Non-Contact Infrared Thermometer: The Puruizt is our situational body temperature sensor, as seen in Fig. 4a. We denoted the Puruizt as situational because it was used primarily during the sampling process because it takes no setup and is quick to get data points. An important feature to note is that when the subject is in the acceptable range, the background light of the Puruizt is green. On the other hand, if the temperature is not in an acceptable range, we know this person has or is developing a fever. The Puruizt, seen in Fig. 4, is a simple to use thermometer that can record data from the last samples and functions well. We operate the Puruizt by pressing the I/O button to turn it on and then holding it up to the subject's forehead and getting a sample.

b) Innovo Deluxe Fingertip Pulse Oximeter: The Innovo is our HR sensor that can track both the heart rate and pulse oximetry of the subject. Unlike the Puruizt, the Innovo can provide continuous data, but it is gated by its inability to record data. As such, we used it for one-shot data because we only record the data seen once it stabilizes. The Innovo works by pressing the button seen in Fig. 5, and then placing the subject's finger inside of it. Afterwards, the subject places their finger flat on the table and waits for a stable value to be displayed on the screen. The portability and easy usage allow it to be carried without a problem and allows us to maneuver around and find volunteers for the sample.

(a) (b)

Fig. 4. (a) Puruizt body non-contact infrared thermometer in its on state showcasing the acceptable and danger range screens. (b) Innovo deluxe fingertip pulse oximeter in its on state showing heart rate and pulse oximetry. These images are property of Puruizt and Innovo, respectively.

B. *High Level Software*

1) Android Studio: Our project is one that is extremely interactive with the user and as such the user should have an engaging way to access their results. The method that we ultimately decided on was the access of data through a mobile application, specifically an android application. The application was designed with mobility in mind, meaning the results can be accessed anywhere at any time. To do this, TCP sockets were used to allow for connectivity and data sharing from anywhere there is Internet availability. This will allow our users to review their data at their leisure.

2) Server: The purpose of this server is to store the gathered and processed information for future access by the study participants. The Android application is a means to access the data that is collected and logged onto the server. Again, as seen in Fig. 5, TCP sockets are used for communication between the server and the Android application.

3) User Interface (UI): A user interface can be best described as a way to interact with a computer or device. The user interface in this project are the sensors and the means of displaying their results from the data collected. The sensors are a means to feed inputs to the device while the displayed results are the outputs or reactions to the inputs. In other words, the interface would comprise of the sensors as well as the screen the results will be displayed on.

Additionally, the previously mentioned android application is also a user interface as it is a way to interact with the project. It is a way for the user and device to exchange information. The input in this case would be a means of identification and the output would be the previously recorded results.

```
socket created
The Host is 169.254.174.11
The Port is 7000
Socket has been binded!
Socket is now listening

Connect with 169.254.174.11:53917

The Server Has Connected To Patient Subject Number:
2
Connect with 169.254.174.11:53981

The Server Has Connected To Patient Subject Number:
8
```

Fig. 5. This is the active Python Server that shares information to the Android Application.

3) TensorFlow: The TensorFlow Software Module was utilized for its machine learning capabilities. More specifically, using its Keras library, we were able to perform various functions on our datasets which have helped us merge the data into one general predictive model. We are currently still in the process of determining the model of best fit for our collected data trying various models such as a Convolutional Neural Network, a Linear Regression, and a Logistic Regression. This software module is backed by various other machine learning libraries such as matplotlib, for graphing utility, and pandas, for converting datasets into categorized tables that can then manipulated into test data which can further be analyzed using keras functions.

4) MAX30205EVSYS Software: The MAX30205EVSYS Software was used primarily for the prototype and for simple data collection. It provides us an easy way of interacting with the MAX30205 without having to go through the Pi and stores the data quickly and easily onto a designated .CSV file as seen below in Fig 6.

The software is easy to use and has many useful features, such as one-shot mode that only takes the instantaneous temperature, switching between Celsius and Fahrenheit, and an easy access to the data logs. The software also allows us to change the .CSV file name whenever we need to make a new data set, and as such provides us an easy way of tracking when we took the data. Unfortunately, the software only allows us to change the refresh rate by increments of ten seconds, to avoid overheating the sensor.

