

REPORT OF LAB5

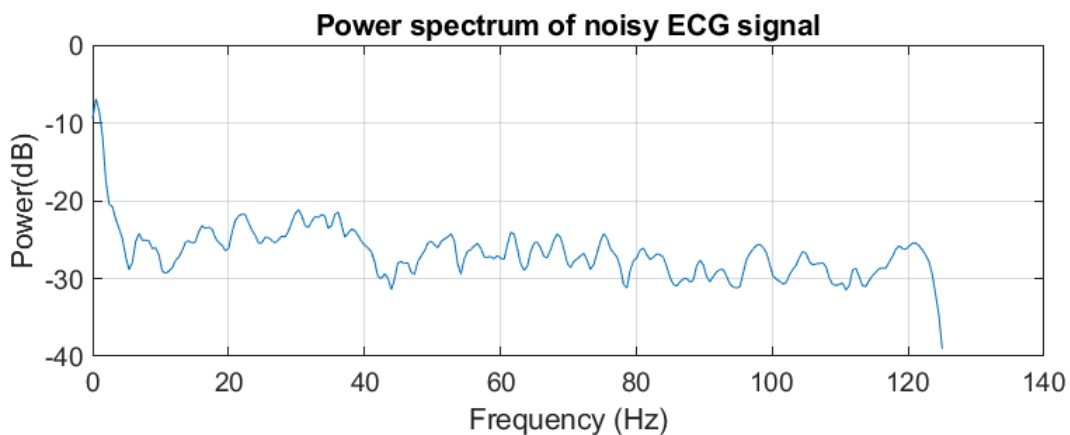
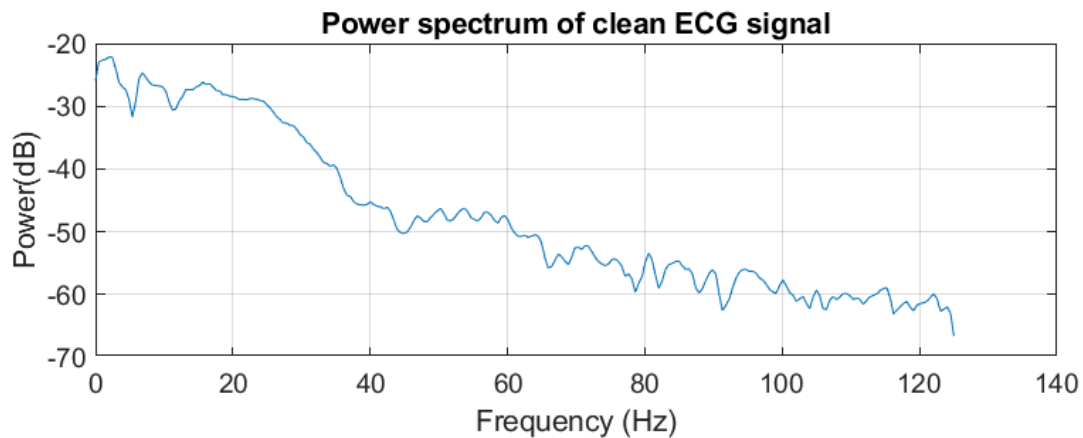
Sara Rezanejad | 99101643

Ali Khosravipour | 99101502

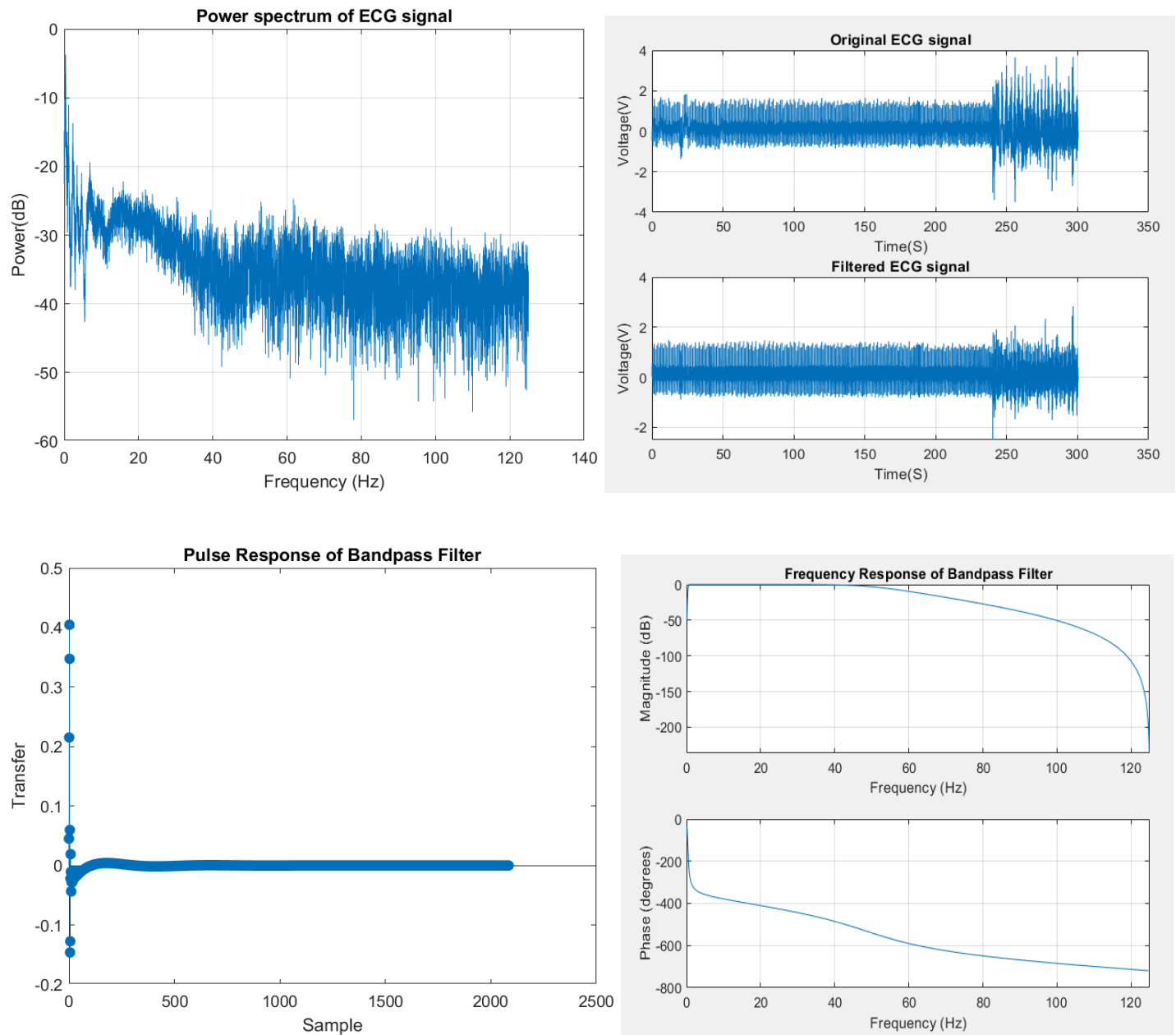
Mohamad Hosein Faramarzi | 99104095

Part 1:

A) The frequency content of the noisy signal is higher at high frequencies. In contrast, the clean signal has more content at lower frequencies.

