

5/3/2024

آز پردازش سیگنال و تصاویر

پزشکی

گزارش آزمایش پنجم

محمدحسین شفیعی زادگان

99104781

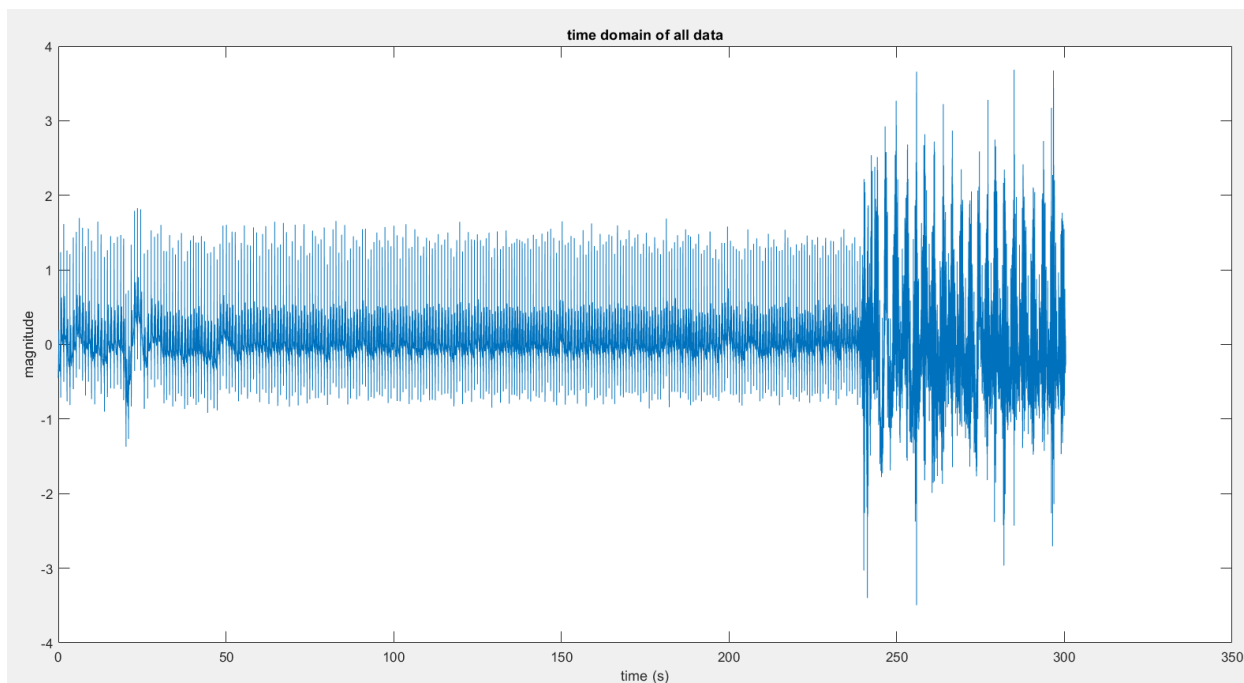
محمدجواد نوروزی

99102434

## فهرست مطالب

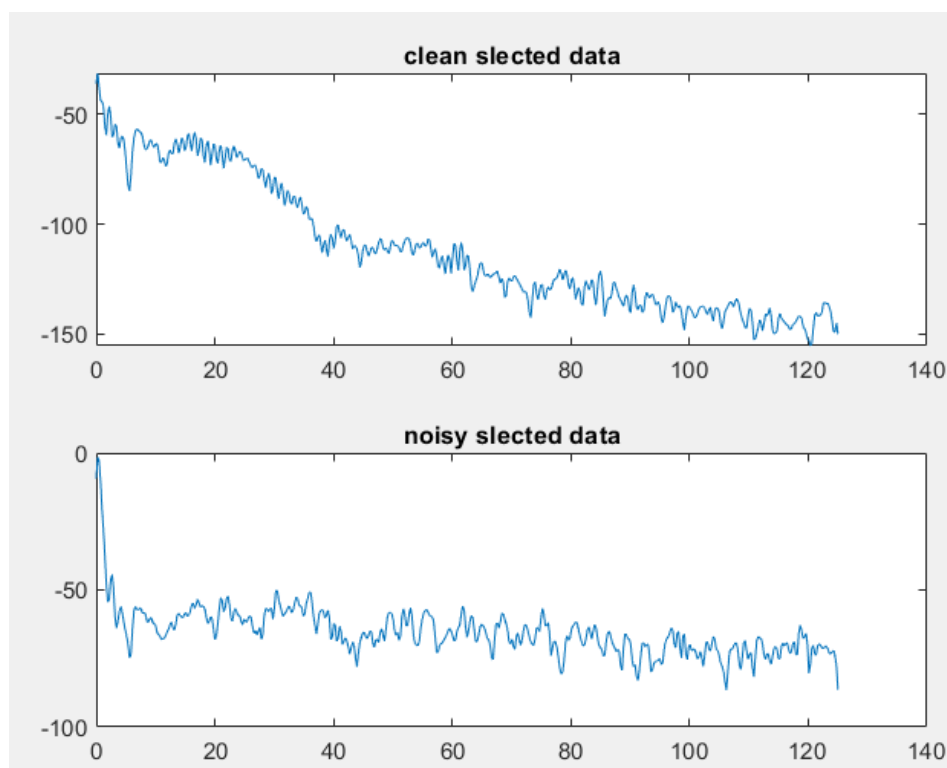
بخش اول: محدودسازی فرکانسی سیگنال/کاهش نویز.....	2
الف) محتوای فرکانسی از سیگنال تمیز.....	2
ب) فیلتر میان گذر.....	3
ج) کارایی فیلتر.....	4
بخش دوم: تشخیص آریتمی های بطنی.....	5
قسمت الف).....	5
قسمت ب).....	5
قسمت پ).....	6
قسمت ت).....	6
قسمت ث).....	6
قسمت ج).....	7
قسمت چ).....	7
قسمت ح).....	7
قسمت خ).....	8
قسمت د).....	8
قسمت ذ).....	8
قسمت ر).....	9
قسمت ز).....	10
قسمت ژ).....	11
قسمت س).....	11

