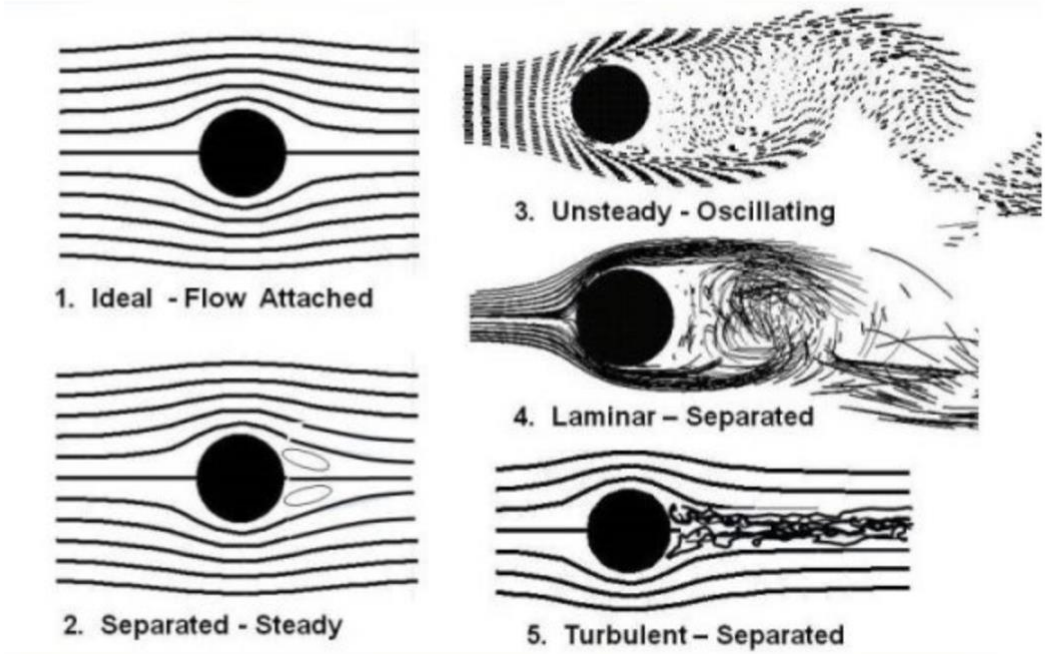
DESE71003 – Sensing and Internet of Things
Imperial College London
Dyson School of Design Engineering

Ebb & Flow



[Image source - NASA Glenn Research Center]

Author: Alexis Almeida

Code & Data URL:
https://github.com/alexisalmeida99/Ebb-and-Flow/

Coursework Video URL:

Date: December 12, 2024

Introduction

Creativity is highest when in a calm state. When meeting deadlines and navigating the normal stressors of life, it can become difficult to maintain the relaxed state creativity demands.
For this project, I wanted to find a way to make emotions 'feel' more tangible. impact by providing visual feedback of how one's emotional state expresses itself on the body, I hope to find a more concrete strategy to work through the mental programs that elicit unnecessary fear responses.

The rising prevalence of anxiety-related conditions has prompted the need for tools that empower individuals to monitor and manage their mental health. Breathing exercises are often recommended as an effective method to alleviate anxiety, yet quantifying their impact remains a challenge. This project aims to bridge this gap by designing and developing a wearable device that can measure physiological indicators of anxiety and eventually evaluate the effectiveness of techniques such as breathwork. "real" feedback at home.

For millennia humanity developed mechanisms to deal with stress and to live a peaceful life. However, with the integration of the internet into our daily lives, stress and mental health conditions have declined for many. Practices of mindfulness and meditation have risen to combat this rise, with many taking moments out of their day to identify patterns of behaviour, triggers of stress, and to attempt to calm their mind. Developments in technology could aid these practices.

The project integrates sensing technologies and machine learning to provide real-time feedback on anxiety levels. Utilizing a heart rate sensor, a galvanic skin response (GSR) sensor, and a skin temperature sensor, the device collects physiological data that correlates with stress and anxiety. This data is processed and analyzed, offering insights into the user's state of mind. The project also serves as an opportunity to delve into machine learning applications in the context of emotional well-being and to develop an accompanying application for visualizing and interpreting the collected data.

