

**ECE 445**  
**SENIOR DESIGN LABORATORY**  
**DESIGN DOCUMENT**  
**SPRING 2024**

**A Wearable Device That Can  
Detect Mood**

**Team #32**

**Junjie Ren (junjier2)**

**Kejun Wu (kejunwu2)**

**Peidong Yang (peidong5)**

**Xinzhuo Li (xinzhuo4)**

Prof. Said Mikki  
ZJU-UIUC Institute  
Zhejiang University, Haining, Zhejiang, China  
March.26, 2024

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Problem and Solution Overview . . . . .	1
1.2	Visual Aid . . . . .	2
1.3	High-level Requirements List . . . . .	2
<b>2</b>	<b>Design</b>	<b>4</b>
2.1	Block Diagram . . . . .	4
2.2	Subsystem Descriptions . . . . .	5
2.2.1	User Interface Subsystem . . . . .	5
2.2.2	Pre-processing Subsystem . . . . .	8
2.2.3	Feature Extraction Subsystem . . . . .	12
2.2.4	Classification Subsystem . . . . .	14
2.2.5	Display Subsystem . . . . .	16
2.3	Subsystem Requirements and Verifications . . . . .	17
2.3.1	User Interface Subsystem . . . . .	17
2.3.2	Pre-processing Subsystem . . . . .	18
2.3.3	Feature Extraction Subsystem . . . . .	20
2.3.4	Mood Classification Model Subsystem . . . . .	21
2.3.5	Display Subsystem . . . . .	22
2.4	Tolerance Analysis . . . . .	23
2.4.1	Pre-processing Subsystem . . . . .	23
2.4.2	Feature Extraction Subsystem . . . . .	25
2.4.3	Sensor Accuracy and Variability . . . . .	25
2.4.4	Display Subsystem . . . . .	25
<b>3</b>	<b>Cost and Schedule</b>	<b>27</b>
3.1	Cost . . . . .	27
3.1.1	Labor . . . . .	27
3.1.2	Parts . . . . .	27
3.1.3	Total Cost . . . . .	28
3.2	Schedule . . . . .	28
<b>4</b>	<b>Ethics and Safety</b>	<b>29</b>
4.1	Ethics . . . . .	29
4.2	Safety . . . . .	29
	<b>References</b>	<b>30</b>

# 1 Introduction

## 1.1 Problem and Solution Overview

Understanding and managing our moods are integral to both our mental well-being and physical health. Mood swings, if not appropriately addressed, can lead to long-term health issues, such as headaches, asthma, heart disease, and significantly impact our daily lives, productivity, and social interactions. The necessity to diagnose and provide care for individuals experiencing mental health issues, including stress, anxiety, and depression, has never been more critical. These conditions not only affect individuals on a personal level but also pose considerable challenges in workplaces, educational institutions, and care centers, thereby necessitating innovative solutions. The rising trend of mental health issues among citizens has encouraged researchers to develop wearable devices monitoring mental health, such as self-harming activities[4].

To address this pressing need, our project introduces a pioneering wearable device, a wrist worn smart device similar to Yang's[5], designed to track and record human mood dynamics. This device employs advanced sensor technology to monitor physiological signals—such as heart rate, successive heartbeat intervals, skin temperature, and skin conductance—that are closely linked to our emotional states. By leveraging these physiological markers, our device offers an objective and precise method for mood recognition, sidestepping the subjectivity and potential inaccuracies inherent in self-reported data. Our device stands out by integrating cutting-edge machine learning algorithms that analyze physiological data to identify mood patterns. This approach not only enhances the accuracy of mood detection but also provides real-time feedback to users, enabling immediate and personalized mood management strategies. Furthermore, by offering insights into an individual's mood fluctuations, our solution empowers users to better understand their emotional well-being, encouraging proactive mental health care.

The potential applications of our technology extend far beyond personal use. By providing accurate and objective mood assessments, our device can be a valuable tool for hospitals, schools, and caring centers. It can aid in the early detection of mental health issues, monitor the effectiveness of treatment plans, and support care strategies. In educational settings, understanding students' moods can help in creating a supportive learning environment, while in the workplace, it can contribute to a healthier, more productive work culture. Our project represents a significant advancement in the intersection of technology and mental health care. By combining psychological knowledge with wearable technology, we not only aim to enhance individual well-being but also to support institutions in providing better care for those with mental health concerns or those simply seeking to improve their mood management with the support of our mood detection results. Through continuous refinement and real-world application, we are committed to making a positive impact on both individual lives and society at large.

## 1.2 Visual Aid

This visual aid part includes a figure (Figure 1) demonstrating how our project resolves the mood detection in details. The user will wear the device combined with several sensors, the Arduino PCB design will record the signals and extract features we need. Our pre-trained machine learning model will output the corresponding potential mood based on these features, and display it on computer with the help of Graphical User Interface (GUI). Ideally, by connecting the wearable device along with the user, the system is able to reflect real-time mood.

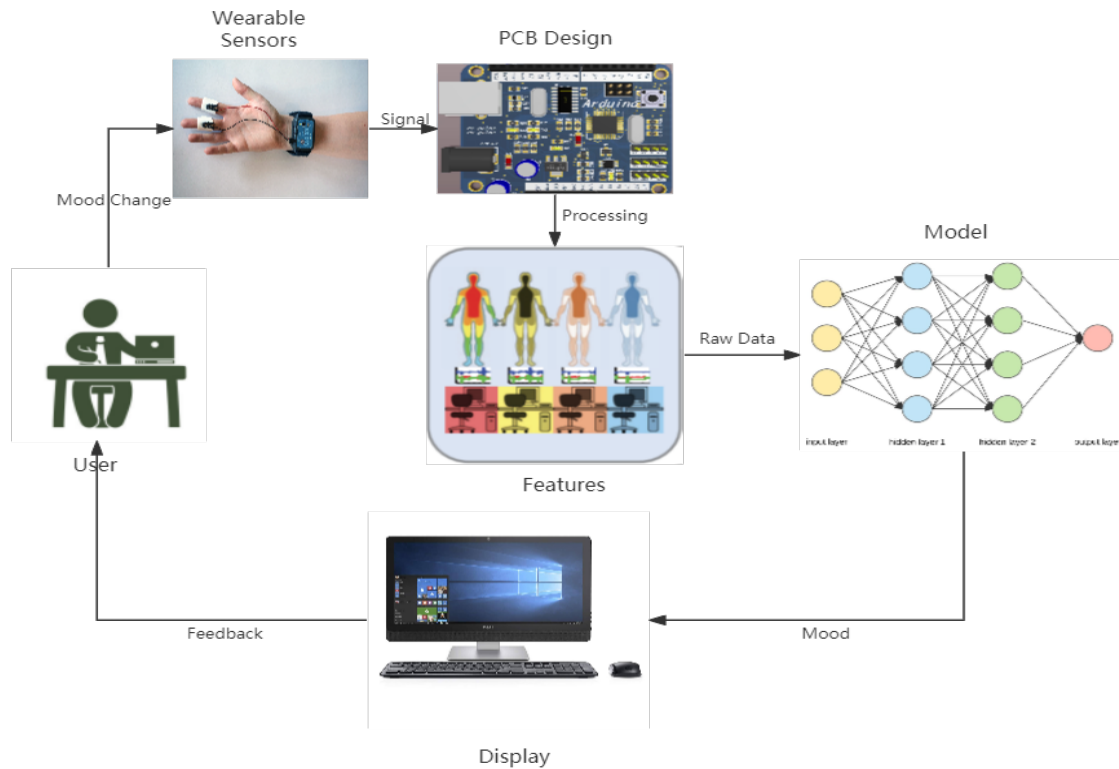


Figure 1: Visual Aid. The visual aid depicts the process where users wear integrated sensors to extract features based on the users' mood, and the machine learning model will predict mood based on features and the screen will display feedback to users

## 1.3 High-level Requirements List

- Accuracy of Mood Prediction: The system must accurately identify and classify an individual's mood states based on the data collected from wearable sensors within an acceptable error rate, ideally, the accuracy should be greater than 0.55 indicated by the state-of-the-art. This involves distinguishing between three to five emotional states and providing reliable predictions that correlate strongly with self-reported mood condition. This requirement highly depends on the accuracy of machine learning models, we will focus on data processing and algorithm

improvement to reach at least an accuracy of 0.55.

