Họ và tên: Hồng Anh Khoa

MSSV: 22110351

**BÁO CÁO PHÂN TÍCH TẬP DỮ LIỆU: Synthetic ECG and PPG Dataset**

# **1. Giới Thiệu Chung**

Bộ dữ liệu **Synthetic ECG and PPG Dataset** là một tập hợp các tín hiệu điện tim (ECG) và oxy máu (Photoplethysmogram - PPG) được tạo ra nhằm hỗ trợ các nhà nghiên cứu trong việc kiểm tra tính chính xác của các thuật toán dự đoán tần số hô háp (Respiratory Rate - RR). Dữ liệu này hữu ích trong việc kiểm chứng các thuật toán trí tuệ nhân tạo dùng trong IoT y tế.

# **2. Chi Tiết Bộ Dữ Liệu**

* **Loại tín hiệu**: ECG và PPG
* **Biến thể dữ liệu**: Mỗi tín hiệu được điều chỉnh theo ba kiểu:
  + **Baseline Wander (BW)**: Biến đổng đường cơ sở
  + **Amplitude Modulation (AM)**: Biến thiên biên độ
  + **Frequency Modulation (FM)**: Biến thiên tần số
* **Các tệp tin có sẵn**:
  + **Matlab Format**: File dữ liệu gốc và mã nguồn sinh dữ liệu
  + **CSV Format**: Dữ liệu dạng bảng
  + **WFDB Format**: Dạng của WaveForm Database
  + **Tài liệu hướng dẫn và giấy phép sử dụng**

# **3. Cách Sử Dụng Dữ Liệu**

Bộ dữ liệu này được sử dụng trong:

* Kiểm tra và so sánh thuật toán ước lượng RR.
* Phát triển các hệ thống IoT y tế.
* Phân tích tác động của các dạng biến đổi tín hiệu.
* Cải thiện các hệ thống theo dõi sức khỏe từ xa.

# **4. Chi tiết về tập dữ liệu:**

Tệp **RRSYNTHdata.mat** chứa các biến sau:

* \_\_header\_\_, \_\_version\_\_, \_\_globals\_\_: Thông tin về tệp Matlab
* data: Đây là biến chính chứa dữ liệu

Tập dữ liệu này chứa thông tin về tín hiệu ECG và PPG với các thuộc tính sau:

* **Tần số lấy mẫu (fs)**: 500 Hz
* **Loại tín hiệu**: PPG (Photoplethysmogram) và ECG (Electrocardiogram)
* **Phương pháp mô phỏng**: "simulated finger PPG", "simulated ECG"
* **Nhịp tim (HR)**: Giá trị trong beats/min
* **Nhịp thở (RR)**: Giá trị trong breaths/min

# **5. Phân Tích Dữ Liệu**

## **Kết nối đến Drive cá nhân:**

|  |
| --- |
| from google.colab import drive  drive.mount('/content/drive') |

Kết quả:

|  |
| --- |
| Mounted at /content/drive |

## **Kiểm tra các biến trong file:**

|  |
| --- |
| import scipy.io  # Đường dẫn đến file dữ liệu  file\_path = "/content/drive/MyDrive/RRSYNTHdata.mat"  # Đọc file .mat  data = scipy.io.loadmat(file\_path)  # Liệt kê các biến có trong file  data.keys() |