This initiative aligns with the broader trend of leveraging Internet of Things (IoT) technologies for health monitoring and management. By focusing on anxiety assessment, the project demonstrates the potential of wearable devices to support mental health interventions and promote self-awareness.

Objectives

1. Measure heart rate
2. Measure galvanic skin response
3. Measure skin temperature
4. Predict anxiety using machine learning models
5. Display results on a live web dashboard giving users real time feedback

System Overview and Architecture

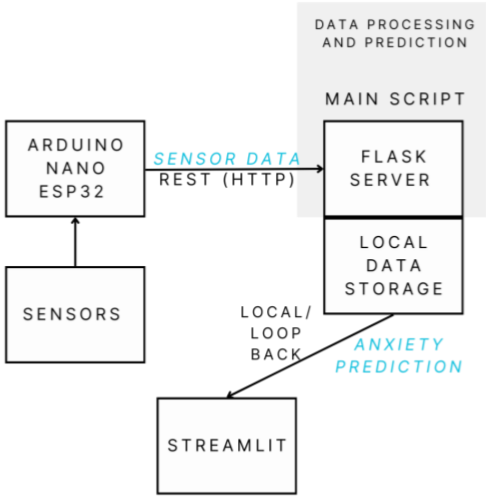


Figure 1: Diagram of system architecture

The end-to-end system includes two sensors attached to an Arduino Nano ESP32. Given the sensitivity of the personal data, I chose to keep everything local. The Arduino samples data from the heart rate/ temperature sensor and GSR sensor and every 10 seconds, sends a packet to the flask server. The flask server then preprocesses the sample as part of our machine learning pipeline. The next step is inference in which I use a Random Forest Classifier taken from scikit-learn. This model makes a prediction which is then stored in our local database in json format. From there, the Streamlit application reads the most recent entries and displays them to our user. Once the flask server shuts down, the data base is cleared due to the sensitive nature of the data. Possible iterations and improvements of this architecture would include encryption of data in transit (https) and at rest so the database does not have to get wiped. This would also allow for further fine tuning of the machine learning pipeline and possibly use of a larger more robust classifier like a dense CNN/ANN (convolutional/ adversarial neural network).

Sensor Background and Set Up

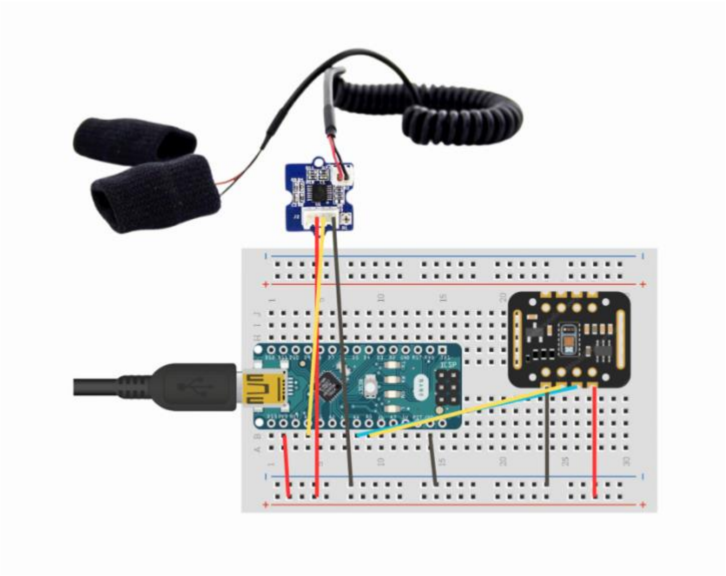


Figure 2: Circuit Schematic depicting sensor set up.

1) Grove - GSR Sensor

Parameter	Value/Range
Operating voltage	3.3V/5V
Sensitivity	Adjustable via a potentiometer
Input Signal	Resistance, NOT Conductivity
Output Signal	Voltage, analog reading
Finger contact material	Nickel

Table 1: Grove GSR Sensor Parameters and Values

Galvanic Skin Response (GSR) measures the electrical conductivity of the skin. Emotional arousal triggers the sympathetic nervous system, increasing sweat gland activity. This enhanced perspiration leads to increased skin conductance, which can be detected by attaching electrodes to two fingers. GSR offers a valuable tool for developing emotion-related projects, such as sleep quality monitors (3).