5) Device Studio: Device studio is a software for a few Maxim sensors, but we only have the MAX86150 that interacts with it. Fig. 7 shows the typical screen when running the software, as we will be using it mainly for just gathering and storing data like the MAX30205 Software. Unlike the MAX30205, the MAX86150 stores continuous data so we can plot the data as a function of mV vs time as opposed to just discrete values. As such, we can plot filtered and raw ECG plots by clicking on the value and having it appear. This enables us to use filtered data for the UI to show a clean graph as opposed to the noise that is produced from the raw ECG. Similarly, when the PPG data is plotted directly beneath our filtered ECG, which allows operators of the device, and participants, to have the opportunity to see the results immediately. The software also allows us to record data onto a singular .CSV file, where each new reading is replaced by new headings such as in Fig. 8. It is important to note that even when there is no data recorded, a new header is produced. Similarly, when changing the header settings, it also produces a new header line.

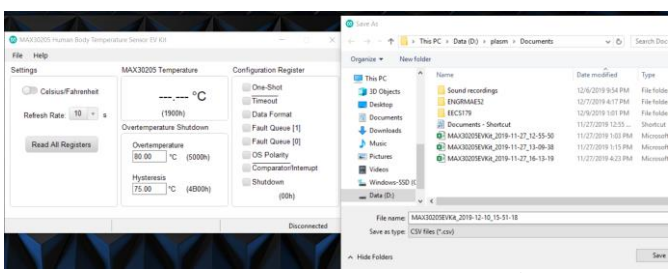


Fig. 6. MAX30205 Human Body Temperature Sensor software screenshot without the sensor connected. We can access the .CSV files for our data model by clicking on Log File.

C. CITI and HIPAA Regulatory Materials

CITI regulations required us to ensure that human participants are only revealed to minimal risk. CITI defines minimal risk as no danger or discomfort greater than the participant may experience in a day-to-day basis. To ensure that we met this criterion, we cleaned our hardware between each use by using 99% isopropyl alcohol (IPA). 99% IPA is electronic safe and is used as a disinfectant for surfaces. As such, we reduce the chance of infection by participation by ensuring that we sterilized all equipment that the participants would interact with. Similarly, as per Health Insurance Portability and Accountability Act (HIPAA) regulations, we need to have an informed waiver of consent to show that all participants were informed of all risks and were voluntary participants. Our informed consent waiver involved two pages of logistical information, such as explaining who we are, the purpose of our project and all risks that participants agree to. Afterwards, on the third page we had a short questionnaire asking about the health status of our participant so we can label their data. Lastly, the final page of our waiver asked for the date, signature, and printed name of the participant and the group member who was there when the participant agreed. This ensured that all parties are well documented and understanding of the situation and thus avoid any misconceptions, as per HIPAA standards.

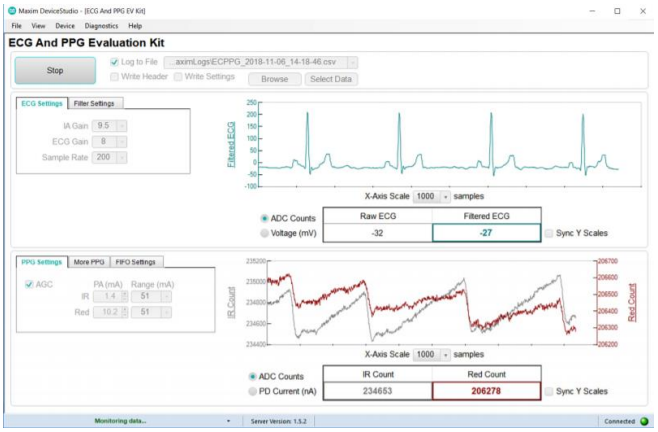


Fig. 7. Device Studio showing an example of the MAX86150 being evaluated and producing an ECG and PPG plot. We have the option of plotting raw or filtered ECG while logging both. The screenshot is property of Maxim Integrated.

Automatic	IR PA (mA)	Red PA (m	IR LED Rar	Red LED R	ALC + FDM	Sample Ra	Pulse Widt	ADC Range	FIFO Rolls	Fl
Time	Sample Co	IR Count	IR Count (	Red Count	Red Count	Raw ECG	Raw ECG	Filtered EC	Filtered ECG	
28:10.9	10795	243630	15226.9	216011	13500.71	-2104	-0.64419	-2138	-0.6546	
28:10.9	10796	243620	15226.28	216022	13501.4	-2092	-0.64052	-2142	-0.65583	
28:10.9	10797	243613	15225.84	216040	13502.53	-2100	-0.64297	-2145	-0.65675	
28:10.9	10798	243621	15226.34	216070	13504.4	-2129	-0.65185	-2146	-0.65705	
28:10.9	10799	243644	15227.78	216077	13504.84	-2123	-0.65001	-2146	-0.65705	
28:10.9	10800	243660	15228.78	216097	13506.09	-2103	-0.64389	-2145	-0.65675	
28:10.9	10801	243681	15230.09	216134	13508.4	-2105	-0.6445	-2143	-0.65613	
28:10.9	10802	243711	15231.97	216172	13510.78	-2103	-0.64389	-2141	-0.65552	
28:10.9	10803	243754	15234.65	216217	13513.59	-2086	-0.63868	-2139	-0.65491	
28:11.0	10804	243785	15236.59	216278	13517.4	-2112	-0.64664	-2137	-0.6543	
28:11.0	10805	243829	15239.34	216358	13522.4	-2089	-0.6396	-2135	-0.65368	
28:11.0	10806	243868	15241.78	216399	13524.96	-2076	-0.63562	-2131	-0.65246	
28:11.0	10807	243923	15245.22	216435	13527.21	-2089	-0.6396	-2127	-0.65123	
28:11.0	10808	243950	15246.9	216491	13530.71	-2091	-0.64021	-2123	-0.65001	