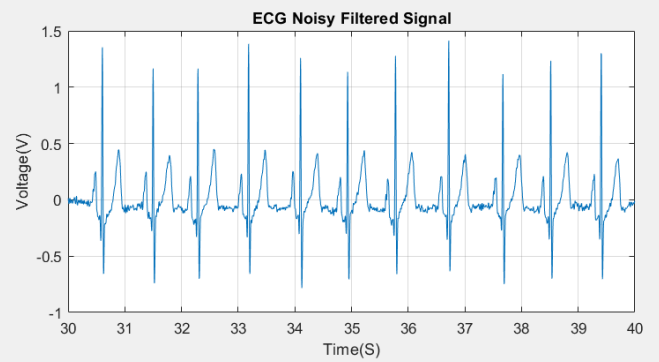
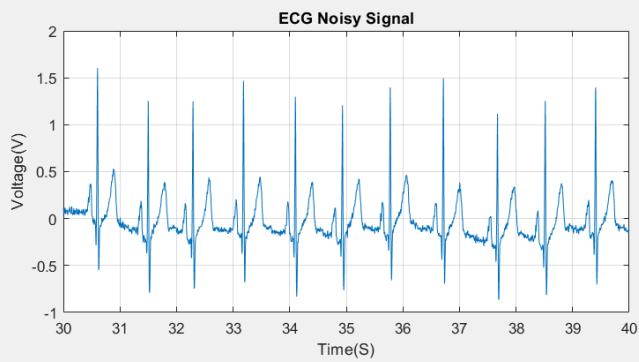
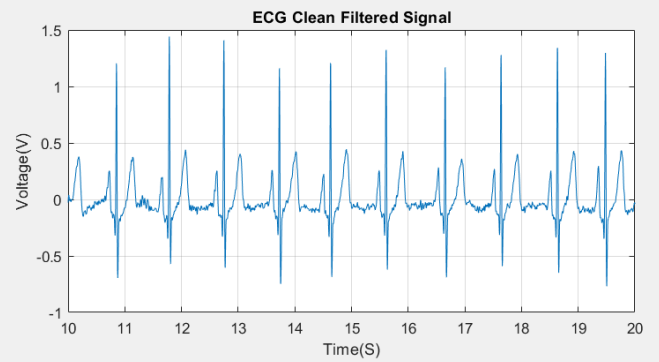
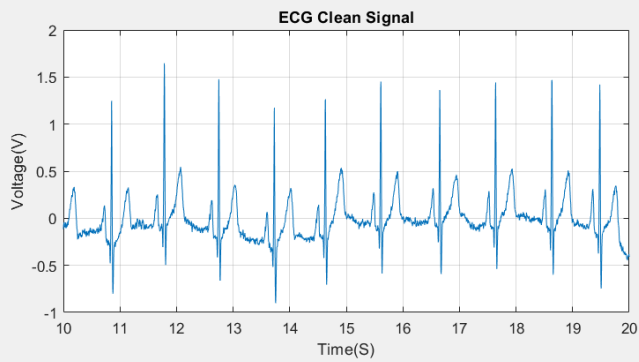
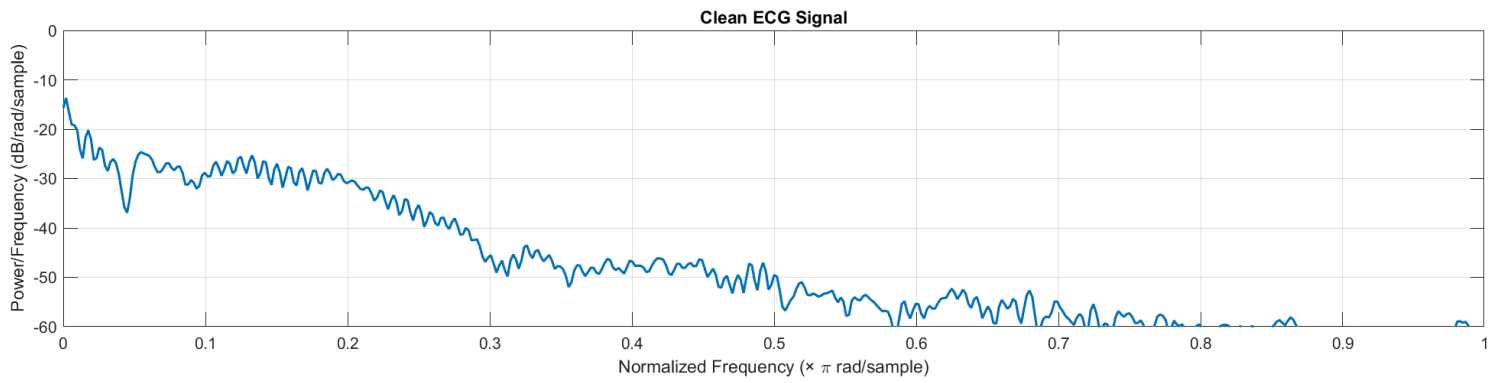
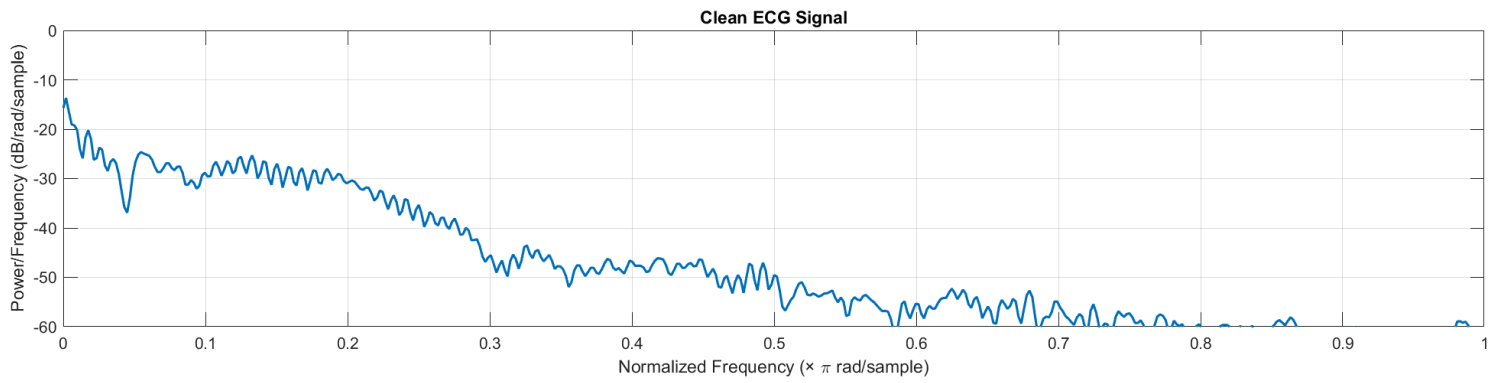