## بخش اول: محدودسازی فرکانسی سیگنال/ کاهش نویز



### الف) محتوای فرکانسی از سیگنال تمیز

در شکل زیر مقایسه دو حالت توان سیگنال قلبی نرمال و نویزی را در مقیاس لگاریتمی مشاهده می کنیم:



همانطور که مشاهده می کنیم، توان حالت نویزی نسبت به حالت نرمال در همه فرکانس ها مقدار بزرگتری می باشد.

## ب) فیلتر میان گذر

همانطوری که در قسمت قبل مشاهده می شود، تفاوت توان در فرکانس های پایین به شدت زیاد میباشد، لذا یک فیلتر میان گذار با مشخصات زیر طراحی می کنیم:

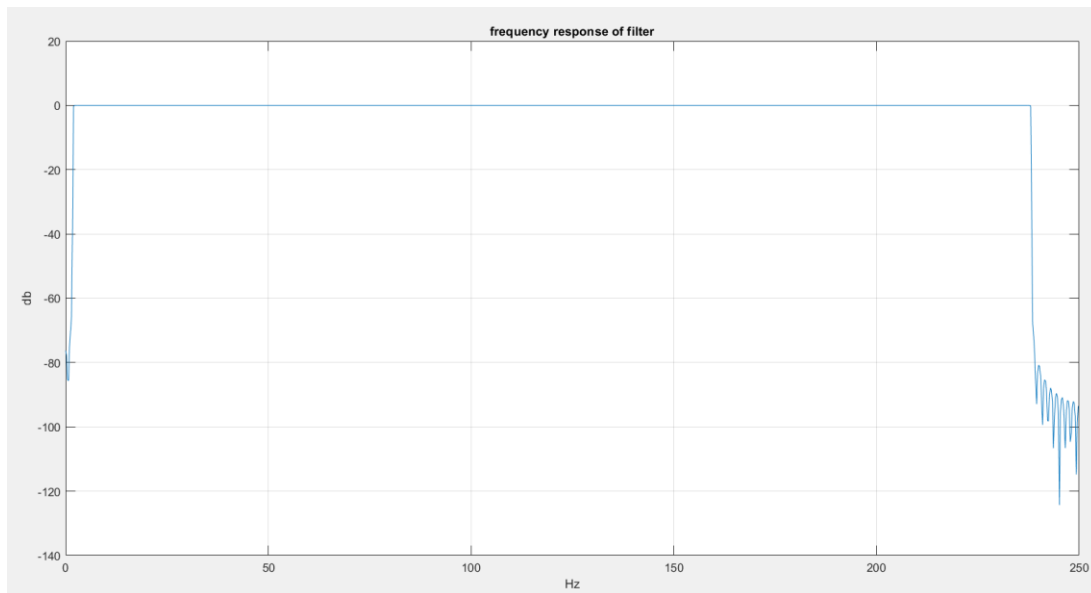
```
[y,d] = bandpass(data(:,2), [low_cut_freq, high_cut_freq], 2*fs)
```

که فرکانس های قطع پایین و بالای آن به شرح زیر هستند و به نحوی انتخاب شده اند که 90 درصد توان سیگنال (به جز توان baseline) را شامل شوند؛ در واقع فرکانس پایین را به صورت دستی تعیین کردیم و فرکانس بالا توسط کد نوشته شده محاسبه شد، به نحوی که شرط مذکور را ارضا کند:

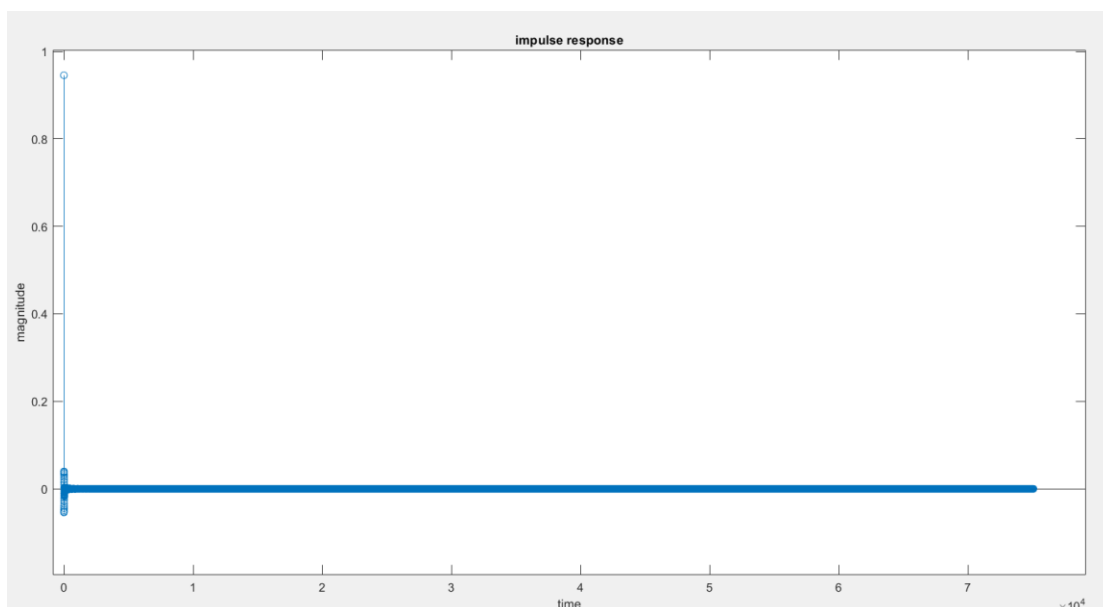
low\_cut\_freq=0.24414

high\_cut\_freq=57.8613

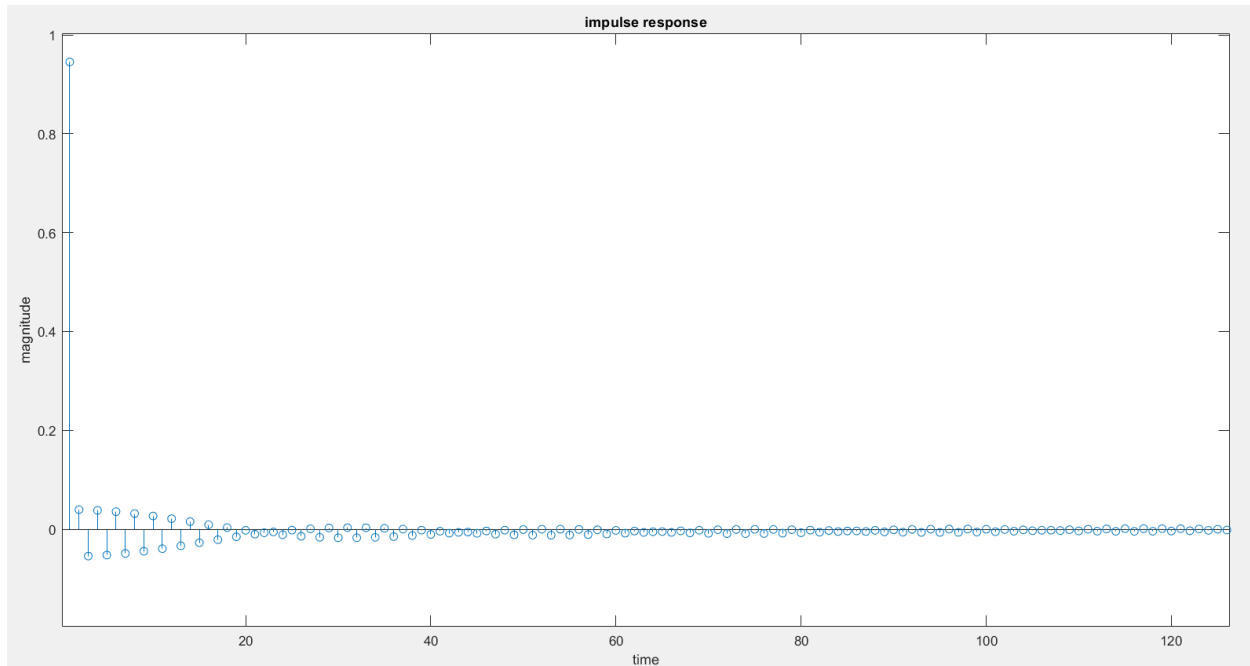
پاسخ فرکانسی:



پاسخ ضربه:

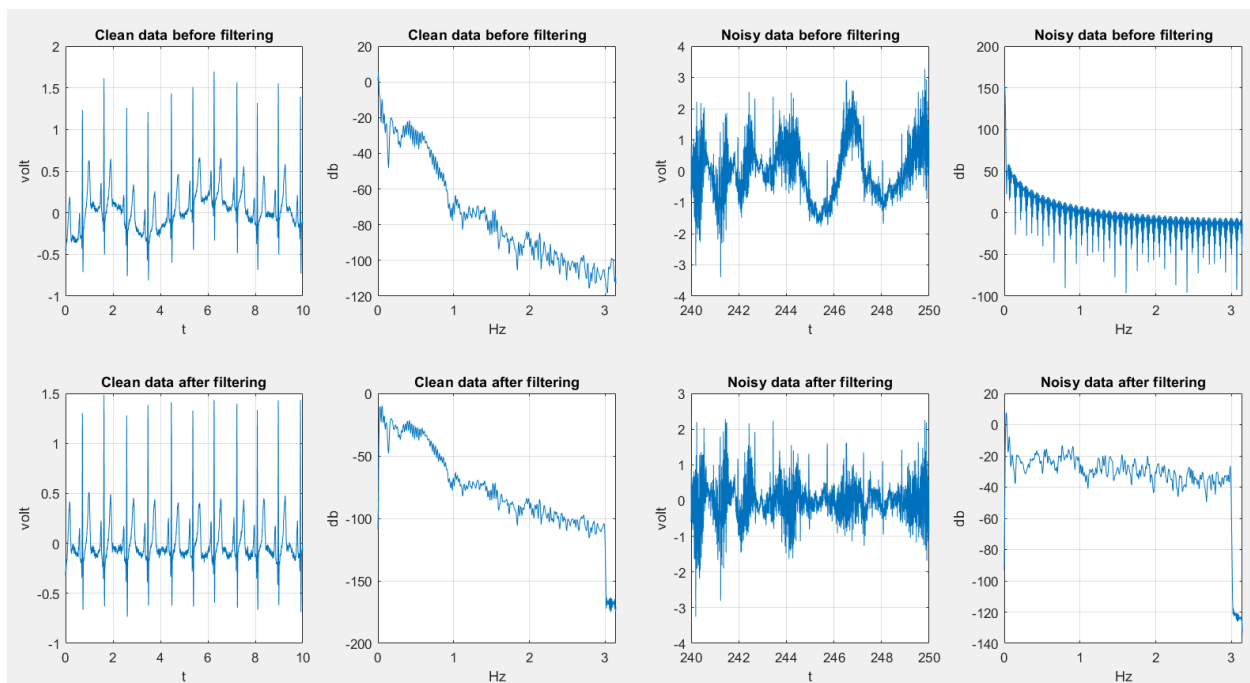


پاسخ ضربه (بزرگنمایی شده تصویر قبل):



### ج) کارایی فیلتر

فیلتر را به داده‌ها اعمال می‌کنیم. نتیجه برای داده‌های نویزی و نرمال در حوزه زمان و فرکانس بصورت زیر می‌باشد:

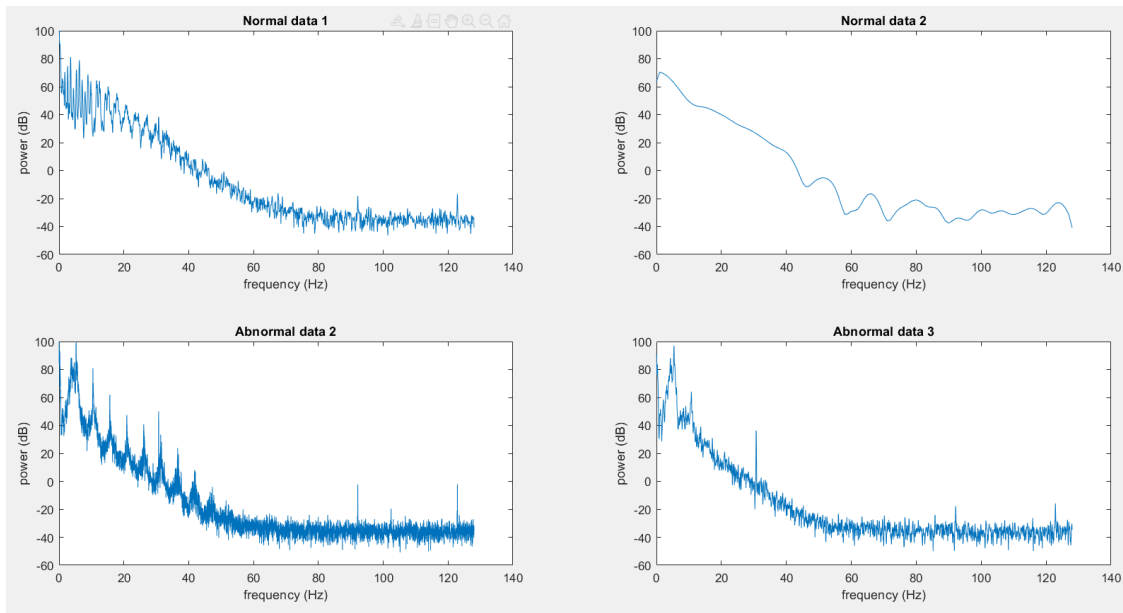


همانطور که می‌بینیم تا حد خوبی توانسته نویزهای سیگنال نویزی مورد نظر را حذف کند. همچنین ترم دی سی هر دو سیگنال هم حذف کرده است.