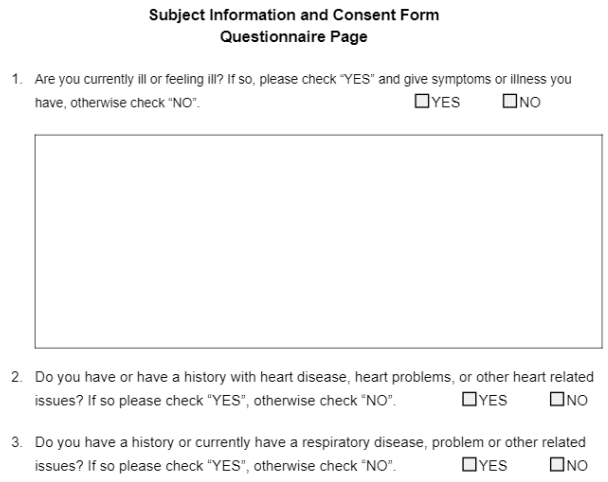
Fig. 8. A .CSV file taken after running through three tests on Device Studio. This is only the first 16 rows of the file, the whole .CSV files is much larger.

III. Methods and Results

A. Methods

1) Developing a Data Model: For our model development we have created our general structure. We have split our data sets into test and train data for predictive modeling. Utilizing machine learning libraries such as Keras from TensorFlow, Matplotlib, and Pandas, we have begun training a model that will predict influenza to a relatively high order. As of current the model is somewhat barebone as there is a lack of data collected from a diverse sample size. We are merely taking various output from function generators and applying the sensors on ourselves. We are currently in the process of testing multiple model types to best predict our data such as a multiple linear regression, which uses more than one predictor to determine a linear output giving all states/levels of sickness, as well as logistic regression, which results in a binary output dictating whether someone has the flu or does not. We have also pondered the idea of using a Recurrent Neural Network as it may possibly help the predictability factor of our data however, we do not have enough data sampled to create an accurate model from this method. As we collect more data in the near future, we will test this algorithm once more and determine whether it will be an accurate method for flu prediction.

2) Acquiring data: To acquire data we followed a rather strict approach. First, we had to find volunteers, that is students on campus who are willing to take the time needed to participate. Afterwards, we give the subject the informed consent waiver and have them fill out the questionnaire so we can identify if we should be cautious about using their data with the general subjects. For instance, in Fig. 9, we see a few questions from the third page of the waiver which asks our subject for basic health information and their age. This allowed us to flag participants who were at risk or had the flu. Unfortunately, most participants we had were healthy and did not have any of the health problems listed in Fig. 9. As such, we only needed to flag participants as with the flu or healthy. Afterwards, we began with gathering the subject's data. We first began with our ECG and PPG sampling, followed by HR and, lastly, the BT sampling. While the MAX86150 records the data onto a unique .CSV file that automatically includes the date and time, we must fill out our own .CSV file for the BT and HR readings. When making our .CSV file, we addressed the flu status of all subjects by including a binary system like in Fig. 10. To complete the process there are two small steps. We first allow the subject to view their respective results and confirm to them that their name is not used anywhere besides on the waiver. The last step is we sterilized our sensors by using our 99% IPA and cleaning all places that received contact. Due to this step, we had to leave approximately 90 seconds to ensure that all places the IPA contacted were dry. This ensured that the equipment would not be damaged and that IPA does not make contact with anyone's skin.



Subject Information and Consent Form
Questionnaire Page

1. Are you currently ill or feeling ill? If so, please check "YES" and give symptoms or illness you have, otherwise check "NO". ☐YES ☐NO

2. Do you have or have a history with heart disease, heart problems, or other heart related issues? If so please check "YES", otherwise check "NO". ☐YES ☐NO

3. Do you have a history or currently have a respiratory disease, problem or other related issues? If so please check "YES", otherwise check "NO". ☐YES ☐NO

Fig. 9. These are the first three questions from page 3 of our informed consent waiver, asking for basic information that would potentially flag participants.