- **Real-Time Processing and Feedback:** The Mood Recognition Framework should be capable of processing data in real-time with our three sensors that can extract four features including heart rate, successive heartbeat intervals, skin temperature, and skin conductance to provide timely feedback to users by displaying on GUI system. To ensure timely feedback and reaction, the system should react to the users' input within 10 seconds and provide the result within 30 seconds on the screen. This enables immediate insight into mood states, allowing for prompt interventions or adjustments to activities and environments to improve mood. This leverages on Arduino programming, thus we must make sure the sensors can work together with Arduino and deliver data to the computer through wires and our PCB design.
- **User Interface and Display:** As a real-time mood detection wearable device, we will make sure the sensors are integrated into a user-friendly device that can fit in the wrist and fingers to produce real-time feedback on the screen. We plan to implement GUI display method on the computer reflecting users' mood with three to five kinds of emojis corresponding to the emotional states. Our system can collect data, feed it to the model, and display the result from the output of the model. This promises the feasibility of our systematic design.

## 2 Design

### 2.1 Block Diagram

The block diagram is divided into five subsystems, which can work together to realize the high-level requirements. The block diagram depicts a system designed to meet the requirements for accurate mood prediction, real-time processing and feedback, user interface and display, and user privacy and data security. The sensors collect physiological data that the system preprocesses with three steps: signal smoothing, outlier removal, and normalization to ensure quality and reliability, which is crucial for achieving the required accuracy of at least 0.3 in mood classification. The pre-processed data is then subjected to feature extraction with Time-frequency analysis, Frequency domain analysis, or nonlinear dynamic analysis and fed into trained machine learning models within the Classification Subsystem to predict mood states with desired accuracy. Real-time processing is facilitated through direct connections with an Arduino, which interfaces with the sensors for continuous data feed. User feedback is immediate via the Display Subsystem, which shows the mood predictions on a GUI, providing real-time insight and allowing for prompt user response. Additionally, by storing and processing data locally, the system aligns with privacy and security standards, ensuring user data is handled confidentially and securely.

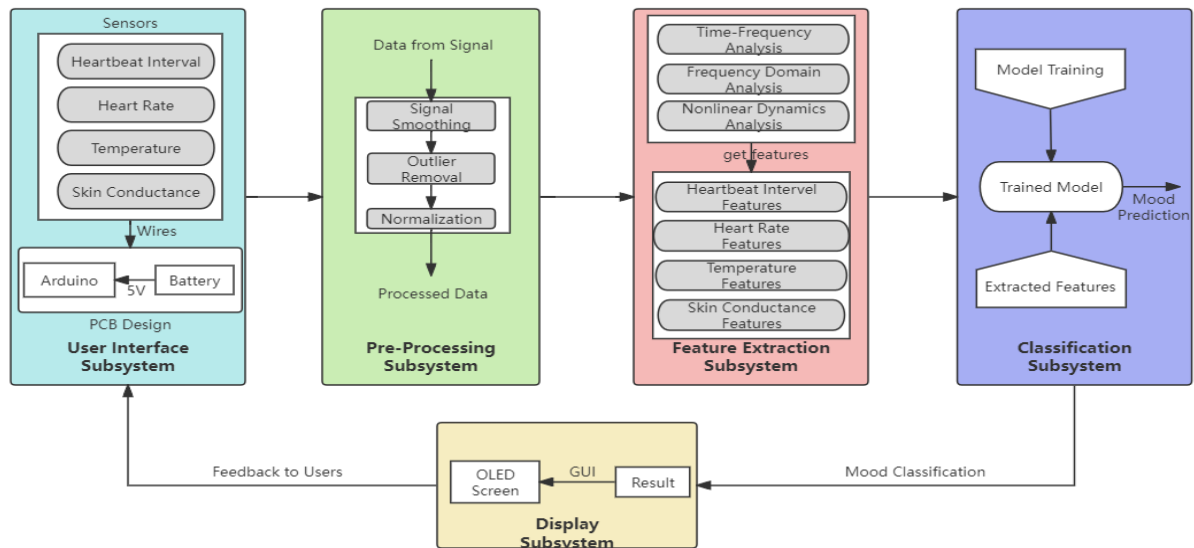


Figure 2: Block Diagram. It showcases a mood prediction system integrating sensors for data collection, pre-processing for quality (signal smoothing, outlier removal, normalization), and advanced analysis (Time-frequency, Frequency domain, nonlinear dynamics) for accurate classification. Features real-time Arduino interfacing, immediate GUI feedback, and prioritizes user privacy with secure local data handling

## 2.2 Subsystem Descriptions

### 2.2.1 User Interface Subsystem

The User Interface Subsystem is designed to monitor various physiological signals to infer the user's mood more accurately. It includes various sensors including pulse rate sensor, skin conductance sensor, and temperature sensor. These sensors and their corresponding schematics are displayed below.

As showed in Figure 3, the PulseSensor is a photoelectric reflective analog sensor for pulse heart rate measurement. The sensor board has a diameter of 16mm, a thickness of 1.6mm and a supply voltage of 3.3v or 5v. It can be worn on fingers, and the collected analog signal can be transmitted to Arduino and other micro-controllers for conversion into digital signals through wire connection. After simple calculation by Arduino micro-controller, the heart rate value can be obtained.

We plan to use this pulse rate sensor to detect two kinds of data: the heart rate and the successive heartbeat intervals. The successive heartbeat intervals (also called RR intervals) could tell us the heart rate variability (HRV). HRV is the measure of the inconsistent gaps between each heartbeat and is used as an index for different aspects of psychology. With these two variables, we can know the current heartbeat of the user, and the heartbeat variability.



Figure 3: Pulse Rate Sensor

The schematic of pulse rate sensor is showed in Figure 4. It utilizes the MCP6001 operational amplifier (op-amp), a single general-purpose op-amp, to amplify the signal from the pulse sensor and remove noise from the signal. The APDS-9008 is an ambient light photo sensor.

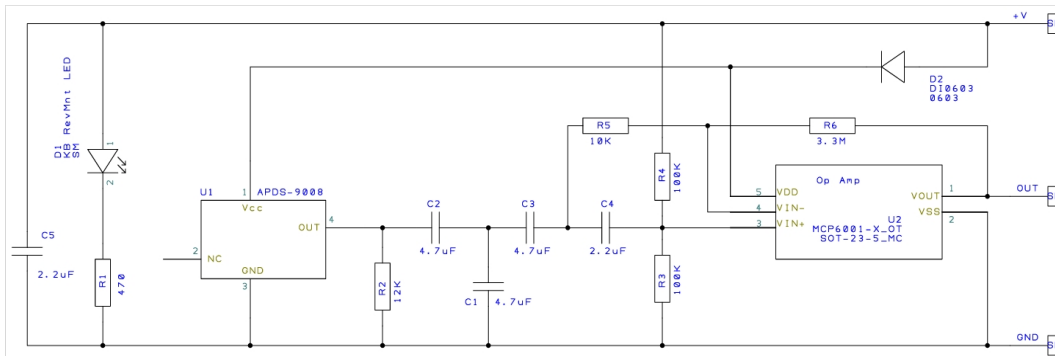


Figure 4: Pulse Rate Sensor Schematic

The skin conductance sensor's physical graph is showed in Figure 5. It has 1 mA working current. Its weight is 4g and the module's size is 34mm\*25mm. It contains a GSR module with two finger grips and a connecting cable. To be more detailed, GSR stands for Galvanic Skin Response and is a measurement of electrical conductance in the skin. Strong emotions stimulate your sympathetic nervous system, causing sweat glands to secrete more sweat. GSR can detect such strong emotions by simply connecting two electrodes to two fingers on one hand. The human body is electrically conductive, and the principle is that when you lie, get excited, or get nervous your fingers sweat, and after sweating the potential response changes, and it is the change in the electrical response that is detected by our skin conductance sensor.