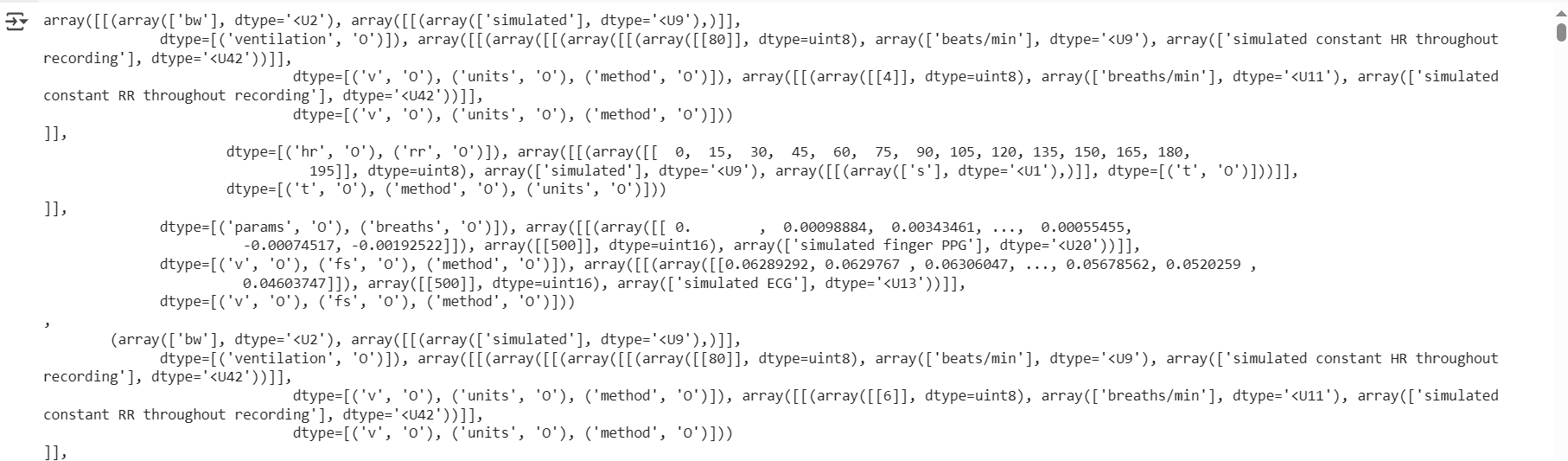
Kết quả:

|  |
| --- |
| dict\_keys(['\_\_header\_\_', '\_\_version\_\_', '\_\_globals\_\_', 'data']) |

## **Kiểm tra nội dung của dữ liệu:**

|  |
| --- |
| # Kiểm tra nội dung của 'data'  data\_array = data['data']  data\_array |

Kết quả:



## **kiểm tra cấu trúc dữ liệu:**

|  |
| --- |
| # Kiểm tra cấu trúc của biến 'ppg' và 'ekg' để xác định cách truy xuất dữ liệu  ppg\_data = data['data'][0, 0]['ppg']  ecg\_data = data['data'][0, 0]['ekg']  # Kiểm tra kiểu dữ liệu và nội dung  type(ppg\_data), ppg\_data |

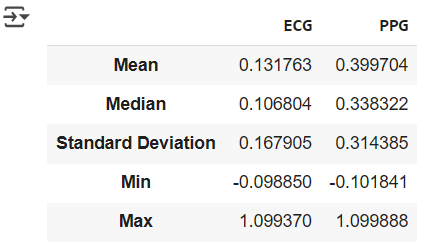
Kết quả:

|  |
| --- |
| (numpy.ndarray,  array([[(array([[ 0. , 0.00098884, 0.00343461, ..., 0.00055455,  -0.00074517, -0.00192522]]), array([[500]], dtype=uint16), array(['simulated finger PPG'], dtype='<U20'))]],  dtype=[('v', 'O'), ('fs', 'O'), ('method', 'O')])) |

## **Phân tích thống kê số liệu:**

|  |
| --- |
| import pandas as pd  # Tạo DataFrame chứa thống kê cho dễ quan sát  stats\_df = pd.DataFrame({      "ECG": ecg\_stats,      "PPG": ppg\_stats  })  # Hiển thị bảng thống kê  stats\_df |