B)



c)

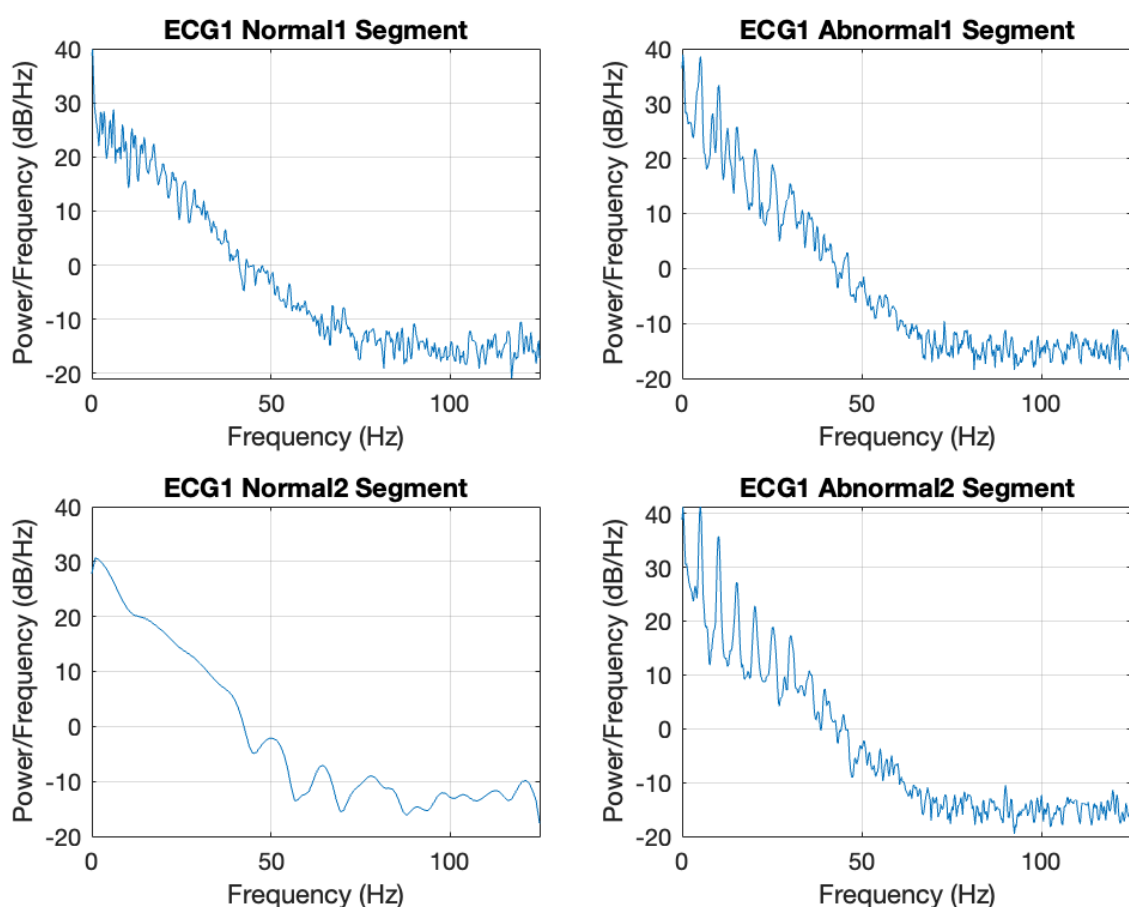
Based on this palette, we can see that the designed filter has a good performance in removing noise because the output of the two graphs are similar.