Figure 5: Skin Conductance Sensor

In principle, only two electrodes need to be placed on the second and third fingers of one hand. The most common method for measuring GSR signals in psychomotor response studies is based on the exosmosis method, in which the resistance of the skin to a small current from an external source is measured. At this point, note that the most common measurement of a GSR signal is not resistance, but conductance measured in Siemens, that is, the reciprocal of resistance in ohms. Electrical conductance makes signal interpretation easier because the more sweat gland activity, the higher the skin conductance. The general schematic is listed below in Figure 6.

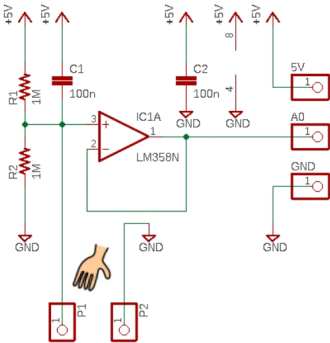


Figure 6: Skin Conductance Sensor Schematic

To measure the skin temperature, we choose the GY906-DCC infrared temperature sensor as shown in Figure 7. The sensor contains an MLX90614 series module, which is a set of general-purpose infrared temperature measurement modules. The module has the advantages of non-contact, small size, high precision and low cost. The DCC represents the sensor measurement accuracy. DCC means medical accuracy up to 280



degrees and can infrared measure temperatures up to 10cm. Generally speaking, the temperature measurement sensor measurement accuracy can be divided into contact and non-contact. Contact temperature measurement can only measure the temperature of the measured object and the temperature sensor after reaching the thermal balance, so the response time is long, and it is easily affected by the ambient temperature; Infrared temperature measurement is to determine the temperature of the object according to the infrared radiation energy of the object to be measured, without contact with the object to be measured. The infrared method to measure the temperature has the characteristics of high-temperature resolution, fast response speed, wide temperature measurement range, good stability, etc.



Figure 7: Infrared Temperature Sensor

The pins of this sensor is a little different from the above two sensors. In addition to VCC and GND, it has a SCL pin, which is the clock signal of the SMBus interface, and a SDA pin, which is the PWM or SMBus interface data signal, usually in mode from the pin output object temperature via PWM. The schematic of the infrared temperature sensor is displayed in Figure 8.

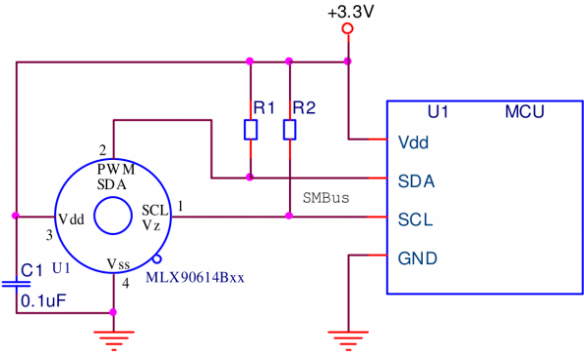


Figure 8: Infrared Temperature Sensor Schematic

With these three sensors mentioned above, the heart rate, the successive heartbeat intervals, the skin conductance, and the skin temperature data can be obtained. We will utilize the Arduino Uno to manage data collection from these sensors. All of them are connected to the Arduino Uno via its analog input pins using wire since the output type of these three sensors are analog signals. The sensors and Arduino wiring diagram is displayed below.

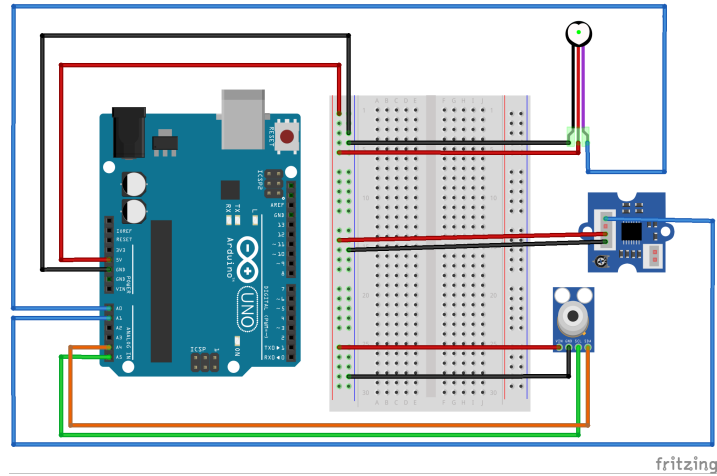


Figure 9: Sensor Arduino Wiring

This user interface sensor subsystem has close connection with the extraction subsystem. The Arduino Uno collects raw data from each sensor, performing initial processing such as filtering and normalization. Given the Uno's limited computational resources, data can be sent to an external computer for processing. For dynamic content sent to the computer, serial communication can be established between the Arduino and the computer. The extraction subsystem on the computer will then extract the features of those signals and data.

### 2.2.2 Pre-processing Subsystem

The pre-processing subsystem serves as the initial filter for the data collected from wearable sensors. Its main function is to clean the raw data by removing noise and irrelevant information that could negatively impact the accuracy of mood recognition. This involves techniques like signal smoothing, outlier removal, and normalization. The preprocessed data is then a more reliable representation of the user's physiological states. The data pre-processing is finished on software level.

This subsystem interfaces directly with the feature extraction subsystem, providing it with high-quality input data. The quality of pre-processing directly affects the efficiency and effectiveness of the subsequent feature extraction stage, making this a critical part of the mood recognition process.

Pre-processing subsystem contains 3 successive parts. They are Signal Smoothing, Outlier Removal and Normalization.

**Signal Smoothing** is the first step in the data preprocessing pipeline aimed at reducing noise and fluctuations in the raw sensor data. It involves applying various filtering techniques to achieve a smoother representation of the signal while preserving important features. In signal smoothing, we use the moving average method that averages neighboring data points. The moving average filter is a widely used technique in signal processing for its simplicity and effectiveness in attenuating high-frequency components while preserving the underlying signal trends.

The moving average filter operates by computing the mean of a sliding window of consecutive data points along the signal. The size of the window, often referred to as the "window length" or "kernel size," determines the degree of smoothing. A larger window results in greater smoothing but may blur out rapid changes in the signal.

The moving average of a signal  $x(t)$  at time  $t$  with a window size  $N$  is computed as:

$$\bar{x}(t) = \frac{1}{N} \sum_{i=t-N+1}^t x(i) \quad (1)$$

Where:

- $\bar{x}(t)$  is the moving average of the signal at time  $t$ ,
- $x(i)$  is the value of the signal at time  $i$ ,
- $N$  is the size of the moving window.

The above equation calculates the average of the signal values within the window centered at time  $t$ . As the window slides along the signal, each data point is replaced with the average of itself and its neighboring points, resulting in a smoothed signal.

Figure 10 shows the algorithm for moving average filter.

---

**Algorithm 1** Moving Average Filter

---

```

1: function MOVINGAVERAGEFILTER(data, window_size)
2:   smoothed_data  $\leftarrow$  []
3:   for  $i$  from 0 to  $\text{len}(\textit{data}) - \textit{window\_size}$  do
4:     window_sum  $\leftarrow$  0
5:     for  $j$  from  $i$  to  $i + \textit{window\_size} - 1$  do
6:       window_sum  $\leftarrow$  window_sum + data[ $j$ ]
7:     end for
8:     smoothed_data.append(window_sum/window_size)
9:   end for
10:  return smoothed_data
11: end function

```

---

Figure 10: Moving Average Filter Algorithm. It is utilized in signal processing to smooth out fluctuations in data by averaging neighboring data points within a specified window size.

We use Signal-to-Noise Ratio (SNR) to quantify the ratio of signal power to noise power present in a signal. A higher SNR indicates a stronger, more dominant signal compared to the noise. The formula to calculate SNR is as follows:

$$\text{SNR} = \frac{\text{Signal Power}}{\text{Noise Power}} \quad (2)$$

**Outlier removal** is the second step in the data preprocessing pipeline aimed at identifying and eliminating data points that deviate significantly from the majority of the dataset. It employs statistical methods to detect and remove spurious outliers.