Body Temperature (°F)	Subject Number	Date (MM/DD/YYYY)	Flu Status
96.8	5	2/5/2020	0
98.4	6	2/5/2020	1
98.4	7	2/5/2020	0
98.9	8	2/5/2020	0
96.6	9	2/5/2020	0
96.8	10	2/5/2020	0

Fig. 10. An example of our .CSV file containing subject body temperature. We remove all identification of our subject aside from their respective subject

B. Results and Performance

The results of the completed project were expected; we have a fully functioning system for data collection, data processing, and displaying vital information to the participants of the study. One of the most notable points of our project is the fact that data collection takes minimal time, approximately 2 minutes. The 4 parameters are collected in a fashionable time and can compete with the RIDT's 15-minute influenza detection system.

Despite the speed of our data collection process, there are sometimes inaccuracies with the sensors. When reviewing the data measurements that occur from improper use of the sensors. The samples remain kept, but the extreme outliers are taken out of the data display and model learning.

Our model utilized the K nearest neighbor supervised learning algorithm to determine flu outcomes on a binary basis i.e. (has the flu or does not have the flu). The model is able to predict with an accuracy of 96% whether a patient has the flu or not. Given our small initial dataset we were forced to create further data to make a more robust model. This data was created through an algorithm extrapolated from the correlations between our collected data entries. The resulting model is shown below determining the number of neighbors, k, providing the highest accuracy.

The resulting k weight with the highest accuracy was found to be 6 with an accuracy of 96.25%.

When looking at the results of the UI, in Fig. 11, we were satisfied that it could interpret data and display it based on our specifications. Although most data can be understood, the large range in PPG and ECG make the information hard to decipher. The variance in these two sets of data force the images to be zoomed out, making it difficult to read the information, such as in Fig. 12. Of course, this is only a problem for viewing as it does not affect the accuracy of the model. Despite the flaw, the user-interface works as intended and fulfills the goal of relaying information to the user.

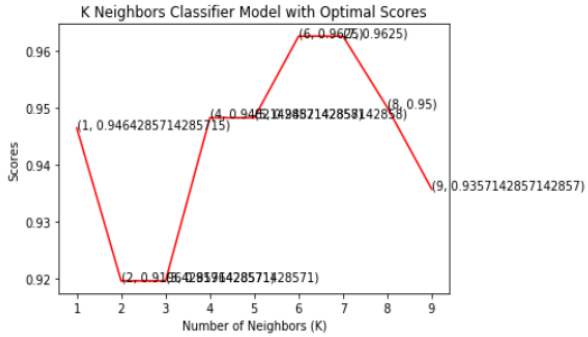


Fig. 11. Our K Nearest neighbor distribution looking through multiple cycles of K values until one with a very high accuracy appears.

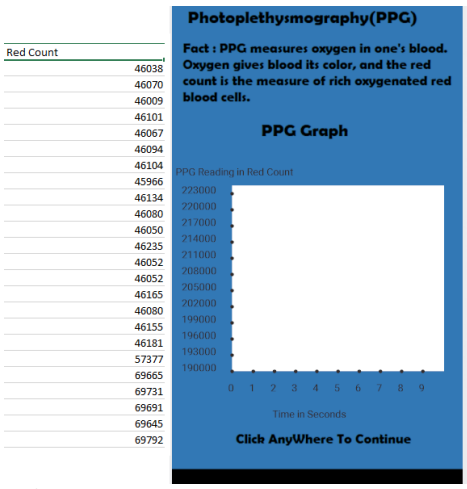


Fig. 12. This figure shows an inaccurate PPG reading. The data is not normal, so the information is not displayed to the user.

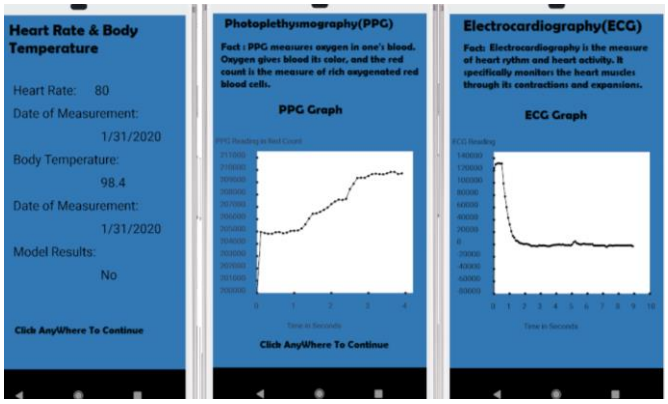


Fig. 13. This figure is the final product of the Android Application and the three different screens the subject can view.