## بخش دوم: تشخیص آریتمی‌های بطنی

### قسمت الف)

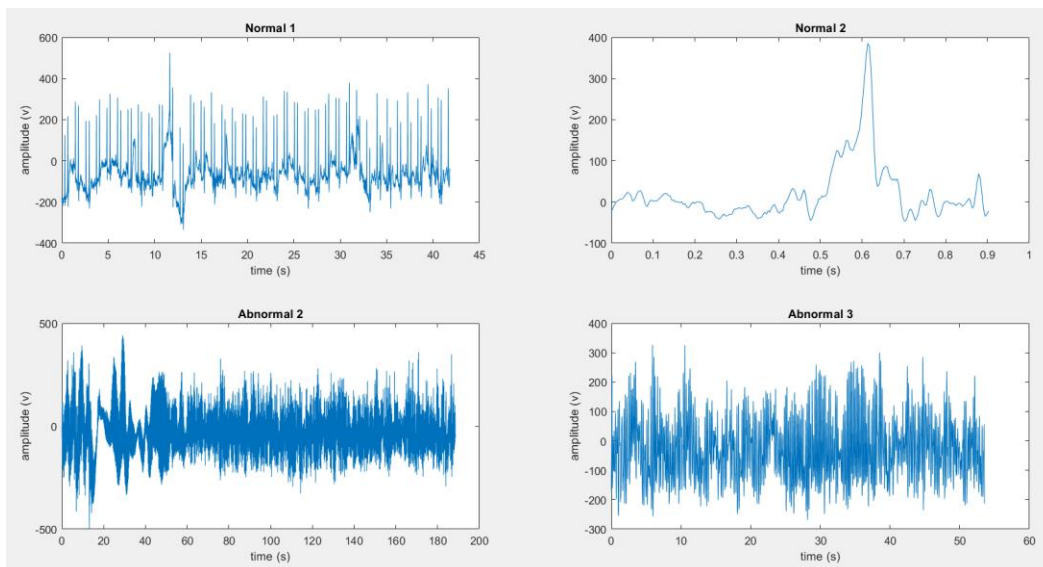
داده "n\_422.mat" را در متلب لود کرده و از آن استفاده می‌کنیم. با توجه به فایل توضیحات داده (زمان بندی سیگنال و لیبیل آن‌ها)، دو بخش نرمال و سه بخش غیرنرمال از سیگنال را جدا می‌کنیم. محتوای فرکانس سیگنال‌ها را در شکل زیر مشاهده می‌کنیم.



مشاهده می‌شود که توان سیگنال نویزی در اکثر فرکانس‌ها بیشتر از توان سیگنال نرمال می‌باشد. این گونه به نظر می‌رسد که نویز به صورت یک نویز سفید به سیگنال اعمال شده است. به این معنی که اسپکتروم سیگنال نرمال با اسپکتروم نویز سفید جمع شده است و اسپکتروم سیگنال نویزی به دست آمده است. پس توان سیگنال در باندهای مختلف می‌تواند در جداسازی این سیگنال‌ها مفید باشد.

### قسمت ب

سیگنال‌های انتخابی قسمت قبل را در حوزه زمان مشاهده می‌کنیم.



مشاهده می شود که سیگنال های غیرنرمال، دارای مولفه های فرکانس بالا با توان زیادی هستند و در شکل زمانی آن ها کمپلکس QRS مشاهده نمی شود.

### قسمت پ

با توجه به طول پنجره و میزان شیفت داده شده، سیگنال ها را جدا کرده و epoch می کنیم. با توجه به زمان یک epoch و بر اساس فایل اطلاعات داده، لیبل مربوط را مشخص می کنیم. بدین صورت، در انتها یک ماتریس دو بعدی خواهیم داشت که هر سطر آن سیگنال زمانی یک پنجره است و یک بردار به طول تعداد پنجره ها خواهیم داشت که حاوی لیبل آن هاست.