The GSR sensor outputs analogue voltages that correspond to a person's skin conductance. These analogue voltages area unique to the microprocessor, and need to thus be converted into nano siemens. To do this conversion, I use Ohm's law to calculate the proportion of the supply voltage to determine the resistance between the electrodes. I used a multimeter to determine the resistance of the potentiometer (the reference resistor) the sensor uses. Using this value and the input voltage value of 3.3V, I then used Ohm's law to find the skin's resistance.

2) MAX30102: Heart Rate and Temperature Sensor

The MAX30102 is a Photoplethysmography (PPG) sensor, an integrated pulse oximeter and heart rate sensor IC from Analog Devices. It communicates using an I2C interface. The MAX30102, along with other optical pulse oximeters, employs a dual-wavelength LED system (red: 660nm, infrared: 880nm) and a photodetector to measure physiological parameters (1)

Parameter	Value/Range
Power supply	3.3V to 5.5V
Current draw	~600µA (during measurements)
Red LED Wavelength	660nm
IR LED Wavelength	880nm
Temperature Range	-40˚C to +85˚C
Temperature Accuracy	±1˚C

Table 2: MAX30201 Sensor Parameters and Values

Oxygenated haemoglobin (HbO2) in arterial blood exhibits a distinct absorption of infrared (IR) light. Blood with higher haemoglobin concentration, appearing redder, absorbs more IR light. During each heartbeat, the pulsatile blood flow through the finger alters the amount of reflected light. This dynamic change in reflected light, captured by the photodetector, generates a waveform that directly correlates to the heart rate (HR). Continuous illumination and photodetector readings rapidly yield an accurate HR pulse reading.

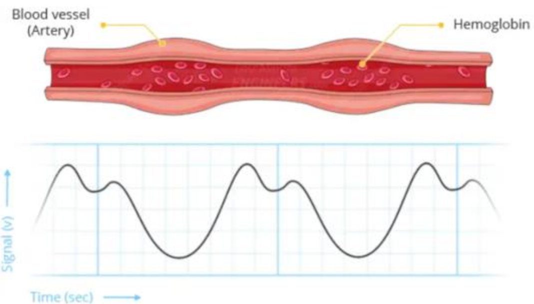


Figure 3: Principles of PPG Sensor [Image source: Last Minute Engineers]

The MAX30102 also includes a temperature sensor on the board with a high accuracy of ±1˚C.