We apply modified z-score statistical method to detect outliers, and then we remove outliers by replacing them with the average of two adjacent points. The formula of modified z-score statistical method is as follows:

$$M_i = \frac{0.6745 \cdot (X_i - \tilde{X})}{\text{MAD}} \quad (3)$$

Where:

- $M_i$  is the modified z-score for the  $i$ th data point,
- $X_i$  is the  $i$ th data point,
- $\tilde{X}$  is the median of the dataset,
- MAD is the median absolute deviation of the dataset.

A data point is regarded as outlier if

$$|M_i| > \text{threshold} = 3 \quad (4)$$

Figure 11 shows the algorithm for modified Z-score method.

---

**Algorithm 2** Modified Z-score Method for Outlier Removal

---

```

1: function MODIFIEDZSCORE(data, threshold)
2:   median ← medianofdata
3:   mad ← medianabsolutedeviationofdata
4:   for  $i$  from 0 to  $\text{len}(\textit{data}) - 1$  do
5:     modified_z_score ←  $\frac{0.6745 \cdot (\textit{data}[i] - \textit{median})}{\textit{mad}}$ 
6:     if modified_z_score > threshold then
7:       Remove data[ $i$ ] from data
8:     end if
9:   end for
10:  return data
11: end function

```

---

Figure 11: Modified Z-score Method for Outlier Removal. It is used to identify and eliminate outliers from a dataset by measuring deviations from the median in terms of median absolute deviation (MAD).

**Normalization** is the final step in the data preprocessing pipeline aimed at standardizing the range of features in the dataset. It scales the data to a predefined range or distribution, typically between 0 and 1 or with a mean of 0 and a standard deviation of 1.

Normalization ensures that the input features have consistent scales, facilitating the training of machine learning models and improving their generalization performance. It contributes to the overall design by making the input data comparable and reducing the impact of scale differences on the classification results.

We use min-max normalization and Z-score normalization. The formula of min-max normalization and Z-score Normalization are as follows, respectively:

$$X_{\text{normalized}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (5)$$

$$X_{\text{normalized}} = \frac{X - \mu}{\sigma} \quad (6)$$

Where:

- $X$  is the original data point,
- $\mu$  is the mean of the dataset,
- $\sigma$  is the standard deviation of the dataset,
- $X_{\min}$  is the minimum value in the dataset,
- $X_{\max}$  is the maximum value in the dataset,
- $X_{\text{normalized}}$  is the normalized data point.

Figure 12 and 13 shows the algorithm for min-max normalization and z-score normalization, respectively.

---

**Algorithm 3** Min-Max Normalization

---

```

1: function MINMAXNORMALIZATION(data)
2:   min_value ← minimumvalueofdata
3:   max_value ← maximumvalueofdata
4:   for i from 0 to len(data) - 1 do
5:     data[i] ←  $\frac{\text{data}[i] - \text{min\_value}}{\text{max\_value} - \text{min\_value}}$ 
6:   end for
7:   return data
8: end function

```

---

Figure 12: Min-Max Normalization. It scales the values of a dataset to a predefined range, typically between 0 and 1, by subtracting the minimum value and dividing by the range.

---

**Algorithm 4** Z-score Normalization

---

```
1: function ZSCORENORMALIZATION(data)
2:   mean ← meanofdata
3:   std_dev ← standarddeviationofdata
4:   for i from 0 to len(data) – 1 do
5:     data[i] ←  $\frac{\text{data}[i] - \text{mean}}{\text{std\_dev}}$ 
6:   end for
7:   return data
8: end function
```

---

Figure 13: Z-score Normalization. It standardizes the values of a dataset by subtracting the mean and dividing by the standard deviation.

### 2.2.3 Feature Extraction Subsystem

The feature extraction subsystem’s function is to distill key indicators from the preprocessed data that are relevant for mood recognition [2]. It translates raw sensor data into a set of features that reflect the user’s physiological and behavioral patterns associated with different moods. For instance, heart rate variability may indicate stress levels, while temperature changes could relate to physical activity or emotional arousal. This subsystem must extract features that are both informative and discriminative for different mood states.

It interfaces with both the pre-processing subsystem, from which it receives the cleaned data, and the mood classification model, to which it sends the extracted features. The success of this block is measured by its ability to provide meaningful features that improve the classification performance of the mood recognition model.

The feature extraction subsystem consists of 2 essential parts, which are Time-Frequency Analysis and Frequency Domain Analysis. They share the same purpose of extracting features.

- 1) Time-frequency analysis is a method used to analyze signals in both the time and frequency domains simultaneously. It allows us to understand how the frequency content of a signal changes over time, which can be particularly useful for analyzing non-stationary signals such as physiological data. We apply Short-Time Fourier Transform (STFT). The formula of STFT is as follows:

$$X(m, \omega) = \sum_{n=-\infty}^{\infty} x(n) \cdot w(n - m) \cdot e^{-j\omega n} \quad (7)$$

Where:

- $X(m, \omega)$  is the STFT of the signal at time  $m$  and frequency  $\omega$ ,

- $x(n)$  is the input signal,
- $w(n - m)$  is the window function centered at time  $m$ ,
- $\omega$  is the angular frequency.

Figure 14 shows the algorithm for Short-Time Fourier Transform.

---

**Algorithm 5** Short-Time Fourier Transform (STFT)

---

```

1: function STFT(data, window_size, overlap)
2:   stft_data  $\leftarrow$  []
3:   for  $i$  from 0 to  $\text{len}(\textit{data}) - \textit{window\_size}$  step  $\textit{overlap}$  do
4:     windowed_data  $\leftarrow$  data[ $i : i + \textit{window\_size}$ ]
5:     stft  $\leftarrow$  FourierTransform(windowed_data)
6:     stft_data.append(stft)
7:   end for
8:   return stft_data
9: end function

```

---

Figure 14: Short-Time Fourier Transform (STFT), which transforms input signal data into its frequency-domain representation by dividing it into short segments and applying the Fourier Transform to each segment separately

- 2) Frequency domain analysis is a technique used to analyze signals in terms of their frequency content. It provides insights into the distribution of signal power across different frequency components. We apply Power Spectral Density (PSD), and Frequency Band Analysis strategy. The formula of PSD is as follows:

$$P_{xx}(f) = \lim_{T \rightarrow \infty} \frac{1}{T} \left| \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) e^{-j2\pi ft} dt \right|^2 \quad (8)$$

Where:

- $P_{xx}(f)$  is the PSD of the signal,
- $x(t)$  is the input signal,
- $f$  is the frequency variable.

The formula of Frequency Band Analysis is as follows:

$$P_{\text{band}} = \int_{f_{\min}}^{f_{\max}} P_{xx}(f) df \quad (9)$$

Where:

- $P_{\text{band}}$  is the power within a specific frequency band,
- $P_{xx}(f)$  is the PSD of the signal,

- $f_{\min}$  is the lower bound of the frequency band,
- $f_{\max}$  is the upper bound of the frequency band.

Figure 15 shows the algorithm for Power Spectral Density.

---

**Algorithm 6** Power Spectral Density (PSD) Estimation

---

```

1: function PSDESTIMATION(stft_data)
2:   psd  $\leftarrow$  []
3:   for i from 0 to len(stft_data) do
4:     psd_point  $\leftarrow$  ComputePSD(stft_data[i])
5:     psd.append(psd_point)
6:   end for
7:   return psd
8: end function

```

---

Figure 15: Power Spectral Density (PSD) algorithm, which provides insights into the distribution of signal power across different frequency components by computing the squared magnitude of the Fourier Transform of the signal.

### 2.2.4 Classification Subsystem

This subsystem is the heart of the mood recognition framework, where the actual classification of the user’s mood occurs. It employs machine learning algorithms to interpret the features extracted from the user’s data and classify them into mood states such as happy, sad, stressed, or relaxed. This subsystem may utilize a variety of modeling techniques, including supervised learning, to train models on labeled mood data.