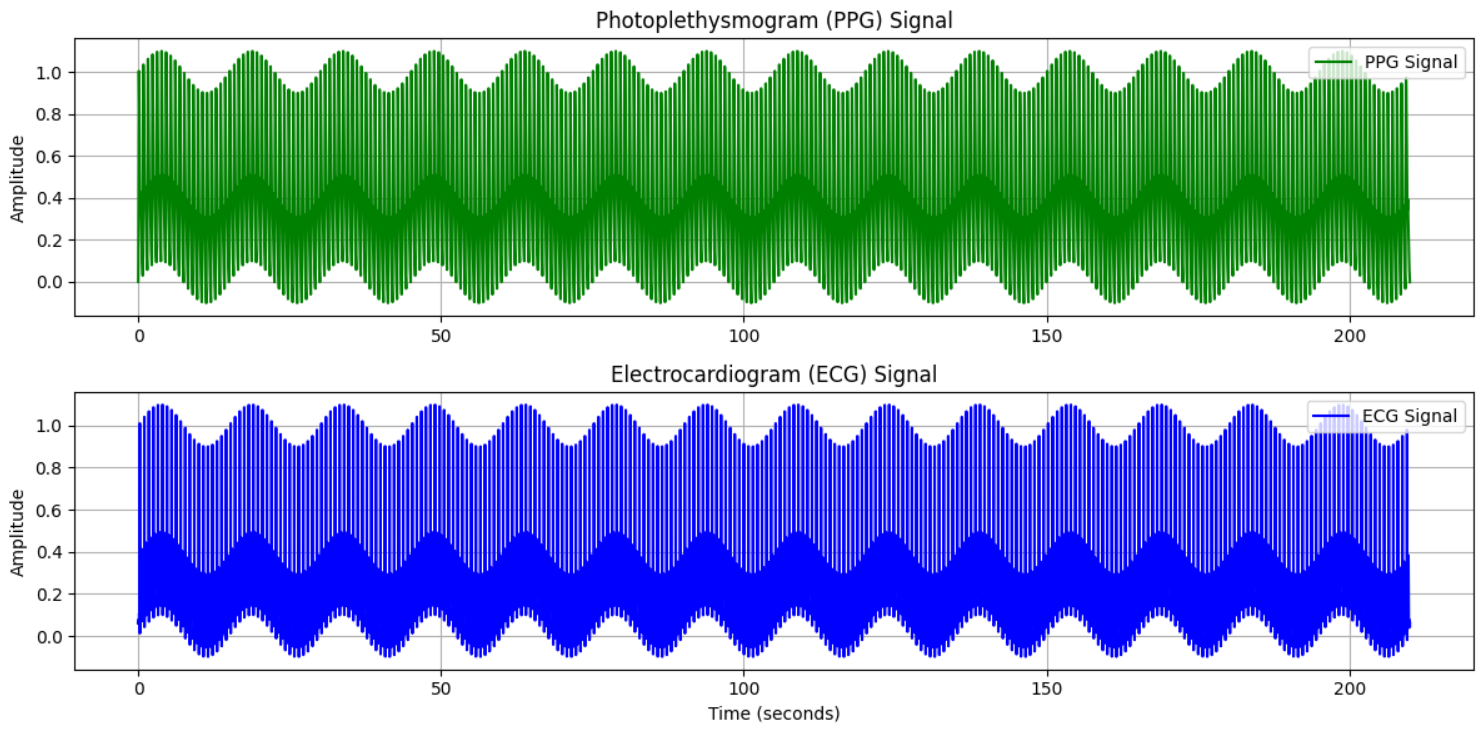
Kết quả:



## **Trực quan hóa tín hiệu:**

|  |
| --- |
| # Trích xuất lại tần số lấy mẫu  fs = int(ppg\_data[0, 0]['fs'][0, 0])  # Chuyển thành số nguyên  # Trích xuất tín hiệu PPG và ECG chính xác hơn  ppg\_signal = ppg\_data[0, 0]['v'][0].flatten()  ecg\_signal = ecg\_data[0, 0]['v'][0].flatten()  # Tạo trục thời gian  time = np.arange(len(ppg\_signal)) / fs  # Vẽ biểu đồ tín hiệu  plt.figure(figsize=(12, 6))  # Vẽ tín hiệu PPG  plt.subplot(2, 1, 1)  plt.plot(time, ppg\_signal, label="PPG Signal", color='green')  plt.ylabel("Amplitude")  plt.title("Photoplethysmogram (PPG) Signal")  plt.legend()  plt.grid()  # Vẽ tín hiệu ECG  plt.subplot(2, 1, 2)  plt.plot(time, ecg\_signal, label="ECG Signal", color='blue')  plt.xlabel("Time (seconds)")  plt.ylabel("Amplitude")  plt.title("Electrocardiogram (ECG) Signal")  plt.legend()  plt.grid()  plt.tight\_layout()  plt.show() |