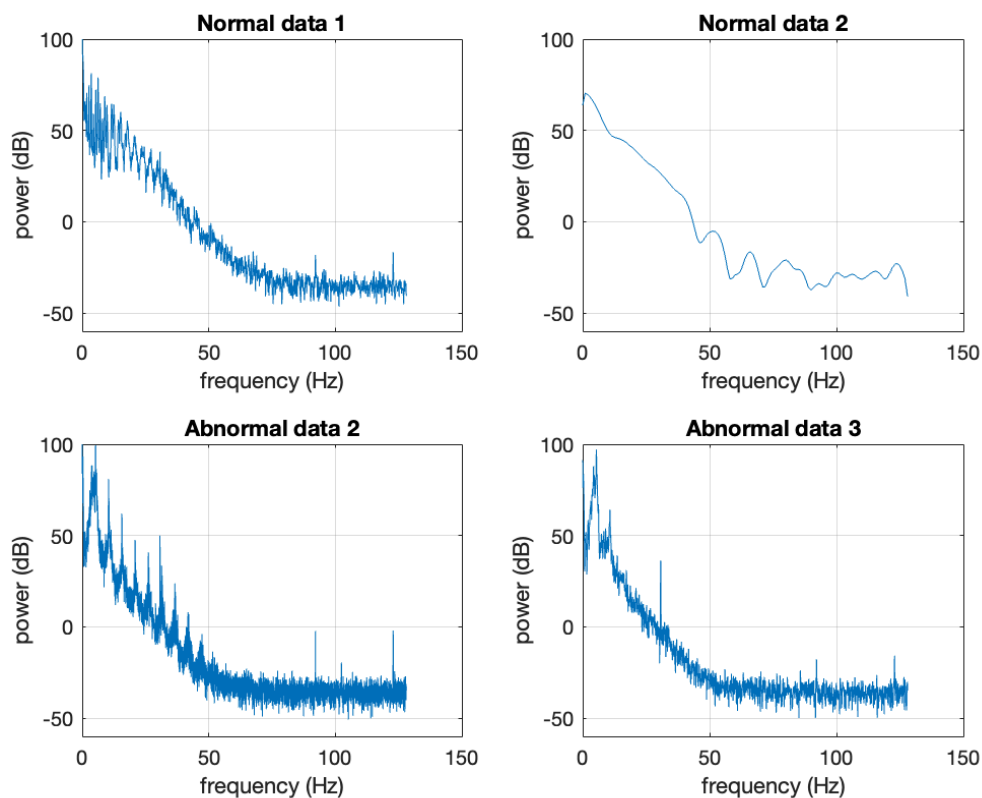
الف) برای طراحی سیستم خود، ابتدا باید معیارهایی را برای تمایز بین ECG طبیعی و فلوتر/فیبریلاسیون بطنی تعیین کنید. یک یا چند فایل داده فهرست شده در زیر را انتخاب کنید، آنها را با استفاده از تابع load در Matlab بخوانید و سیگنال‌های ECG را بررسی کنید (دو فایل اول لیست احتمالاً ساده‌ترین برای شروع هستند). هر بخش ECG غیرنرمال (abnormal) حاوی بخشی است که برای آن بیمار "طبیعی" است. دو یا سه جفت بخش داده را انتخاب کنید، جایی که هر جفت شامل یک ریتم "طبیعی" برای آن بیمار و بخشی است که در آن آریتمی بطنی رخ می‌دهد. محتوای فرکانس تمام بخش‌ها را با استفاده از pwelch تجزیه و تحلیل کنید و بین بخش‌های عادی و آریتمی مقایسه کنید. از مشاهدات خود برای ارائه معیاری برای تشخیص آریتمی‌های بطنی از ریتم‌های طبیعی استفاده کنید.

در این بخش از داده n_422 استفاده می‌کنیم. در ادامه محتوای فرکانسی دو بخش از حالت نرمال و همچنین دو بخش از حالت غیر نرمال را رسم خواهیم کرد تا تفاوت بین این دو حالت در طیف توان مشاهده شود:

Frequency Domain Analysis of ECG1 Segments



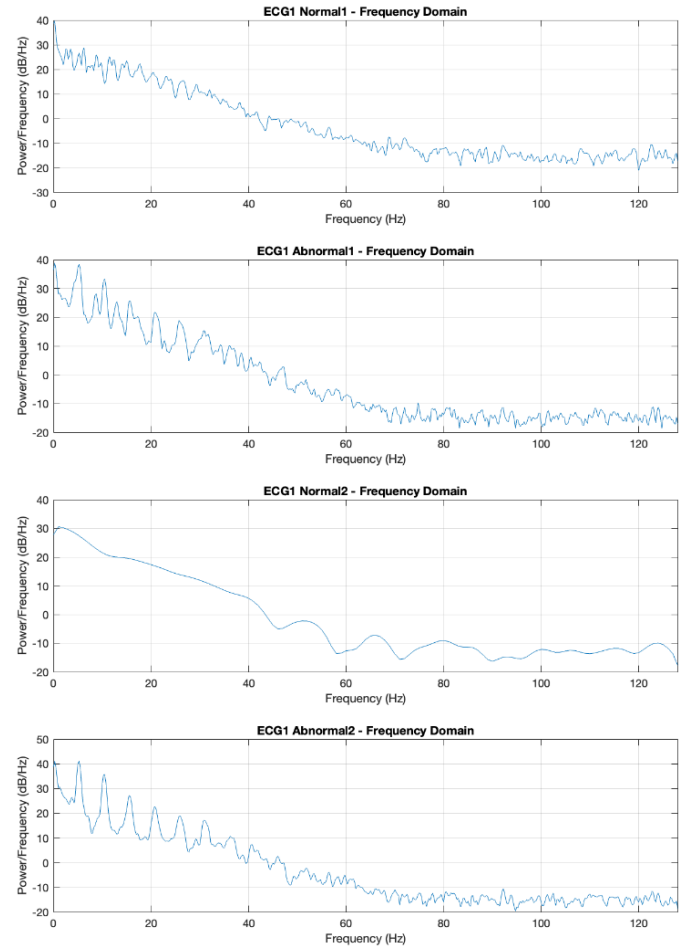
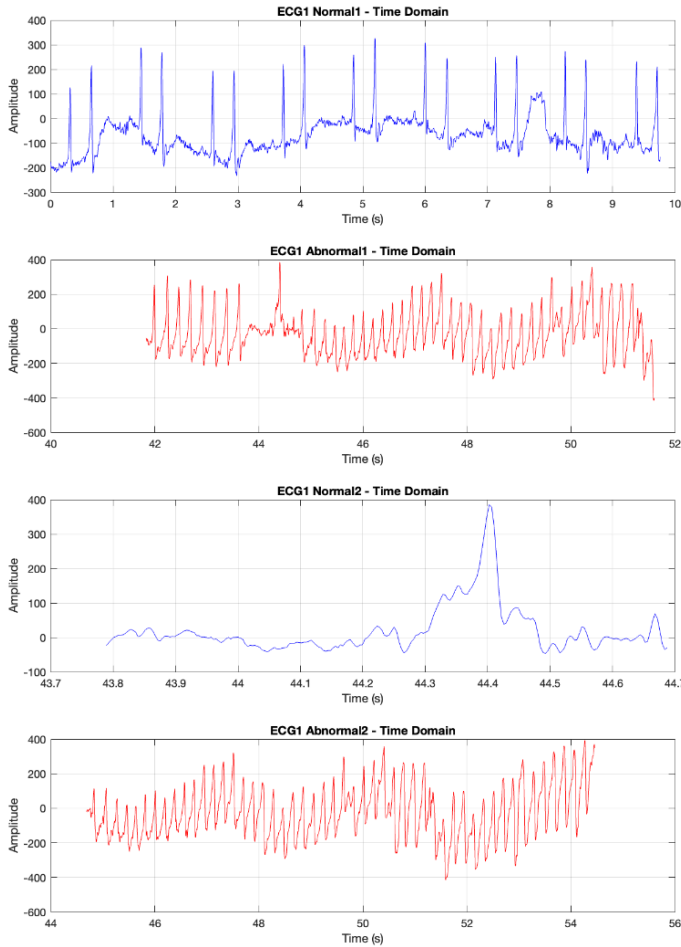
همانطور که مشاهده می کنید چه از نظر اغتشاش چه از نظر دامنه توان در فرکانس های پایین قسمت های غیر نرمال مقادیر بیشتری را خواهند داشت. در شکل زیر یک بخش دیگر از abnormal اضافه شده و همچنین اسکیل توان را برای مشاهده بهتر افکت گفته شده تغییر داده ایم.



