

# Assignment #1 : Sensor Data Parsing and Analysis using Python

MS Computer Science,

Submission: 12 October, 2024

## Objective:

To apply fundamental research programming concepts to parse and analyze sensor data from a mobile device log file using Python.

## Task 1: Sensor Data Parsing and Separation

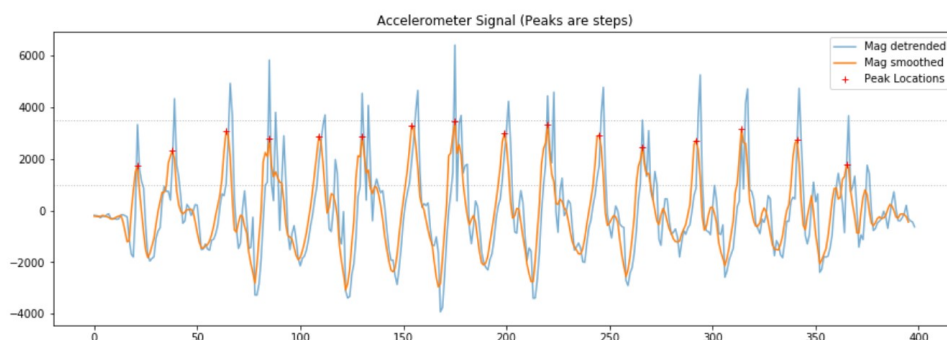
### Instructions:

1. **Obtain the Data:** You will be provided with a mobile device log file containing data from various sensors (e.g., Accelerometer, Gyroscope, Magnetometer, Light, WIFI, pressure).
2. **Create Python Classes:** Design and implement Python classes to represent each sensor type (e.g., Accelerometer, Gyroscope, Magnetometer). Each class should have appropriate attributes and methods to handle sensor-specific data.
3. **Parse the Log File:** Utilize Python's parsing libraries (e.g., CSV, JSON) to read the log file and extract sensor data.
4. **Separate Sensor Data:** Based on the extracted data, instantiate corresponding sensor objects and store their data in separate files or data structures.
5. **Perform Computations :** If applicable, implement computations within the sensor classes to derive additional information from the raw data (e.g., calculate averages, standard deviations, or correlations).

## Task 2: Sensor Data Analysis and Visualization

### Instructions:

1. **Choose a Sensor:** Select one of the parsed sensor data sets for analysis.
2. **Identify Interesting Findings:** Analyze the sensor data to discover patterns, trends, or anomalies. Consider statistical analysis, visualization techniques, or domain-specific knowledge to extract meaningful insights. (Some suggestion are given in Appendix)
3. **Visualize Data:** Use Python's data visualization libraries (e.g., matplotlib, seaborn) to create informative graphs and charts that illustrate your findings.



4. **Document Your Work:** Copy your Python code and the generated visualizations into a Microsoft Word document. Include explanations for your code and analysis.

### **Submission:**

Submit your Microsoft Word document containing the code, output, and analysis to the LMS.

By completing this assignment, you will gain practical experience in data parsing, analysis, and visualization using Python, which are essential skills for research programming.

# Appendix: Sample Interesting Facts to Visualize from Sensor Data

Here are some interesting facts that you can represent in graphs for the various sensors mentioned:

## Accelerometer

- **Activity Recognition:** Visualize the accelerometer data to identify different activities like walking, running, sitting, or lying down.
- **Orientation Detection:** Plot the accelerometer data in 3D to visualize the orientation of the device.
- **Impact Detection:** Detect sudden changes in acceleration to identify impacts or shocks.

## Gyroscope

- **Rotation Detection:** Plot the gyroscope data to visualize the rotational movements of the device.
- **Angular Velocity:** Calculate and visualize the angular velocity of the device.
- **Stabilization:** Analyze the gyroscope data to understand how the device is stabilized.

## Magnetometer

- **Compass Orientation:** Use the magnetometer data to determine the compass heading of the device.
- **Magnetic Field Disturbances:** Identify anomalies in the magnetic field data to detect external disturbances.
- **Metal Detection:** Detect the presence of metal objects near the device.

## Pressure

- **Altitude Estimation:** Calculate the altitude based on the pressure data.
- **Weather Patterns:** Analyze pressure changes over time to identify weather patterns.
- **Barometric Pressure:** Plot the barometric pressure to monitor atmospheric conditions.

## Light

- **Ambient Light:** Visualize the ambient light levels to understand the lighting conditions.
- **Day/Night Detection:** Determine the time of day based on the light levels.
- **Light Source Identification:** Identify different light sources based on the light spectrum.

## Proximity

- **Object Detection:** Detect the presence of objects near the device.
- **Gesture Recognition:** Analyze proximity changes to recognize gestures.
- **Screen On/Off:** Determine when the screen should be turned on or off based on proximity.

## Humidity

- **Environmental Conditions:** Monitor humidity levels to understand environmental conditions.

- **Comfort Level:** Analyze humidity and temperature data to determine comfort levels.
- **Weather Patterns:** Identify weather patterns based on humidity changes.

## Temperature

- **Environmental Conditions:** Monitor temperature levels to understand environmental conditions.
- **Comfort Level:** Analyze temperature and humidity data to determine comfort levels.
- **Temperature Anomalies:** Detect temperature anomalies to identify potential issues.

## Orientation

- **Device Orientation:** Visualize the device orientation in 3D.
- **Rotation Detection:** Detect rotations around different axes.
- **Stabilization:** Analyze orientation data to understand stabilization mechanisms.

## GNSS/GPS

- **Location Tracking:** Plot the device's location on a map.
- **Movement Patterns:** Analyze movement patterns to identify routes or activities.
- **Altitude Changes:** Visualize altitude changes over time.

## WIFI

- **Network Connectivity:** Analyze WIFI signal strength to assess network connectivity.
- **Location Estimation:** Use WIFI access points to estimate location.
- **Network Scanning:** Visualize the number of detected WIFI networks.

## Bluetooth

- **Device Discovery:** Analyze Bluetooth data to identify nearby devices.
- **Connection Strength:** Visualize the strength of Bluetooth connections.
- **Device Interactions:** Analyze Bluetooth data to understand device interactions.

## BLE 4.0

- **Beacon Detection:** Detect BLE beacons and visualize their proximity.
- **Indoor Positioning:** Use BLE beacons for indoor positioning.
- **Device Interactions:** Analyze BLE data to understand device interactions.

## Sound

- **Noise Levels:** Measure and visualize noise levels.
- **Sound Source Identification:** Identify different sound sources based on sound characteristics.
- **Sound Events:** Detect sound events like clapping or knocking.

## RFID Reader

- **Tag Detection:** Detect and visualize RFID tags.
- **Tag Identification:** Identify different types of RFID tags.

- **Tag Interactions:** Analyze RFID data to understand tag interactions.

## IMU XSens

- **Combined Sensor Data:** Analyze the combined accelerometer, gyroscope, and magnetometer data from the IMU.
- **Motion Tracking:** Track complex motions using the IMU data.
- **Orientation Estimation:** Estimate device orientation using the IMU data.

These are just a few examples of interesting facts that can be visualized from sensor data. Students can explore other aspects of the data and create their own visualizations based on their research interests.