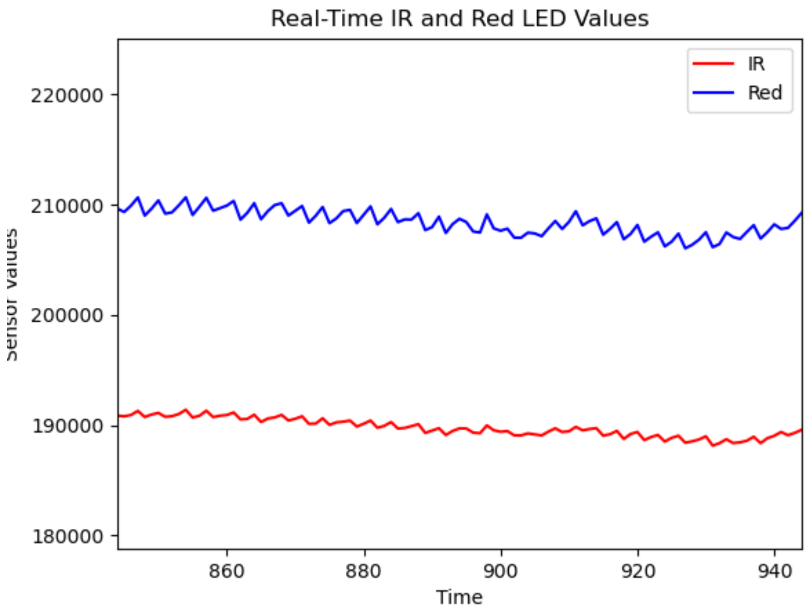


Figure 4: MATPlotlib graph of filtered MAX30102 sensor output

MAX30102 also includes a FIFO buffer embedded for storing data samples. The FIFO has a 32-sample memory bank, allowing it to hold up to 32 SpO2 and heart rate samples.

```
void loop() {
        long irValue = particleSensor.getIR();

  if (checkForBeat(irValue) == true)
  {
    if (irValue < 30000) Serial.println("No finger detected!");
    else {


    //We sensed a beat!
    long delta = millis() - lastBeat;
    lastBeat = millis();

    beatsPerMinute = 60 / (delta / 1000.0);

    if (beatsPerMinute < 255 && beatsPerMinute > 20)
    {
      rates[rateSpot++] = (byte)beatsPerMinute; //Store this reading in the array
      rateSpot %= RATE_SIZE; //Wrap variable

      //Take average of readings
      beatAvg = 0;
      for (byte x = 0 ; x < RATE_SIZE ; x++)
        beatAvg += rates[x];
      beatAvg /= RATE_SIZE;
    }
  }
  }
```

Figure 5: Arduino code – main loop with BPM computation

In the main loop for the sensor, I calculate the heart rate in beats per minute (BPM) using a peak detection algorithm provided by the creators of the sensor. This calculation is very expensive computationally which meant that I needed to be creative if I wanted to collect additional readings for GSR and temperature data because the Arduino was under so much load. Specifically, the heart rate needed to be sampled at a very precise rate in order to preserve

```
unsigned long lastSend = 0;
const unsigned long interval = 10000;

void loop() {
        long irValue = particleSensor.getIR();

 if (checkForBeat(irValue) == true)
  {
   if (irValue < 30000) Serial.println("No finger detected!");
   else {


   //We sensed a beat!
   long delta = millis() - lastBeat;
   lastBeat = millis();

   beatsPerMinute = 60 / (delta / 1000.0);

   if (beatsPerMinute < 255 && beatsPerMinute > 20)
   {
     rates[rateSpot++] = (byte)beatsPerMinute; //Store this reading in the ar
     rateSpot %= RATE_SIZE; //Wrap variable

     //Take average of readings
     beatAvg = 0;
     for (byte x = 0 ; x < RATE_SIZE ; x++)
       beatAvg += rates[x];
     beatAvg /= RATE_SIZE;
   }
  }
  }


       String jsonPayload = "{";
       jsonPayload += "\"ir\": " + String(irValue) + ",";
       jsonPayload += "\"beatAvg\": " + String(beatAvg) + ",";
       jsonPayload += "\"HR\": " + String(beatsPerMinute)+ ",";


       if (lastBeat - lastSend >= interval){
         send_packet(jsonPayload);
         lastSend = lastBeat;
       }
```

Figure 6: Heart rate calculation and sampling

stability and accuracy. This meant that the additional sensors needed to be exercised in between heartbeat calculations. Additionally, I also had to handle the networking component which was equally as expensive.

To achieve stable readings, I simply chose to sample my sensors at the last possible moment before I sent a packet which was every 10 seconds. In the remaining 9.99 seconds, the Arduino was performing heartbeat calculations.

Data Gathering

Originally, I tried to create a system that would classify emotions based off valence and arousal. To this end, I downloaded the DEAP dataset which contained over 40 features and these two labels. Unfortunately, the signals I was interested in, such as Heart rate, GSR, and skin temperature were extremely difficult to ascertain. Although the dataset advertised these signals as features, they were incompatible with my sensors for various reasons. The main one being that they published analogue counts for their readings I assume, because the data was incomprehensible even after denoising, detrending, and imputing. They also did not give much information in how they calculated/ what devices they used for the signals I cared about as the study was mainly based on EEG data. Ultimately, this forced me to pivot to predicting anxiety instead of arousal/valence. Thankfully, the dataset provided by Professor David Boyle on his research into *Detecting Subclinical Social Anxiety Using Physiological Data from a Wrist-worn Wearable* (4) was extremely well documented and pre-processed.

