MIT-BIH Arrhythmia Database Modern 2023

Course Title:

- **Digital Signal Processing**

Name:

- ***Mohamed Mahmoud Mossad Mohamed Elseragy***

CH:

- **CH2100032**

Section:

- **Sec (1)**

Supervised by:

- **Assoc. Prof. Dr. Marwa Eid**
- **Eng. Omnia M. Osama**

Project Objective:

The goal of this project is to analyze Electrocardiogram (ECG) signals from the MIT-BIH Arrhythmia Database to detect abnormal heart rhythm patterns (Arrhythmias), using signal processing and data analysis techniques.
This type of analysis helps in developing intelligent systems that can monitor and detect heart rhythm disorders early and accurately, supporting smart healthcare and real-time medical monitoring applications.

Tools and Techniques Used:

Analysis Tools:

- **Python** – The primary programming language used.
- **Pandas** – For loading and cleaning the data.
- **NumPy** – For mathematical operations and transformations.
- **Matplotlib / Seaborn** – For data visualization and exploratory analysis.

- **SciPy / FFT** – For transforming the signal into the frequency domain (Fourier Transform).
- **Glob** – For loading multiple files at once.

About of Datasets:

Source:

- https://www.kaggle.com/datasets/protobioengineering/mit-bih-arrhythmia-database-modern-2023

Content**:**

- This dataset contains data related to **Electrocardiogram (ECG)**, which is used to study **Arrhythmias** (irregular heart rhythms).
- The dataset includes 48 ECG recordings collected from 47 patients at the **Massachusetts Institute of Technology (MIT) Hospital**, with 23 different types of heart rhythm abnormalities.
- The data is formatted in **CSV files**, making it easy to analyze using tools such as Python and Pandas.

Purpose of the Dataset:

- The primary purpose of this dataset is to **assist researchers in analyzing various arrhythmia patterns** using data analysis techniques such as machine learning and signal analysis.
- It is mainly used to develop **heart monitoring systems** and **predictive models** that can detect **arrhythmia patterns** at an early stage.

Dataset Loading & Sampling:

Initially analysis, more efficient and manageable, a subset of **10 files** was selected for processing and exploration.

The files were loaded using Python's glob module, and each file was read into a Data Frame. An additional column called source file was added to identify the origin of each row. Any unnecessary columns, such as unnamed index columns, were removed during this process.

Data Preprocessing:

- **Detection of Missing Values:**
  The dataset was first examined to check for any missing (NaN) values across its columns. This step is crucial to avoiding issues in downstream analysis or model training.

- **Handling Missing Values:**
  It was found that some values were missing in the ECG signal columns MLII and V5. These missing values were handled by replacing them with the mean value of each **respective column, ensuring that the overall statistical distribution of the data remained consistent.**

- **Checking for Infinite Values:**
  An additional check was performed to verify the absence of infinite values (inf) in the dataset. Infinite values can disrupt mathematical operations and need to be addressed if present. Fortunately, no such values were found in this case.

Exploratory Data Analysis (EDA):

ECG Signal Visualization & Relationship:

1. Signal Distribution Analysis:
   - Histograms were plotted for both MLII and V5 signals.
   - These plots revealed the overall distribution and spread of signal amplitudes.
   - Helps in identifying skewness, outliers, and typical value ranges in both leads.
2. Correlation Analysis:
   - A correlation heatmap was generated to assess the linear relationship between the two signals.
   - The coefficient correlation showed how strongly MLII and V5 are related to each other.
   - to evaluate signal redundancy or complementarity.
3. Pair plot (Scatter Matrix):
   - A pair plot was used to visualize the relationship and distribution between MLII and V5.
   - It helps in spotting clusters, trends, or anomalies in the signal behavior

4. Time-Series Signal Vis:
    - Both MLII and V5 signals were plotted against time (in milliseconds).
    - This visualization simulates the actual ECG waveform, giving a clear picture of the heartbeat patterns.
    - It's especially useful for detecting arrhythmia, signal noise, or abnormalities visually.

Time Domain Analysis:

- The ECG signals were filtered over a specific time range, from 1000 milliseconds to 2000 milliseconds, to isolate and analyze the signals within this segment.
- Two signals, MLII and V5, were plotted against time to visually inspect their amplitude and behavior during this time window.
- This allowed us to verify the time-domain characteristics of the signals, ensuring they are within expected ranges.

Frequency Domain Analysis:

- Fast Fourier Transform (FFT) was applied to the MLII and V5 signals to convert them into the frequency domain and analyze their frequency content.
- The frequency spectrum was plotted to visualize the magnitude of different frequency components.
- By removing the means from the signals before applying the FFT (icleaning the signals), we could ensure that only the actual frequency components were observed, without interference from any DC offset.

Low-Pass Filtering:

- To remove high-frequency noise and focus on the relevant low-frequency components of the ECG signals, a **low-pass filter** was applied to both **MLII** and **V5** signals.
- The **filtered signals** were plotted to visualize the impact of the filter, which helps in reducing unwanted high-frequency components, especially noise.
- The filter used had a **cutoff frequency of 1.0 Hz**, which allowed us to preserve the important physiological frequencies of the heart's electrical activity.

List of IDs:
First, a set of sample IDs (101,106,107,108,109,111,112,113) was specified. These IDs represent the data samples we want to work with from the database.

Generating File Paths:
After selecting the IDs, we created file paths based on these IDs. These paths point to the locations of the CSV files containing the data for each sample's ECG signal.

Function to Extract Data:
We then created a function called plot original, which reads the data from each CSV file based on the provided path. It extracts the time and ECG signal (V1) for a certain number of points (2048 points).

Gathering Data:
This function is applied to all the file paths in the list, which results in a new list that holds the time and signal data for each sample.

Plotting the Signals:
the signals for all samples are plotted using subplots in a 2x4 grid layout. Each signal is displayed with a title corresponding to its sample ID, making it easy to compare the different ECG signals.

Here is the link to the notebook:
https://www.kaggle.com/code/mohamedmahmoud111/dsp-for-ecg