The mood classification model interfaces with the feature extraction subsystem, receiving features as input, and with the display subsystem, providing the mood predictions. Its performance is critical, as it directly determines the accuracy and reliability of the mood recognition the framework provides.

The mood classification subsystem consists of 2 essential parts, which are Random Forest algorithm and Extreme Gradient Boosting(XGBoost) algorithm. They share the same purpose of extracting features and they complement with each other. We finally balance the trade-off between these two algorithms to decide which one to use, considering the classification results of valence, arousal, stress and disturbance.

- 1) Random Forest is an ensemble learning method that combines multiple decision trees to improve classification accuracy. It’s robust to overfitting and works well for high-dimensional feature spaces.



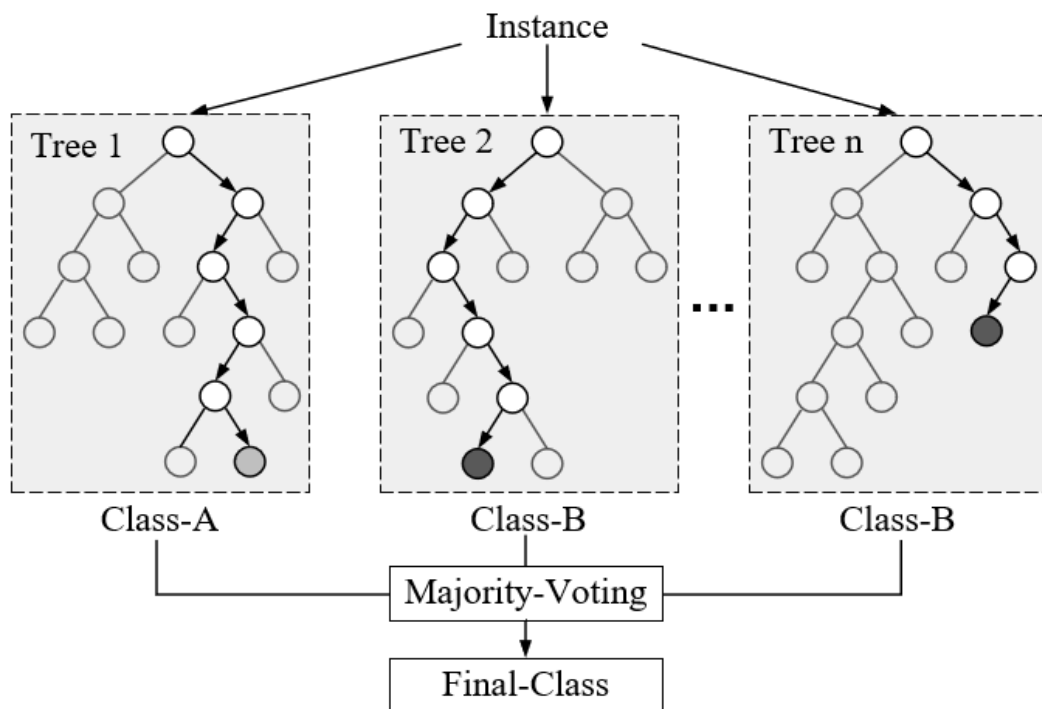


Figure 16: Random Forest Model[A powerful ensemble learning method that combines multiple decision trees to enhance classification accuracy and mitigate overfitting.]

- 2) XGBoost is an efficient and scalable implementation of gradient boosting decision trees. It builds a series of decision trees sequentially, where each tree corrects the errors made by the previous ones, leading to improved overall performance.

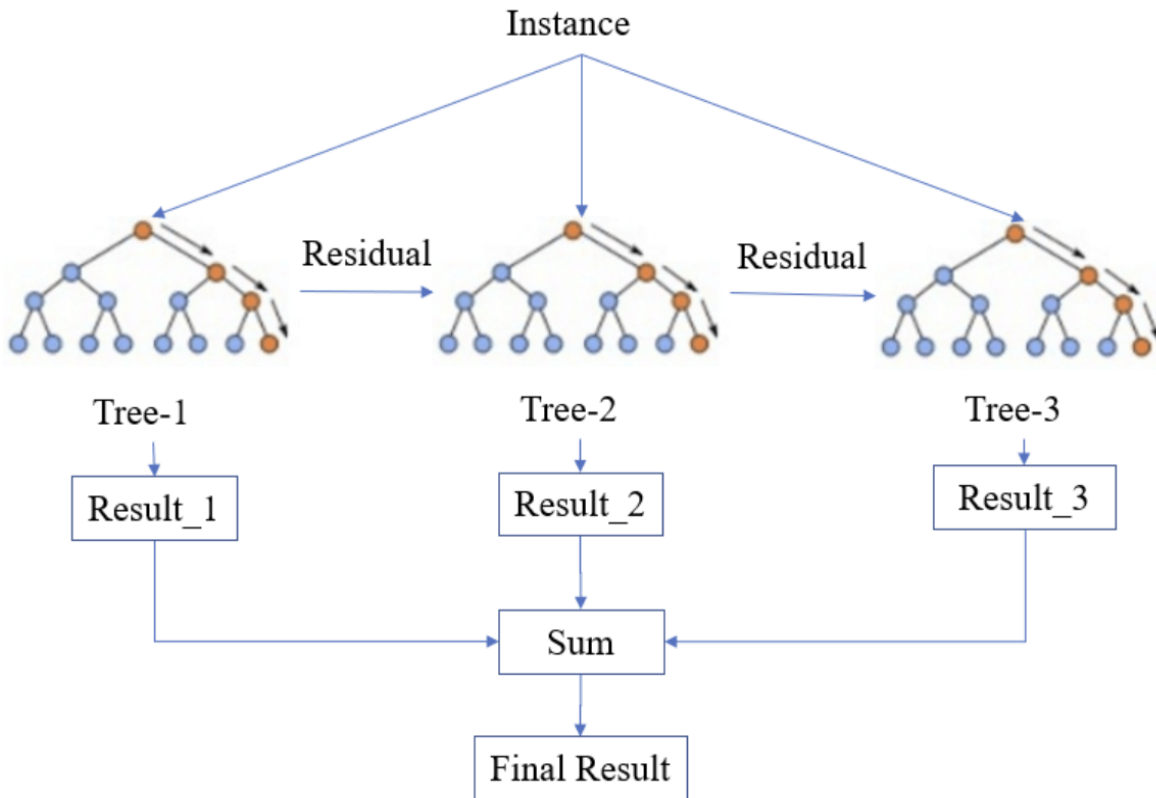


Figure 17: XGBoost Model[A highly efficient implementation of gradient boosting decision trees, sequentially building trees to correct errors and improve performance.]

### 2.2.5 Display Subsystem

The display subsystem is designed to display the detected mood and corresponding suggestions to the user in an intuitive and engaging way, which is crucial for enhancing the user experience. We plan to display the mood states in text ("Happy", "Sad", etc.) along with relevant icons or emojis, in accordance with the specific mood detected by classification subsystem. To further strengthen the feedback, we can also display tips or recommendations based on the detected mood. For example, some encouraging sentences can be placed on the screen after detecting that the user is sad. We can also advise some activities to users, such as hanging out or playing video games, to relax and please users themselves. Overall, this subsystem enhances user interaction by offering real-time feedback to improve or maintain people's emotional well-being.

We plan to implement Graphical User Interface (GUI) in Python to achieve this subsystem. The computer will firstly output the results of detected mood after utilizing Random Forest algorithm in classification subsystem. Then the displaying subsystem will receive the processing results and give corresponding feedback. Designing a GUI that incorporates mood detection and provides appropriate suggestions requires careful consideration of user interaction, visual presentation, and integration with the mood detection subsystem. Here are components of our GUI:

- 1) Display Components: Design areas where the detected mood and suggested activities or content will be displayed.
- 2) Visual Elements: Use visual cues such as colors, icons and images to make the GUI visually appealing. We will utilize emojis to represent 3 to 5 kinds of moods.
- 3) Integration with Mood Detection Subsystem: Implement functionality (messaging protocols such as WebSocket, or direct function calls) to communicate with the mood detection subsystem to retrieve the user's current mood.