IV. Conclusion

The influenza virus is an annual devastator that requires containment to reduce the number of impacted people. As such, *Watch the InFLUence* is here to help contain by detecting early signs of the influenza by analyzing and comparing different body parameters. Although we had minor malfunctions with our Raspberry Pi and other sensors we were able to develop the baseline for a more automated system as an embedded medical device. With our frontend UI and our finalized model we are able to display all the necessary parameters and predictions to help detect the influenza in a patient as soon as possible. Although our model without the proper hardware will not immediately detect the virus at the fastest rate given the lack of hardware and in person data, it will still determine an outcome with a very high certainty. Our initial goal at the beginning of two quarter interval may not have been met exactly due to all the confounding factors, but we still were able to pick it up due to ingenuity found between all project members' collaboration. We are pleased with the results after the number of unforeseen obstacles we faced and are glad to have partaken in such a project. With our software running in full functionality, only hardware integration is needed to allow this project to take the next leap forward.

Expanding beyond this year, the project can be broadened by incorporating new models to detect signs of heart problems and to enhance overall performance. This is consistent with the influenza because it is known that the influenza places a lot of stress on the heart and can lead to multiple issues arising in patients with weak hearts. As such, incorporating this into our diagnosis could potentially save at-risk patients who could potentially have a heart attack or other heart related disorders because of the flu. In addition, the very recent development of the COVID19 virus, also known as the "Corona" virus, which shares many parallels with the flu is a topic worth expanding on. Adapting and further parametrizing our model could help in the relief effort by finding the strict outliers that can differentiate between the two viruses. On the other hand, instead of adding new features, there are other ways to enhance the ability of our model. One of the limiting factors of our model is that developing flu symptoms leads to nuances in user-specific data. This means that without having access to our subject's ECG, PPG, HR or BT we lack the ability to give a near-100% diagnosis. If we had our user's data continuously monitored, minor fluctuations such as an increase in resting BT or HR, we could begin flagging subjects.

Acknowledgment

Team Influence would like to thank our advisor, Prof. R. Esfandyarpour for agreeing to be our advisor and giving us his assistance, the TAs of EECS159 for their guidance, and to the Undergraduate Research Opportunity Program for giving us funding to help make this project possible. We would also like to thank Prof. S. Kleinfelder for allowing us to enter his lab section for quarter 2. I. Madrigal would also like to thank IEEE for allowing us to use their soldering station, Prof. P. Tseng for directing us to Prof. R. Esfandyarpour, and lastly Prof. R. Esfandyarpour and his grad student, Seyed Ali Hashemi, for helping us with our sensor problems. A. Banerjee would like to

thank Beverly A. Quon for providing feedback on a week to week basis concerning project progression and Prof. R. Esfandyarpour for advising the group on the project decision while also providing the major building blocks needed to see it through. Team Influence would also like to acknowledge the hard work and enthusiasm of all our team members: A. Banerjee, I. Madrigal, and Q. To.

REFERENCES

[1] "How Long Is the Flu Contagious?" webmd.com. https://www.webmd.com/cold-and-flu/how-long-flu-contagious (accessed Oct 6, 2019),

[2] S. Ghebrehewet, P. MacPherson, and A. Ho, "Influenza," *BMJ*, vol. 2015, no. 355, Dec. 2016, doi: 10.1136/bmj.i6258.[Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5141587/. Accessed: Oct. 8, 2019.

[3] "Diagnosis." cdc. https://www.cdc.gov/flu/symptoms/testing.htm (Accessed Oct. 8, 2019).

[4] Wertheim, et al., "Monitoring respiratory rate from pulse oximeter plethysmogram analysis in acutely wheezy young children," *European Respiratory Journal*, vol. 48, no. 60, Nov. 8, 2016, doi: 10.1183/13993003.congress-2016.PA1227. [Online]. Available: https://erj.ersjournals.com/content/48/suppl_60/PA1227.article-info. Accessed: Oct. 20, 2019

[5] Maxim Integrated, "MAX86150 Evaluation System," 319-100012,April 2017 [Revised: June 2019].

[6] Maxim Integrated, "MAX30205," 19-8505, March 2016.

[7]Somani, Uma, Lakhani, Kanika; Mundra Manish. "Implementing Digital Signature with RSA Encryption Algorithm to Enhance the Data Security of Cloud in Cloud Computing". *2010 1st International Conference on Parallel, Distributed and Grid Computing.*