### قسمت ت

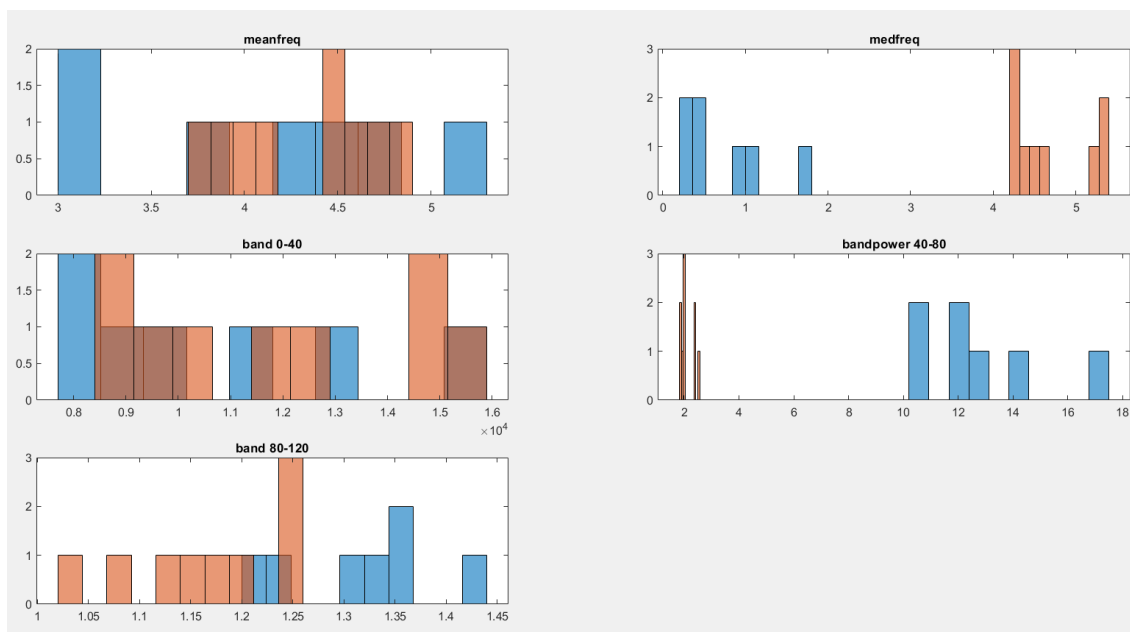
برای پنجره ها، 5 ویژگی فرکانسی زیر را محاسبه می کنیم و نتایج را در ماتریس ویژگی ها که هر سطر آن 5 ویژگی یک پنجره را نشان می دهد، ذخیره می کنیم.

Features:

- Mean frequency
- Median frequency
- Power at band 0-20 Hz
- Power at band 20-40 Hz
- Power at band 40-80 Hz

### قسمت ث

حال برای هر ویژگی یک هیستوگرام رسم می کنیم.



مشاهده می شود که ویژگی های median frequency و power at 40-80 Hz به خوبی دو نوع سیگنال را از هم تفکیک می کنند.

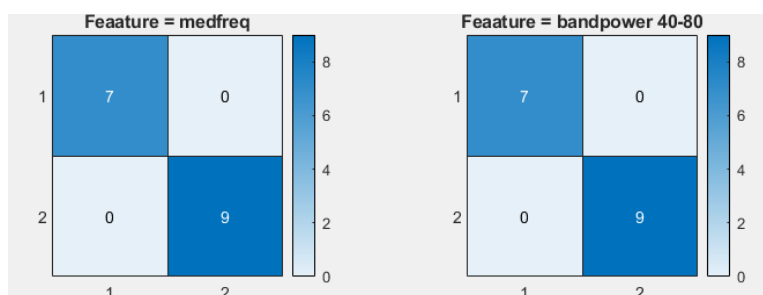
## قسمت ج

با توجه به هستوگرام های قسمت قبل برای دو ویژگی انتخابی یک مقدار آستانه در نظر گرفته و به صورت زیر، تابع "va\_detect.m" را تکمیل می کنیم.

```
if feature == "medfreq"
    f_val = medfreq(seg, Fs);
    if f_val > 3
        alarm(i) = 1;
    else
        alarm(i) = 0;
    end
elseif feature == "bandpower40"
    f_val = bandpower(seg, Fs, [40 80]);
    if f_val > 5
        alarm(i) = 0;
    else
        alarm(i) = 1;
    end
end
end
```

## قسمت چ

حال برای دو کلاس موردنظر معیار های طبقه بندی را محاسبه و گزارش می کنیم.



### Command Window

```
Feature = median frequency
accuracy = 1.000000 , sensitivity = 1.000000 , specificity = 1.000000
```

```
Feature = bandpower at 40-80 Hz
accuracy = 1.000000 , sensitivity = 1.000000 , specificity = 1.000000
```

```
fx >> |
```

به خوبی عملکرد طبقه بند را مشاهده می کنیم.