## 2.3 Subsystem Requirements and Verifications

### 2.3.1 User Interface Subsystem

The User Interface subsystem consists of a PCB design with three sensors and Arduino. Its responsibility is to obtain physiological data from our users and transmit them to the computer for further processing. The requirements and verifications of the user interface sensor system are listed in Table 1.

Table 1: R&V table of User Interface

Requirements	Verifications
The user interface sensor subsystem is capable of obtaining at least four types of physiological data simultaneously. It should process the primary data and transmit the data to an external computer for further processing using wire.	All types of data could be seen simultaneously on the computer's screen instead of using the sensor to test one by one.
The sensor must detect pulse rate changes within 10 seconds and maintain accuracy within $\pm 2$ bpm.	<ol style="list-style-type: none"> <li>1.Utilize the professional medical heart rate instrument to test the pulse rate and regard it as a standard.</li> <li>2.Measure the detected pulse rate with our sensor and compare it to the standard tested before.</li> <li>3.Compare the standard data with the detected pulse rate and ensure our results are within the accuracy range of <math>\pm 2</math> bpm.</li> </ol>
The accuracy of the skin conductance sensor's output is within the range of $\pm 5\%$ .	<ol style="list-style-type: none"> <li>1.Prepare a circuit with known resistances that simulate skin conductance levels (e.g., 50k ohm to 10m ohm).</li> <li>2.Connect the GSR sensor to the calibration circuit and measure the conductance at each resistance level.</li> <li>3.The output of the sensor would be recorded and the percentage error would be calculated.</li> </ol>

The infrared temperature sensor must stabilize to within $\pm 0.2^{\circ}\text{C}$ of the target temperature in less than 10 seconds.	<ol style="list-style-type: none"> <li>1. Prepare an infrared thermometer.</li> <li>2. Our sensor and the reference thermometer are put in the same environment. Both of them measure the skin temperature.</li> <li>3. After comparison, if the readings from the infrared sensor consistently match the reference thermometer within <math>\pm 0.2^{\circ}\text{C}</math>, the accuracy of the SKT sensor could be proved.</li> </ol>
---	---

### 2.3.2 Pre-processing Subsystem

The data pre-processing subsystem consists of 3 successive parts. They are Signal Smoothing, Outlier Removal and Normalization.

- 1) Signal Smoothing is a technique used to remove high-frequency noise from a signal while preserving its underlying trends or features. See Table 2 for its Requirements and Verifications.

Table 2: R&V table of Signal Smoothing

Requirements	Verifications
Smoothed signals should exhibit a significant reduction in noise compared to the original signal, with a minimum increase in SNR of 10 dB.	<ol style="list-style-type: none"> <li>1. Apply a random noisy signal from K-EmoPhone dataset to the smoothing algorithm.</li> <li>2. Using the software SciPy from Python language, measure the signal-to-noise ratio (SNR) of the original and smoothed signals by power spectral density method.</li> <li>3. See if the smoothed signal exhibits a minimum increase in SNR of 10 dB compared to the original signal.</li> <li>4. Repeat the above steps and ensure that the tested dataset covers at least 1/3 of the standard dataset, verifying the smoothed signals exhibit consistent noise reduction and preservation of signal features.</li> </ol>
The mean squared error (MSE) between the original and smoothed signals should be below the threshold of $0.1 \times \text{noise variance}$ , indicating minimal distortion.	<ol style="list-style-type: none"> <li>1. Calculate the mean squared error (MSE) between the original and smoothed signals using Math Solver, a powerful software calculator.</li> <li>2. See if the MSE between the original and smoothed signals is below the threshold of <math>0.1 \times \text{noise variance}</math>, indicating acceptable distortion.</li> <li>3. Repeat the above steps for all signals tested with SNR, verifying the smoothing algorithm does not introduce significant distortion.</li> </ol>

- 2) Outlier removal is a crucial step in data preprocessing aimed at identifying and eliminating data points that deviate significantly from the rest of the dataset. See Table 3 for its Requirements and Verifications.

Table 3: R&V table of Outlier Removal

Requirements	Verifications
The outlier removal process should effectively identify data points that deviate significantly from the rest of the dataset.	1. Apply the outlier detection algorithm to a test dataset containing known outliers. Compare the identified outliers with the ground truth. 2. See if identified outliers match the known outliers in the test dataset with an accuracy degree higher than 85%.
The outlier removal process should effectively remove data points that deviate significantly from the rest of the dataset.	1. Remove the identified outliers from the test dataset. Calculate the statistical properties (e.g., mean, standard deviation) of the dataset before and after outlier removal. 2. After outlier removal, see if the statistical properties of the dataset align more closely with the underlying distribution of the data. Specifically, see if the mean and standard deviation are closer to the values calculated from a clean dataset.

- 3) Normalization is used to rescale the features of a dataset to a standard range, typically between 0 and 1 or with a mean of 0 and a standard deviation of 1. See Table 4 for its Requirements and Verifications.

Table 4: R&V table of Normalization

Requirements	Verifications
All features in the dataset should be uniformly scaled to prevent dominance by features with larger scales.	<ol style="list-style-type: none"> <li>1. Apply the normalization algorithm to the dataset and examine the range of normalized values for each feature</li> <li>2. Check if all normalized feature values fall within the specified range (e.g., 0 to 1 for min-max normalization, mean of 0 and standard deviation of 1 for z-score normalization).</li> <li>3. Calculate the minimum and maximum values of each normalized feature. If all normalized values fall within the specified range, the requirement for uniform scaling is met.</li> </ol>
The normalization process should not introduce bias or distortions in the data.	<ol style="list-style-type: none"> <li>1. Calculate the mean and standard deviation of the normalized dataset.</li> <li>2. Check if the mean of the normalized dataset is close to 0, and the standard deviation is close to 1 for z-score normalization. For min-max normalization, check if the mean is within a small tolerance of the midpoint of the range, and the standard deviation is approximately equal to half the range.</li> <li>3. If the mean is close to 0 and the standard deviation is close to 1 (or within specified tolerances), the requirement for bias and distortion is met.</li> </ol>

### 2.3.3 Feature Extraction Subsystem

The feature extraction subsystem consists of 3 essential parts. They are Time-Frequency Analysis, Frequency Domain Analysis, and Nonlinear Dynamics Analysis.

Since the above 3 strategies share the same purpose of extracting features and they complement with each other, their methods for requirements and verifications could be identical, as shown in the Table 5.

Table 5: R&V table of feature extraction

Requirements	Verifications
<p><b>Accuracy:</b> The feature analysis should accurately identify the dominant frequency components of the signal with an error margin of less than 15%</p>	<ol style="list-style-type: none"> <li>1. Generate synthetic signals with known dominant frequency components, from the K-EmoPhone dataset.</li> <li>2. Apply the feature analysis method to the synthetic signals and extract the dominant frequency components.</li> <li>3. Compare and calculate the percentage error between the identified dominant frequencies and the known frequencies, and see if the error margin is less than 15% .</li> </ol>
<p><b>Resolution:</b> The feature analysis should have a frequency resolution of at least 10 Hz to distinguish between closely spaced frequency components.</p>	<ol style="list-style-type: none"> <li>1. Generate synthetic signals with closely spaced frequency components, from the K-EmoPhone dataset.</li> <li>2. Apply the feature analysis method to the synthetic signals and examine the frequency spectrum. Measure the distance between adjacent frequency peaks to determine the frequency resolution.</li> <li>3. Verify that the feature analysis method can resolve adjacent frequency components with a separation of at least 10 Hz.</li> </ol>

### 2.3.4 Mood Classification Model Subsystem

The mood classification model subsystem consists of 2 essential supervised learning algorithms. They are Random Forest and Extreme Gradient Boosting (XGBoost).

Since the above 2 algorithms share the same purpose of classifying mood and they complement with each other, their requirements and verifications methods could be identical, as shown in the Table 6.