Data Analysis

To train the classifier, I implemented a supervised machine learning algorithm, called Random Forest Classification and trained against 10-fold cross validation achieving an accuracy of 97.85%.

```
python train.py
Successfully loaded './data/master_experiment_1.csv'.
Dropped 'Unnamed: 0' column.
Renamed columns to ['HR', 'ST', 'EDA', 'Labels'].
Performed train-test split with test size = 20.0%.
Initialized Random Forest with specified hyperparameters.
Created a pipeline with StandardScaler and Random Forest.
Starting 10-fold cross-validation...
Cross-validation completed in 100.57 seconds.
Cross-Validation Accuracy: 97.85% (+/- 0.26%)
Cross-Validation Log Loss: 0.1322 (+/- 0.2810)
Training the model on the entire training set...
Model training completed.
Evaluating the model on the test set...
Test Accuracy: 97.62%
Test Log Loss: 0.1296
Trained model saved as 'random_forest_experiment1.pkl'.
```

Figure 7: Output of train.py

To use the model for prediction on the Flask app, I loaded in models, weights as pickle file. I then took a raw reading from the Arduino and preprocessed it. This preprocessing included translating my GSR readings into EDA and denoised. Then I took a tupple of HR, ST, and EDA values and fed them to the model. The output was a label of 1 or 0 as it was a binary classification task where one meant anxious. I saved the tupple as well as the prediction to my local JSON database for future presentation on the streamlit APP.

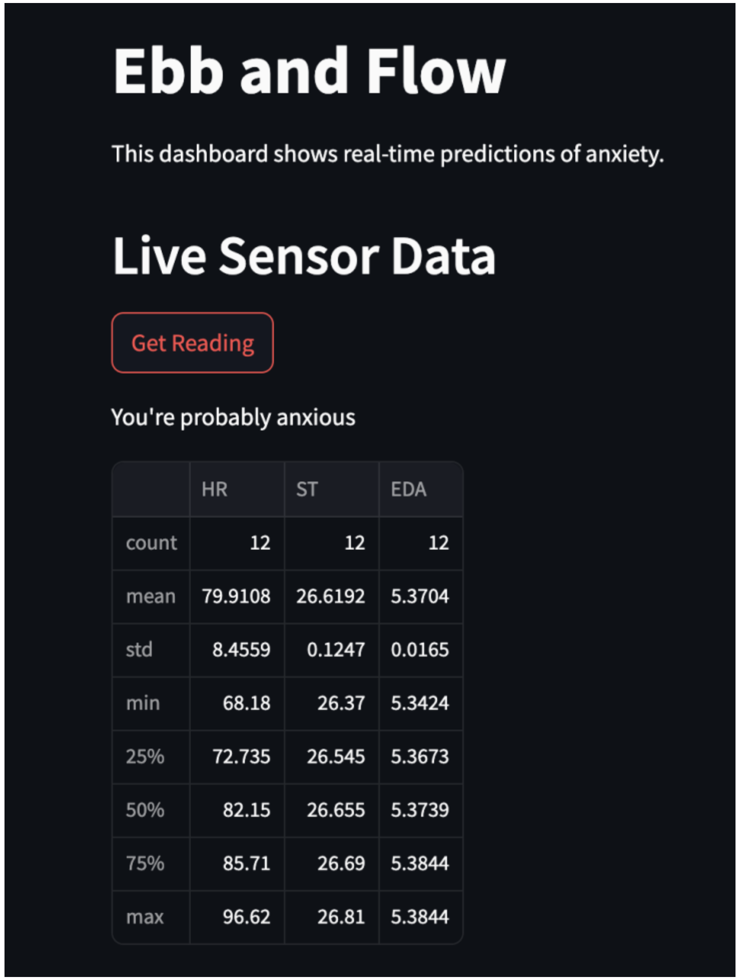


Figure 8:Streamlit Ebb and Flow Dashboard

Discussion

The development of the Ebb & Flow system demonstrates the potential of wearable technology to enhance mental health awareness and intervention. By successfully integrating heart rate, GSR, and skin temperature sensors with machine learning models, the project offers a novel approach to predicting and understanding anxiety levels in real time.