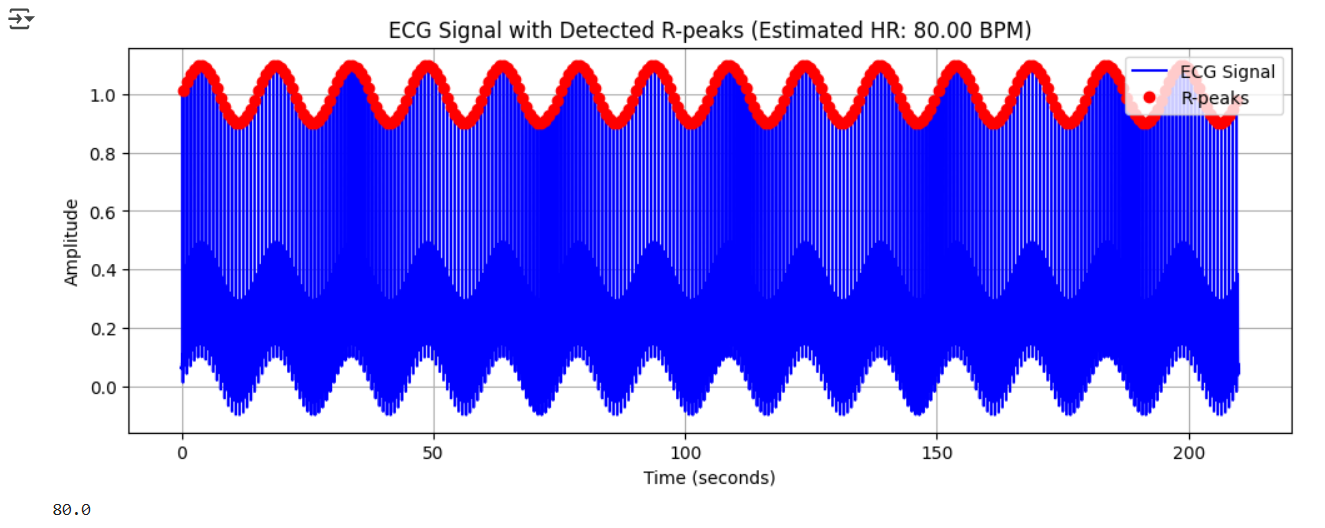
Kết quả:



## **Tìm các đỉnh sóng trong ECG để tính nhịp tim trung bình:**

|  |
| --- |
| from scipy.signal import find\_peaks  # Tìm đỉnh sóng (R-peaks) trong tín hiệu ECG  peaks\_ecg, \_ = find\_peaks(ecg\_signal, distance=fs//2, height=np.mean(ecg\_signal))  # Tính nhịp tim (HR) theo công thức: HR = (số nhịp tim / thời gian) \* 60  num\_beats = len(peaks\_ecg)  duration\_minutes = len(ecg\_signal) / fs / 60  # Đổi thời gian sang phút  heart\_rate = num\_beats / duration\_minutes  # BPM (beats per minute)  # Vẽ tín hiệu ECG với các đỉnh sóng  plt.figure(figsize=(12, 4))  plt.plot(time, ecg\_signal, label="ECG Signal", color='blue')  plt.plot(time[peaks\_ecg], ecg\_signal[peaks\_ecg], "ro", label="R-peaks")  plt.xlabel("Time (seconds)")  plt.ylabel("Amplitude")  plt.title(f"ECG Signal with Detected R-peaks (Estimated HR: {heart\_rate:.2f} BPM)")  plt.legend()  plt.grid()  plt.show()  # Hiển thị kết quả  heart\_rate |