ب) بخش های آریتمی بطنی چه تفاوتی با بخش های نرمال در هر دو حوزه زمان و فرکانس دارند؟ (نمودارهای مربوطه را بیاورید)

بازه های مربوطه را چه در حوزه فرکانس و چه در حوزه زمان رسم می کنیم. با توجه به اشکال حوزه فرکانس میزان توان در فرکانس های پایین تر و همچنین اغتشاش توان در حالت غیر نرمال بیشتر است. در حوزه زمان نیز ترکیب QRS و شکل منظم سیگنال در حالت نرمال به وضوح می شود در حالیکه در سیگنال غیر نرمال همچین چیزی را نمی بینیم. نوسانات سیگنال غیر نرمال نیز در حوزه زمان بیشتر خواهد بود.

Time and Frequency Domain Analysis of Updated ECG1 Segments



قسمت پ)

با توجه به طول پنجره و میزان شیفت داده شده، سیگنال‌ها را جدا کرده و آن‌ها را به بازه‌های زمانی یا epoch تقسیم می‌کنیم. این تقسیم‌بندی طوری است که هر پنجره یک بخش مجزا از سیگنال را تشکیل می‌دهد. سپس با توجه به زمان هر epoch و اطلاعات موجود در فایل لیبل مربوط به هر پنجره را مشخص می‌کنیم.

در پایان یک ماتریس دو بعدی خواهیم داشت که هر سطر آن نمایانگر داده‌های سیگنال زمانی یک پنجره است. علاوه بر این، یک بردار خواهیم داشت که طول آن برابر با تعداد پنجره‌ها است و شامل لیبل‌های مربوط به هر پنجره می‌باشد.

ت) در مورد سیگنال n_{422} فقط تشخیص پنجره‌های *VFIB* از *Normal* برای ما مهم است. ابتدا می‌خواهیم کارایی ویژگی‌های فرکانسی را در طبقه‌بندی این دو نوع پنجره بررسی کنیم. از بین ویژگی‌های مختلف فرکانسی، چند ویژگی را انتخاب کرده و برای همه پنجره‌های داده محاسبه نمایید. می‌توانید از دستورات زیر استفاده کنید (*bandpower*) (با در نظر گرفتن محدوده‌ای مختلف فرکانسی/ با استفاده از نتایج بخش (ب))، *meanfreq*، *medfreq* و ...).

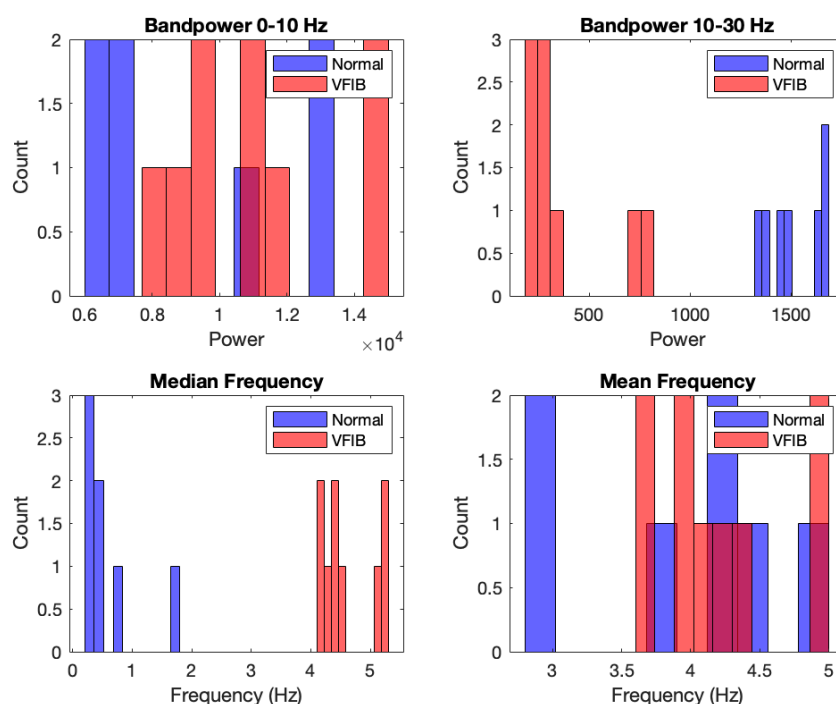
برای این کار از چهار ویژگی زیر برای طبقه بندی و جداسازی سیگنال ها استفاده کرده ایم. نتیجه را نیز در تصویر زیر می توانید مشاهده کنید. با توجه به نتایج دو ویژگی دوم و سوم گزینه های خوبی برای تفکیک هستند:

1- bandpower 0-10Hz

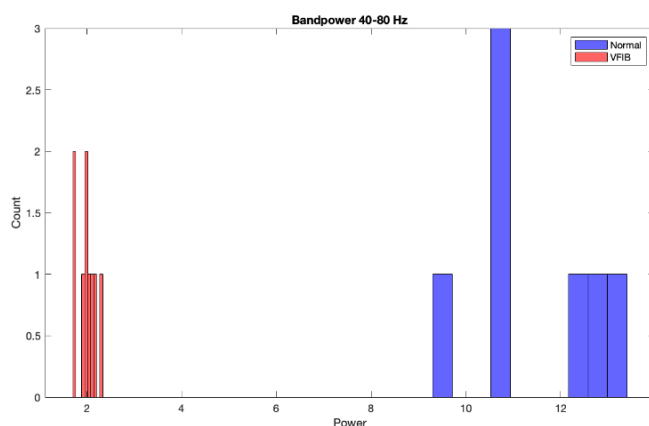
2- banpower 10-30Hz

3-Medfreq

4-Meanfreq



همانطور که مشاهد می کنید دو ویژگی پاور در ۱۰ الی ۳۰ هرتز و همچنین median frequency به خوبی حالت نرمال و VFIB را از یکدیگر جدا کرده اند. در انتخاب بازه فرکانسی دست ما باز است و می توانیم از بازه فرکانسی ۴۰ الی ۸۰ هرتز نیز استفاده کنیم:



(چ) برای هر یک از ویژگی‌های بخش (ج)، ماتریس *Confusion* (دو کلاس، با در نظر گرفتن *VFIB* به عنوان *positive* و *Normal* به عنوان *negative*) را تشکیل داده و معیارهای *Accuracy*، *Sensitivity* و *Specificity* را محاسبه نمایید. در مورد پنجره‌هایی که برچسبی به جز *Normal* و *VFIB* دارند، چه تشخیصی انجام شده است؟

برای این بخش ما با توجه به ویژگی‌های مختلف دو تابع *va_detect* و *va_detectSuper* را استفاده کرده ایم که اولی براساس دو ویژگی اول ذکر شده نوشته شده و دومی تمام ویژگی‌ها را تست خواهد کرد (بعلاوه بخش‌های بعدی) تا بهترین آنها از نظر صحت و *sensitivity* و *speceficity* مشخص شود:

```
% Analysis loop: each iteration processes one frame of data
for i = 1:frame_N
    % Get the next data segment
    seg = ecg_data(((i - 1) * frame_step + 1):((i - 1) * frame_step + frame_length));

    % Calculate bandpower in the 10-30 Hz range
    bandpower_10_30Hz = bandpower(seg, Fs, [10 30]);

    % Calculate median frequency
    median_freq = medfreq(seg, Fs);

    % Set alarm if any of the thresholds are exceeded
    if bandpower_10_30Hz < bandpower_threshold || median_freq > median_freq_threshold
        alarm(i) = 1;
    end
end
```

Total number of segments: 59

Confusion Matrix:

7	1
0	11

از دو ویژگی اول دقت زیر را نتیجه می‌گیریم :

Sensitivity: 1

Specificity: 0.875

Accuracy: 0.94737

در ادامه با اضافه کردن ویژگی *bandpower 40-80* کارایی

بهتر خواهد شد:

Confusion Matrix:

0	7
0	8

Feature = bandpower at 40-80 Hz

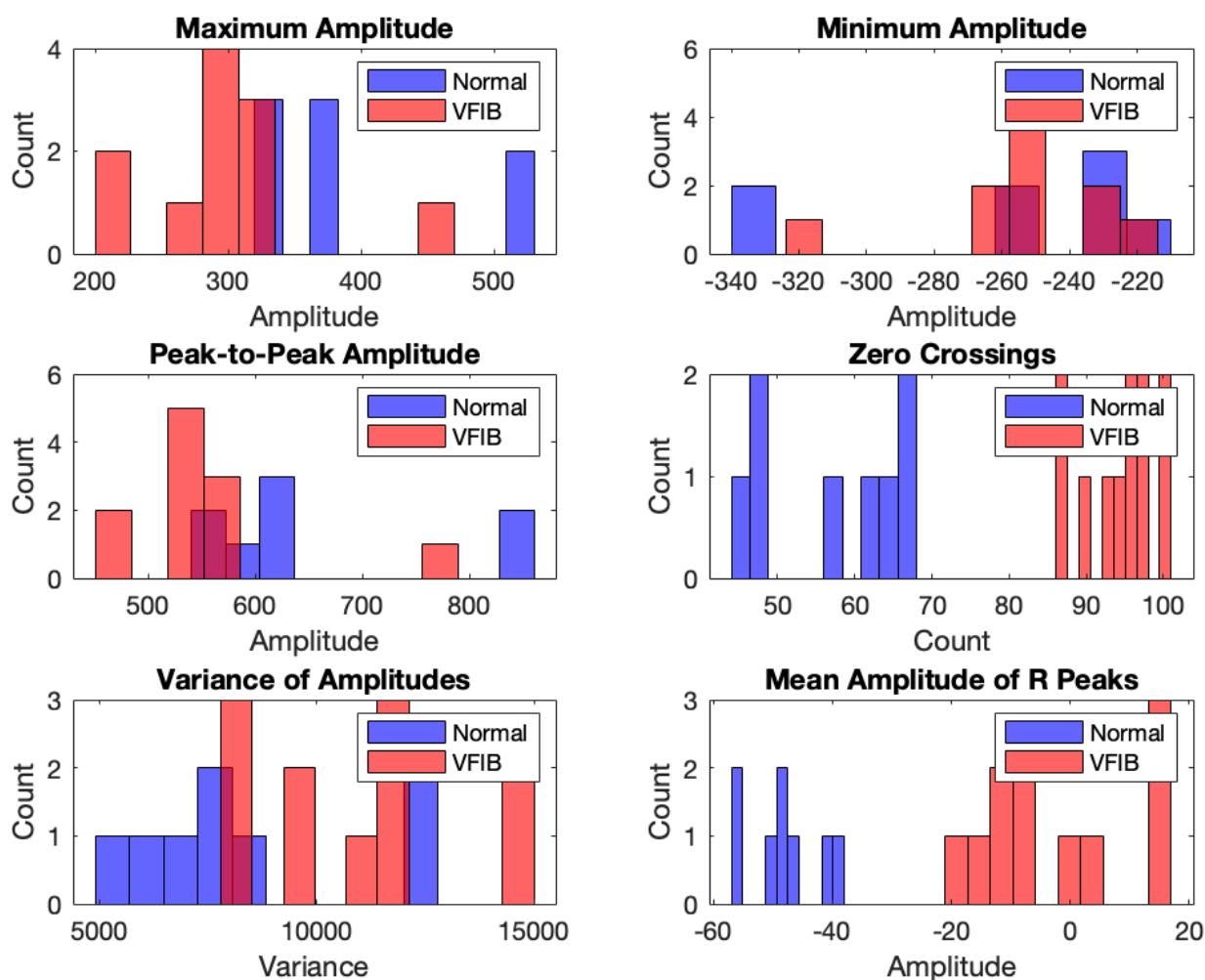
Sensitivity: 1

Specificity: 1

Accuracy: 1

خ) برای هر یک از ویژگی‌های محاسبه‌شده در قسمت (ج)، هیستوگرام (دستور *hist*) ویژگی را برای دو کلاس *Normal* و *VFIB* رسم کرده و با هم مقایسه کنید. آیا دو کلاس با استفاده از این ویژگی (ها) قابل جداسازی هستند؟ برای هر ویژگی آستانه‌ای برای طبقه‌بندی دو کلاس انتخاب نمایید.