Table 6: R&V table of Mood Classification

Requirements	Verifications
The mood classification model should achieve a classification accuracy of at least 55% when predicting mood states based on the given psychological data.	<ol style="list-style-type: none"> <li>1.Using supervised learning algorithm like Random Forest and XGBoost, train the classification model based on the labelled data K-EmoPhone dataset.</li> <li>2.Use the trained model to predict mood states for the samples in the testing dataset. Compare the predicted mood states with the ground truth labels to assess classification accuracy.</li> <li>3.Calculate the classification accuracy as the percentage of correctly predicted mood states out of the total number of samples in the testing dataset. See if it is higher than the desired 55% threshold.</li> </ol>

### 2.3.5 Display Subsystem

We will utilize Graphical User Interface (GUI) in Python to display the detected mood and corresponding suggestions to users. Designing a display via a GUI involves various considerations to ensure usability, functionality, and aesthetic appeal. Providing appropriate suggestions to users based on the detected mood involves understanding the user’s emotional state and offering content, activities, or interventions that are likely to resonate positively with them. The requirements and verifications of the display subsystem are listed in Table 7.

Table 7: R&V table of Display Subsystem

Requirements	Verification
The interface must integrate with Mood Classification Model. After the model generates the detected mood based on our input, the interface should give real-time feedback.	1.Once the mood output is stored in a txt file, the GUI can open and read it, and then update the relevant GUI components to reflect the detected mood. This could involve changing colors or icons displaying four mood classifications with corresponding emojis.
The interface should provide appropriate suggestions to users based on the detected mood to make positive impacts on users.	<ol style="list-style-type: none"> <li>1.Consult professional psychology references for professional advice related to each detected mood.</li> <li>2.Offer a variety of suggestions to cater to different preferences and interests. Include options for relaxation, entertainment, socializing, and self-care to accommodate diverse user needs.</li> </ol>



<p>The interface has aesthetic design elements including color schemes, typography, icons, and visual hierarchy to enhance usability and appeal.</p>	<ol style="list-style-type: none"> <li>1.Design the layout of GUI, considering the placement and arrangement of components to ensure clarity and usability. Use principles of visual hierarchy and proximity to organize components logically.</li> <li>2.Choose appropriate colors, emojis, and other visual elements to enhance the aesthetic appeal and usability of the GUI. Consider using different colors or visual cues to represent different mood categories.</li> <li>3.Define how users will interact with the GUI components to input or select the mood information. Ensure intuitive interaction patterns and provide appropriate feedback to users.</li> </ol>
--	--

## 2.4 Tolerance Analysis

### 2.4.1 Pre-processing Subsystem

Outlier Removal presents the greatest risk within this block, as it may lead to the inadvertent removal of valuable data or the retention of noisy data if not properly calibrated. A statistical analysis can assess the likelihood of true outliers versus false outliers based on the historical variability of the data. To mitigate this risk, it is imperative to employ robust statistical methods, such as the median absolute deviation (MAD), which is less sensitive to extreme values compared to the standard deviation. As mentioned previously, we set the MAD threshold empirically to 3 to identify outliers.

A thorough experiment regarding outlier identification has been conducted on the participant P01’s data, and Figure 18 shows a satisfying result.

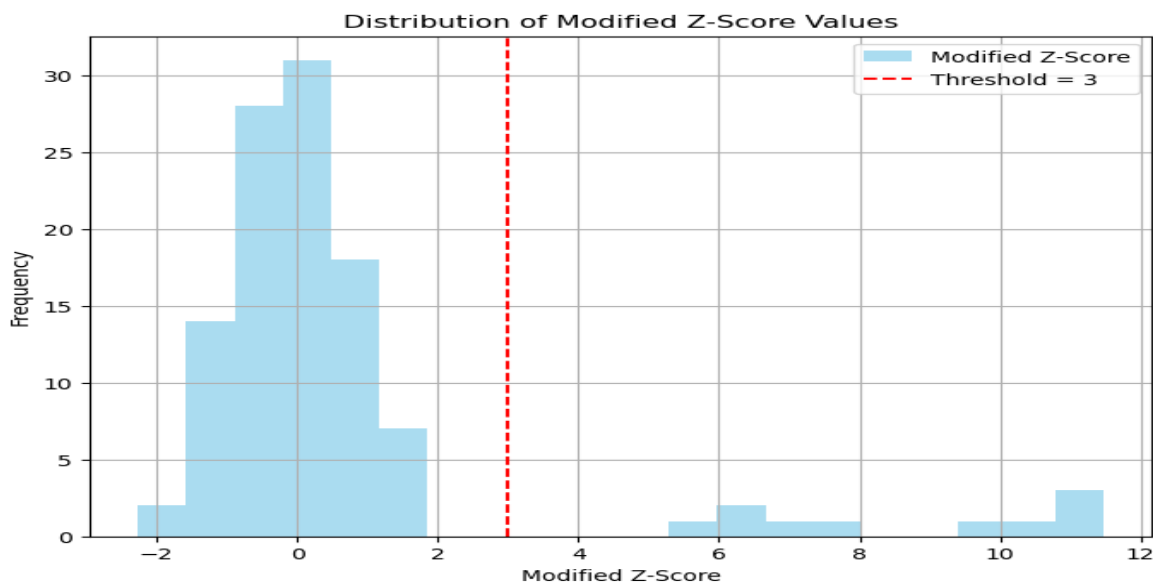


Figure 18: Modified Z-Score to verify the Outlier Removal result

Mathematically, the effectiveness of outlier removal can be evaluated by quantifying the change in the standard deviation of the dataset before and after the removal of outliers. Let's denote the standard deviation before outlier removal as  $\sigma_{\text{original}}$  and after outlier removal as  $\sigma_{\text{removed}}$ . The percentage change in the standard deviation ( $\Delta\sigma$ ) can be calculated using the formula:

$$\Delta\sigma = \frac{\sigma_{\text{original}} - \sigma_{\text{removed}}}{\sigma_{\text{original}}} \times 100\% \quad (10)$$

The effectiveness of the outlier removal process can then be assessed based on the magnitude of  $\Delta\sigma$ . For example, if  $\Delta\sigma$  is below a threshold of 10%, it indicates that the removal of outliers has effectively reduced the variability in the dataset without significantly altering its overall distribution.

To ensure success, the number of data points discarded during outlier removal should not exceed a certain threshold of 15% relative to the total dataset size. This threshold helps maintain the integrity of the dataset while still eliminating outliers. Additionally, conducting sensitivity analysis by varying the threshold can provide insights into the robustness of the outlier removal process and help optimize its parameters for different datasets.

We should also consider de-noising and extracting data accurately from the signal during the processing process, we will mathematically verify our data relying on what we mentioned at the R&V part, leveraging on advanced signal processing methods and our designed process including signal smoothing, outlier removal, and normalization to minimize the inaccuracy. At least 95% of the tested data should be within 5% of the average of all data in the dataset used for training.

### 2.4.2 Feature Extraction Subsystem

We can quantify the feature extraction tolerance using the root mean square error (RMSE) between the extracted features and the ground truth or expected values.

**Root Mean Square Error (RMSE):** The root mean square error measures the square root of the average of the squares of the errors between the extracted features ( $\hat{f}_i$ ) and the ground truth or expected values ( $f_i$ ). Mathematically, it is given by:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{f}_i - f_i)^2} \quad (11)$$

RMSE provide a quantitative measure of the deviation between the extracted features and the ground truth or expected values. A lower value of RMSE indicates better performance of the feature extraction method in terms of accuracy and reliability.

To ensure the success of feature extraction, the RMSE should be kept below a threshold of 15%.

### 2.4.3 Sensor Accuracy and Variability

As for the sensor accuracy, each sensor has a known accuracy interval (e.g.,  $\pm 2$ bpm for heart rate,  $\pm 5\%$  for skin conductance). The data obtained by the sensors are also determined by various factors. For example, the variation in pulse detection accuracy can be due to motion artifacts, skin tone, and ambient light, and the variation in skin conductivity readings can be due to factors such as skin moisture, temperature, and sensor placement. In this case, we'll be focusing on all the physiological information that a person has at rest. We will use signal-to-noise ratio (SNR) to quantify the heart rate sensor's ability to distinguish pulse signals from noise under different conditions. We will assess the acceptable range of SNR where the heart rate can be accurately detected. This tolerance analysis emphasizes the importance of robust sensor selection.