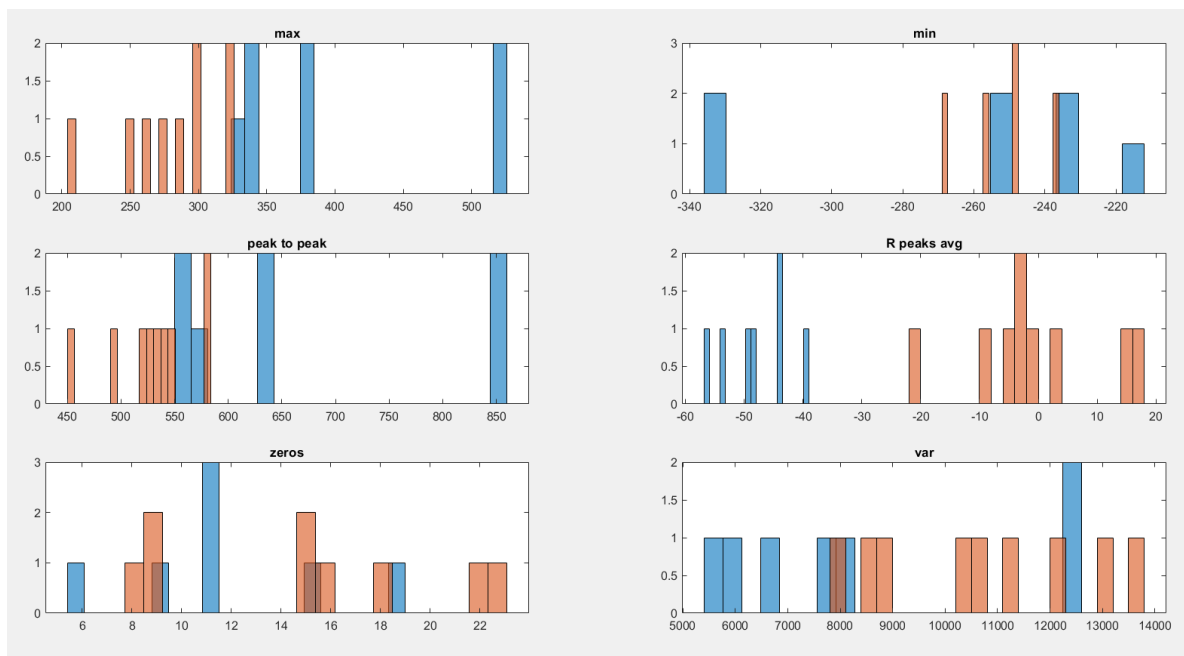
## قسمت ح

برای پنجره ها 6 ویژگی مورفولوژیک شامل ماکزیمم، مینیمم، دامنه peak to peak ، میانگین دامنه قله ها، تعداد عبور از صفر و واریانس سیگنال را محاسبه می کنیم.



## قسمت خ

هیستوگرام هر ویژگی به صورت زیر خواهد بود.



با توجه هیستوگرام های بالا، می توان از ویژگی های ماکزیمم دامنه و میانگین دامنه قله ها (موج R) استفاده کرد.

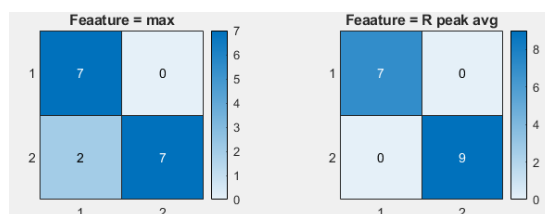
## قسمت د

برای این دو ویژگی جدید، تابع "va\_detect" را کامل می کنیم. مقادیر آستانه از روی هیستوگرام ها به دست آمده اند.

```
elseif feature == "max"
    f_val = max(seg);
    if f_val > 320
        alarm(i) = 0;
    else
        alarm(i) = 1;
    end
elseif feature == "R-peak-avg"
    f_val = mean(findpeaks(sig));
    if f_val > -30
        alarm(i) = 1;
    else
        alarm(i) = 0;
    end
end
```

## قسمت ذ

حال برای دو کلاس موردنظر معیار های طبقه بندی را محاسبه و گزارش می کنیم.



```

Feature = max
accuracy = 0.875000 , sensitivity = 0.777778 , specificity = 1.000000

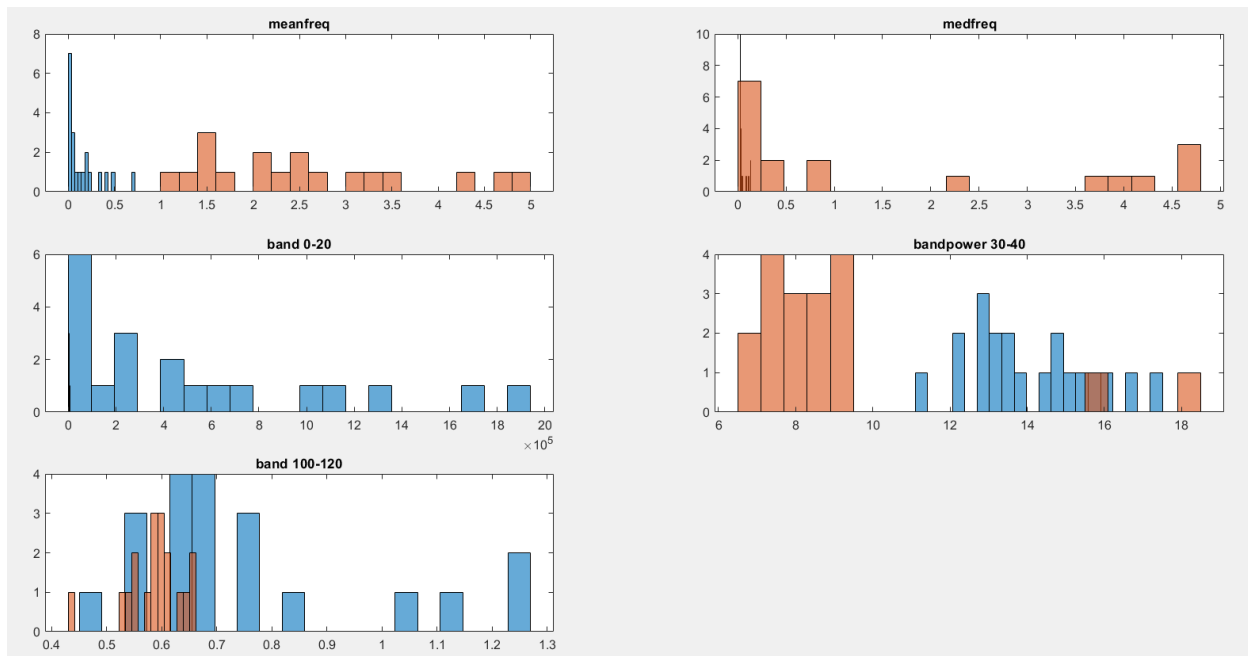
Feature = R peak average
accuracy = 1.000000 , sensitivity = 1.000000 , specificity = 1.000000