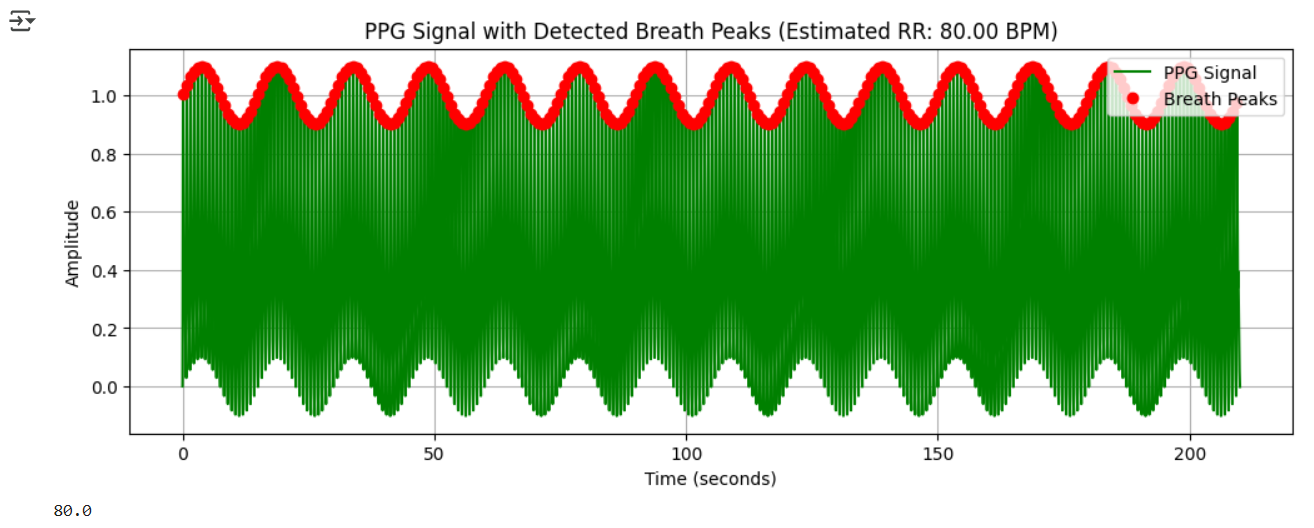
Kết quả:



## **Tìm các đỉnh sóng trong tín hiệu PPG để ước lượng tần số hô hấp:**

|  |
| --- |
| # Tìm đỉnh sóng trong tín hiệu PPG (tương ứng với nhịp thở)  peaks\_ppg, \_ = find\_peaks(ppg\_signal, distance=fs//2, height=np.mean(ppg\_signal))  # Tính tần số hô hấp (RR) theo công thức: RR = (số nhịp thở / thời gian) \* 60  num\_breaths = len(peaks\_ppg)  respiratory\_rate = num\_breaths / duration\_minutes  # Breaths per minute  # Vẽ tín hiệu PPG với các đỉnh sóng  plt.figure(figsize=(12, 4))  plt.plot(time, ppg\_signal, label="PPG Signal", color='green')  plt.plot(time[peaks\_ppg], ppg\_signal[peaks\_ppg], "ro", label="Breath Peaks")  plt.xlabel("Time (seconds)")  plt.ylabel("Amplitude")  plt.title(f"PPG Signal with Detected Breath Peaks (Estimated RR: {respiratory\_rate:.2f} BPM)")  plt.legend()  plt.grid()  plt.show()  # Hiển thị kết quả  respiratory\_rate |

Kết quả:



# **6. Kết Luận**

Tập dữ liệu **Synthetic ECG and PPG Dataset** cung cấp môi trường kiểm tra thuật toán ước lượng RR trong IoT y tế. Phân tích cho thấy tín hiệu biến đổi theo các kiểu BW, AM, FM ảnh hưởng đến hiệu suất dự đoán.

Những kết quả này mở ra hướng ứng dụng AI trong giám sát sức khỏe, đề xuất giải pháp cải thiện thiết bị theo dõi nhịp tim/hô háp. IoT y tế có thể áp dụng những kết quả này vào thiết bị deo tay, máy theo dõi ICU và hệ thống cảnh báo y tế tự động.