برای هر پنجره، ۶ ویژگی مورفولوژیکی شامل ماکزیمم، مینیمم، دامنه قله به قله (*peak-to-peak*)، میانگین دامنه قله‌ها، تعداد عبور از صفر و واریانس سیگنال محاسبه می‌شود.



همانطور که مشاهده می‌کنید دو ویژگی *Zero Crossing* و *Mean Amplitude of R Peaks* به خوبی دو حالت را از یکدیگر جدا خواهند کرد. از این دو ویژگی برای کامل کردن *va_detect* استفاده خواهیم کرد.

(د) با توجه به نتایج قسمت (خ)، دو ویژگی را انتخاب کرده و برای هر یک، تابع *va_detect.m* را تکمیل کرده و بردار *alarm* (خروجی تشخیص داده شده توسط ویژگی مورد بررسی) را به دست آورید.

تابع *va_detect2* با این دو ویژگی به صورت زیر کامل شده است :

```
for i = 1:frame_N
    % Get the next data segment
    seg = ecg_data(((i - 1) * frame_step + 1):((i - 1) * frame_step + frame_length));

    % Calculate zero crossings
    zero_crossings = sum(abs(diff(seg > 0)));

    % Calculate mean amplitude of R peaks using findpeaks
    [pks, locs] = findpeaks(seg); % Local maxima
    [vals, val_locs] = findpeaks(seg); % Local minima
    vals = vals; % Convert minima back to positive values
    mean_R_amplitude = mean([pks; vals]);

    % Set alarm if any of the thresholds are exceeded
    if zero_crossings > zero_crossing_threshold || mean_R_amplitude > mean_R_amplitude_threshold
        alarm(i) = 1; % VFIB
    end
end
```

نتیجه ی پیش بینی این تابع به صورت زیر است که نشان می دهد *detector* به خوبی با این ویژگی ها پیش بینی لازم را انجام داده است (هر دو ویژگی به صورت یکسان و به درستی عمل خواهند کرد) :

Confusion Matrix:

8	0
0	11

Sensitivity: 1

Specificity: 1

Accuracy: 1

Confusion Matrix:

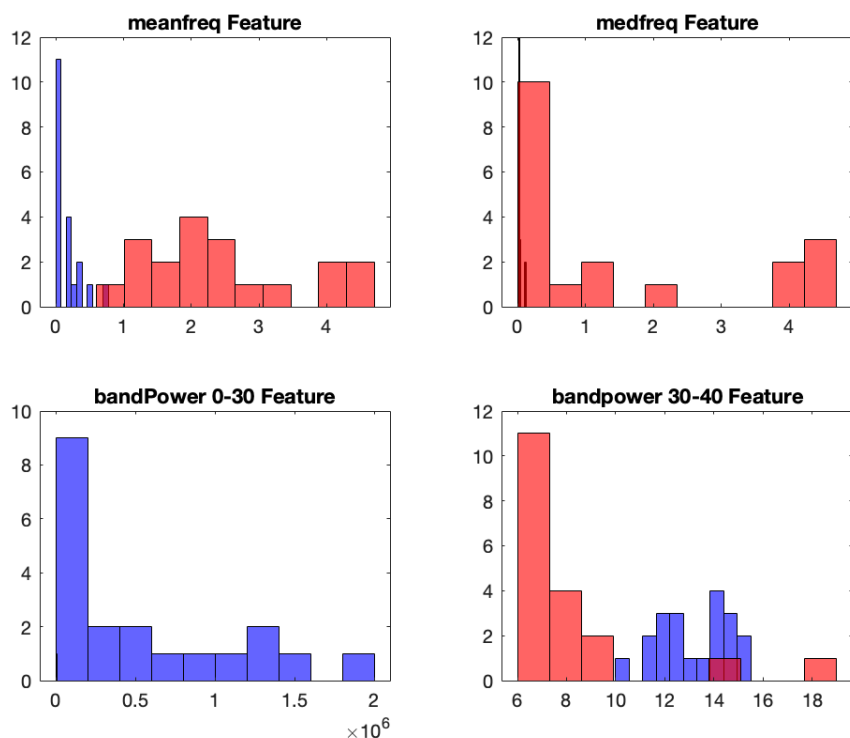
8	0
0	11

Sensitivity: 1

Specificity: 1

Accuracy: 1

(ر) بخش‌های (پ) تا (ذ) را بر روی داده دوم (n_424.mat) تکرار نمایید.



در ریزالت بالا دو ویژگی meanfreq و medfreq بهتر عمل خواهند کرد.

Feature = Mean Frequency

Sensitivity: 0.66667

Specificity: 1

Accuracy: 0.74359

Confusion Matrix:

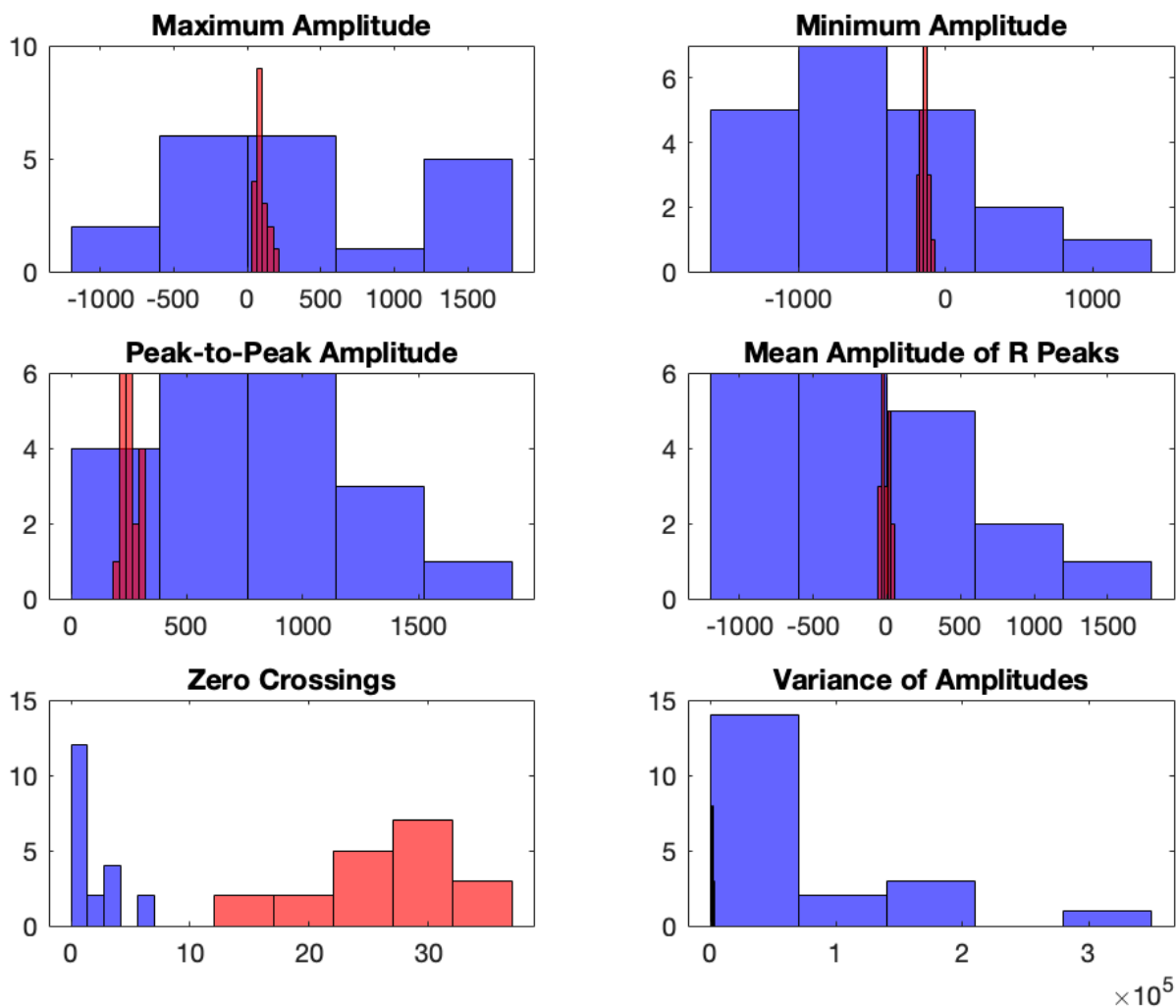
16	4
3	16

Feature = Median Frequency

Sensitivity: 0.84211

Specificity: 0.8

Accuracy: 0.82051



با توجه به ویژگی بالا می توان دو ویژگی Zero crossing و PEAK-PEAK را انتخاب کنیم و تابع detector را به کمک این دو ویژگی صحت سنجی کنیم.

Feature = Zero Crossings

Sensitivity: 1

Specificity: 1

Accuracy: 1

Confusion Matrix:

18	2
0	19

Feature = Peak-Peak Amplitude

Sensitivity: 1

Specificity: 0.90476

Accuracy: 0.94872

با تحلیل تمام آشکارسازها در حوزه زمان و فرکانس برای دو مجموعه داده، مشخص شد که برای داده اول، دو ویژگی فرکانسی، شامل میانه و توان باند فرکانسی 80-40 , 30-10 هرتز، به همراه میانگین قله‌های (R peak Avg)، بهترین عملکرد را در طبقه‌بندی دارند. در مقابل، برای داده دوم، تعداد عبور از صفر به عنوان بهترین ویژگی شناسایی شد.

ژ) بهترین آشکارساز به دست آمده برای داده اول را بر روی داده دوم اعمال کرده و نتیجه آن را بررسی کنید. همین کار را با بهترین آشکارساز به دست آمده برای داده دوم انجام دهید و آن را بر داده اول اعمال کنید.

frame_N = ابتدا بهترین آشکارساز به دست آمده برای داده اول
59 را بر روی داده دوم اعمال می‌کنیم:

Confusion Matrix:

12	8
5	14

Feature = Peak-Peak Amplitude

Sensitivity: 0.70588

Specificity: 0.63636

Accuracy: 0.66667

Confusion Matrix:

8	0
11	0

حال بهترین آشکارساز به دست آمده برای داده دوم

را بر روی داده اول اعمال می‌کنیم:

Sensitivity: 0

Specificity: 1

Accuracy: 0.42105

همانطور که می بینیم آشکارساز بر روی داده اول به خوبی روی داده دوم کارایی ندارد و به عکس. علت این مسئله داینامیک و تغییر نوع آریتمی از زمان به زمان و تحت شرایط مختلف و شخص به شخص می باشد و

نه متریک ثابتی برای دیتکت کردن آریتمی می توان یافت و نه آستانه ثابتی برای این ویژگی ها برای پیدا کردن آریتمی می توان پیشنهاد کرد. در نتیجه باید در زمان به زمان و تحت شرایط مختلف ویژگی های نامبرده شده صحت سنجی شود و بر اساس داینامیک سیگنال و همچنین آریتمی بهترین ویژگی انتخاب گردد.

س) از بین آشکارسازهای طراحی شده، یک آشکارساز را به عنوان بهترین آشکارساز انتخاب کرده و آن را بر حداقل یکی دیگر از سیگنال ها اعمال کنید و نتایج آن را بررسی کنید. به طور خاص، دو فایل آخر به شما این فرصت را می دهند که ببینید آیا سیستم شما در صورت وجود نویز زیاد، اما بدون آریتمی بطنی، هشدار کاذب (false alarm) تولید می کند یا خیر. آیا آشکارساز شما هشدار کاذب (false alarm)، تشخیص از دست رفته (missed detections) یا هر دو را تولید می کند؟ در چه شرایطی آشکارساز شما بیشتر مستعد خطا است؟

با توجه به نتایج قسمت های قبل ویژگی R-peak mean را انتخاب می کنیم و روی داده n_426 پیاده سازی را انجام می دهیم تا ببینیم تا چه حد به نویز زیاد مقاوم هست.

```
frame_N =
```

```
57
```

```
Confusion Matrix:
```

```
16    3
15   21
```

```
Feature = Peak-Peak Amplitude
```

```
Sensitivity: 0.51613
```

```
Specificity: 0.875
```

```
Accuracy: 0.67273
```

همانطور که می بینیم از نظر accuracy نه خیلی بد و نه خیلی خوب عمل می کند. بر اساس نتایج، دیتکتور در شرایط نویز زیاد بدون آریتمی هشدار کاذب تولید می کند (3 مورد مثبت کاذب) و همچنین دارای تشخیص های از دست رفته است (15 مورد منفی کاذب). sensitivity پایین (0.51613) نشان دهنده ضعف در شناسایی آریتمی های واقعی است، درحالی که specificity اندازه بیشتری دارد (0.875) و عملکرد مناسبی در

شناسایی موارد غیرآریتمی نشان می‌دهد. خطاها بیشتر در حضور نویز زیاد یا سیگنال‌های آریتمی ضعیف رخ می‌دهد.