Considering the noise and other influencing factors on the sensors, we would decrease the influence of the environment as much as possible, for example, keeping the skin clear, keeping the users still, and decreasing weather influences by conducting the experiment indoors. With these methods, we wish to minimize the influences by other factors to the sensors.

### 2.4.4 Display Subsystem

For the GUI designed to provide suggestions to users based on their detected mood, a tolerance analysis involves evaluating various aspects of the interface to ensure that it remains usable, functional, and visually appealing despite potential variations or deviations. Using GUI is also crucial for our design for the purpose of displaying emojis, suggestions, and other potential feedback to the users. A proper integration between the model and display system is needed, which we decided to use Python to

build the connection. To ensure timely feedback and reaction, the system should react to the users' input within 10 s and provide the result within 30 s on the screen.

Firstly, the crucial thing is that the suggestions provided to users may be misleading or irrelevant due to the accuracy of mood classification model. Moreover, we need to ensure that the layout remains visually appealing and functional despite changes in screen size, resolution, or content length. It is significant to determine acceptable tolerances for spacing, alignment, and content organization to maintain readability and usability.

## 3 Cost and Schedule

### 3.1 Cost

#### 3.1.1 Labor

Refer to the sample and past senior design projects, we estimate our salary to be \$40/hour, 10 hours/week for each group member. The estimated salary is listed in Labor Cost Table 8.

Table 8: Labor Cost. The table shows the member name in the senior design project (Member), the hourly salary of member (Hourly Salary), the working hours in the project per person (Working Hours) and the total salary per person (Total).

Member	Hourly Salary	Working Hours	Total
Junjie Ren	\$40	$10 \times 14 = 140h$	$\$40 \times 140 = \$5600$
Peidong Yang	\$40	$10 \times 14 = 140h$	$\$40 \times 140 = \$5600$
Xinzhuo Li	\$40	$10 \times 14 = 140h$	$\$40 \times 140 = \$5600$
Kejun Wu	\$40	$10 \times 14 = 140h$	$\$40 \times 140 = \$5600$
Sum			\$22400

#### 3.1.2 Parts

Our parts and manufacturing prototype costs are estimated to be \$39.4 in total. See Parts Cost Table 9 in detail.

Table 9: Parts Cost. The table shows the parts description (Description), the manufacturer of the parts (Manufacturer), the quantities needed in our project (Quantity), the unit cost of the parts (Cost/Unit) and the total cost of the the parts (Total Cost).

Description	Manufacturer	Quantity	Cost/Unit	Total Cost
Pulse Sensor (MDL0025)	Sichiray	1	\$5.5	\$5.5
Grove Galvanic Skin Response (GSR)	Sichiray	1	\$6.3	\$6.3
Infra Red Ther- mometer (GY-906-DCC)	Melexis	1	\$4	\$4
Arduino uno	Sichiray	1	\$23.6	\$23.6
Sum				\$39.4

### 3.1.3 Total Cost

The total cost of our senior design project is:

$$\$22400 + \$39.4 = \$22439.4$$

## 3.2 Schedule

The weekly schedule table is given in table 10.

Table 10: Weekly schedule table. The table shows different weeks (Date) and individual schedule for each group member (Junjie Ren, Peidong Yang, Xinzhuo Li and Kejun Wu) in different weeks.

Date	Junjie Ren	Peidong Yang	Xinzhuo Li	Kejun Wu
3.25-3.31	Research the labeled data	Test the function of sensors	Research appropriate machine learning methods	Test the working of Arduino
4.1-4.7	Pre-processing the labeled data	Measure some physiological data by using the hardware parts	Write machine learning code	Measure some physiological data by using the hardware parts
4.8-4.14	Write and debug machine learning code	Import the measured data into computer	Write and debug machine learning code	Import the measured data into computer
4.15-4.21	Evaluate the accuracy of our trained model	use GUI in python to display possible outcomes	Improve our trained model to make it more accurate	Integrate other possible sensors to Arduino
4.22-4.28	Improve and debug machine learning model	Integrate all hardware parts as a wearable device	Improve and debug machine learning model	Merge measured data with machine learning model
4.29-5.5	Test and improve the mood classification model	Test and improve the display subsystem	Test and improve the mood classification model	Test and improve the user interface subsystem
5.6-5.12	Mock Demo	Mock Demo	Mock Demo	Mock Demo
5.13-5.19	Final Demo and Final Report	Final Demo and Final Report	Final Demo and Final Report	Final Demo and Final Report
5.20-5.26	Final Report and Team Evaluation	Final Report and Team Evaluation	Final Report and Team Evaluation	Final Report and Team Evaluation

## 4 Ethics and Safety

### 4.1 Ethics

Our initiative addresses the widespread influence of workplace stress, anxiety, and depression, acknowledging them as pivotal challenges that undermine individual well-being and overall productivity. Driven by the imperative for proactive interventions, we aspire to introduce a wearable device integrated with sophisticated sensors and an innovative mood recognition framework. Our objective is to foster a healthier work environment, marking a substantial advancement at the crossroads of technology and mental health within contemporary workplaces. This endeavor aligns seamlessly with the ACM code's commitment to contribute to society and human well-being, recognizing the universal stakeholder role of all individuals in the realm of computing [1].

The collection and use of personal and potentially sensitive data to train our model could infringe on an individual's privacy if not handled correctly, especially in compliance with the ACM Code regarding to respect privacy and confidentiality [1]. We promise to collect only the data necessary for mood recognition to reduce the risk of privacy breaches, and we will ensure that users are fully informed about what data is collected, how it will be used, and obtain their consent. Moreover, we will implement strict access controls so that only authorized personnel can access sensitive data.

### 4.2 Safety

Ensuring safety is a top priority in our project. We have successfully finished the UIUC online safety training. Adhering to safety guidelines, it is compulsory to have a minimum of two team members present in the lab during experiments.

For electrical usage, we will use the USB portal of computer as the power supply of Arduino, and the computer is powered by 220 volts of electricity. Our group fully understands and adheres to the guidelines for safe electricity usage. We will routinely check the computer and Arduino to ensure they operate in a proper environment. Moreover, we have wearable devices to measure the heart rate, body temperature and skin conductance, and those sensors will be connected with Arduino. Then Arduino will be connected to the computer to import data. There is a risk of getting an electric shock when we measure the physiological data of participants. We will strictly follow the safe current limits for electromedical apparatus [3] to ensure the safety of participants. We'll make sure the wires are intact and connected well to prevent the occurrence of electrical leakage.

If the system inaccurately assesses a user's mood, it could lead to inappropriate recommendations or actions of user. Ensuring high accuracy of the mood prediction algorithms and providing users with context about the limitations of the system can minimize this risk.

## References

- [1] ACM. "ACM Code of Ethics and Professional Conduct." (2018), [Online]. Available: <https://www.acm.org/code-of-ethics>.
- [2] A. Khan, D. Windridge and J. Kittler, "Multilevel Chinese Takeaway Process and Label-Based Processes for Rule Induction in the Context of Automated Sports Video Annotation," in *IEEE Transactions on Cybernetics*, vol. 44, no. 10, pp. 1910-1923, Oct. 2014, doi: 10.1109/TCYB.2014.2299955.
- [3] American National Standard Institute. "Safe Current Limits for Electromedical Apparatus" (1993). Published By: Association for the Advancement of Medical Instrumentation (AAMI)
- [4] L. Malott, P. Bharti, N. Hilbert, G. Gopalakrishna and S. Chellappan, "Detecting self-harming activities with wearable devices," 2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), St. Louis, MO, USA, 2015, pp. 597-602, doi: 10.1109/PERCOMW.2015.7134105.
- [5] J. Yang, Y. Li and M. Xie, "MotionAuth: Motion-based authentication for wrist worn smart devices," 2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), St. Louis, MO, USA, 2015, pp. 550-555, doi: 10.1109/PERCOMW.2015.7134097.