Key Insights and Challenges

1. **Data Interpretation**: The transition from valence-arousal classification to anxiety prediction proved pivotal. Utilizing well-documented datasets ensured reliable model training and facilitated accurate predictions. However, the initial challenges with incompatible datasets highlight the importance of robust data preprocessing and documentation.
2. **System Efficiency**: Managing computational loads on the Arduino Nano ESP32 while maintaining sensor accuracy was a significant challenge. By optimizing the sampling intervals, the system achieved a balance between accurate readings and resource management, underscoring the importance of efficient hardware-software integration.
3. **User-Centric Design**: The use of Streamlit for real-time feedback ensures an intuitive user experience. However, extending functionality to include customizable visualizations of one's anxiety or integration with personal health apps could further enhance user engagement.
4. **Lack of resources**: A significant challenge faced during this project was the limited access to specialized electrical engineering facilities and resources. As a student in the Dyson School of Design Engineering (DSDE), I encountered difficulties troubleshooting issues with my sensor setup, as the necessary equipment and support were often unavailable. This limitation forced me to rely on trial and error using components sourced from Amazon, which significantly extended the development timeline. Weeks of effort were consumed in identifying and resolving basic connectivity and functionality issues that could have been expedited with access to professional-grade tools and expertise.
   This experience highlights the importance of interdisciplinary collaboration and access to comprehensive facilities when undertaking complex projects that combine design engineering with advanced electronics. In future iterations, partnering with electrical engineering labs or securing access to a broader range of equipment could greatly enhance the efficiency of development processes and the reliability of outcomes.

Future Work and Improvements

- **Visualizations of stress:** In an ideal scenario with unlimited time and resources, the project could have been extended to include a visualization model to represent the impact of stress on an individual's problem-solving ability. This visualization would leverage a metaphorical approach, modeling stress as a turbulent fluid flowing through a pipe with a blockage. High stress levels would correspond to chaotic, turbulent flow, emphasizing how stress hinders progress. Conversely, employing stress-reduction techniques such as breathwork would gradually transform the fluid dynamics into a smooth, laminar flow, illustrating the efficiency gained when one achieves a state of calm. This dynamic model could serve as both a diagnostic tool and a motivational aid, enabling users to see tangible evidence of how stress management enhances their mental clarity and problem-solving abilities. Integrating such a visualization would have not only enriched the user experience but also deepened the educational value of the project by offering a vivid, intuitive representation of the interplay between emotional states and cognitive performance. Future iterations could explore developing this concept using real-time simulation software or interactive visual dashboards, further bridging the gap between physiological data and actionable insights.

- **Scalability and Security:** Containerizing the system using Docker would make it more scalable and suitable for broader deployment. Implementing encryption for data in transit and at rest would ensure privacy, a critical factor for mental health applications.
- **Advanced Machine Learning Models:** Employing more sophisticated classifiers, such as convolutional neural networks (CNNs) or adversarial neural networks (ANNs), could improve prediction accuracy and adaptability to diverse user profiles.
- **Broader Physiological Insights:** Expanding the sensor suite to include additional indicators, such as respiratory rate or EEG data, could provide a more comprehensive understanding of stress and anxiety dynamics.
- **Long-Term Impact Evaluation:** Conducting longitudinal studies to evaluate the effectiveness of the device in real-world scenarios would offer valuable insights into its impact on mental health and well-being.

References

1) **Last Minute Engineers.** (2024, July 4). MAX30102 Pulse Oximeter & Heart Rate Sensor Arduino Tutorial. Retrieved from https://lastminuteengineers.com/max30102-pulse-oximeter-heart-rate-sensor-arduino-tutorial/
2) **Analog Devices.** (n.d.). *MAX30102 Datasheet*. Retrieved from https://www.analog.com/media/en/technical-documentation/data-sheets/MAX30102.pdf [1]
3) **Seeed Studio.** (n.d.). *Grove-GSR Sensor Wiki*. Retrieved from https://files.seeedstudio.com/wiki/Grove-GSR_Sensor/res/Grove-GSR_Sensor_WiKi.pdf
4) Shaukat-Jali R, van Zalk N, Boyle DE. Detecting Subclinical Social Anxiety Using Physiological Data From a Wrist-Worn Wearable: Small-Scale Feasibility Study. JMIR Form Res. 2021 Oct 7;5(10):e32656. doi: 10.2196/32656. PMID: 34617905; PMCID: PMC8532020.
5) **rs2416.** (2024, July 5). *Detecting_Social_Anxiety*. Retrieved from https://github.com/rs2416/Detecting_Social_Anxiety (File name: data_preprocessing.py)
6) **University of Queen Mary, University of London.** (n.d.). *DEAP: A Dataset for Emotion Analysis using Physiological and Audiovisual Signals*. Retrieved from https://www.eecs.qmul.ac.uk/mmv/datasets/deap/
7) ChatGPT