```

*fx* >>

## قسمت ر

حال قسمت های قبل را برای داده "n\_424.mat" تکرار می کنیم. کدهای این بخش در فایل مجزا هستند.

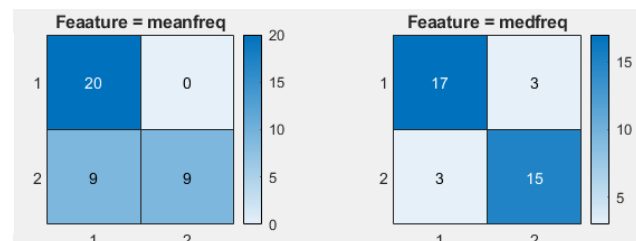


ویژگی های mean frequency و median frequency انتخاب می شوند.

```

elseif feature == "meanfreq"
    f_val = medfreq(seg, Fs);
    if f_val > 0.5
        alarm(i) = 1;
    else
        alarm(i) = 0;
    end
end

```



## Command Window

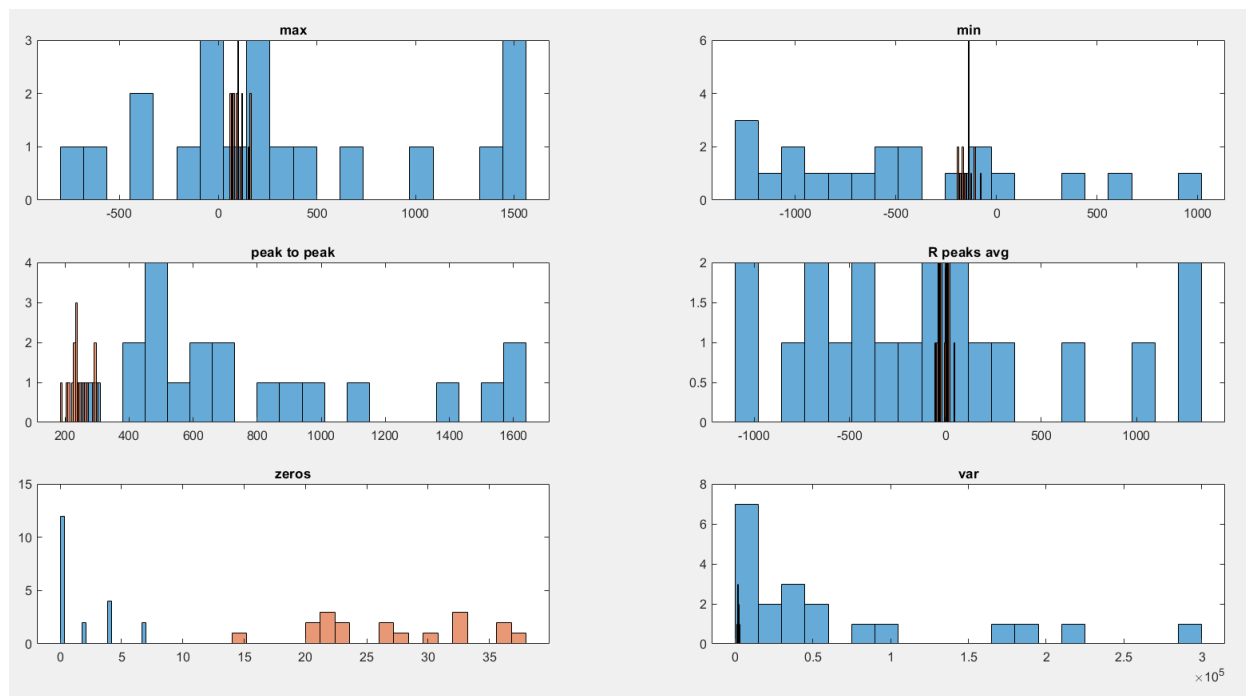
```

Feature = mean frequency
accuracy = 0.763158 , sensitivity = 0.689655 , specificity = 1.000000

Feature = median frequency
accuracy = 0.842105 , sensitivity = 0.850000 , specificity = 0.833333

```

*fx* >>



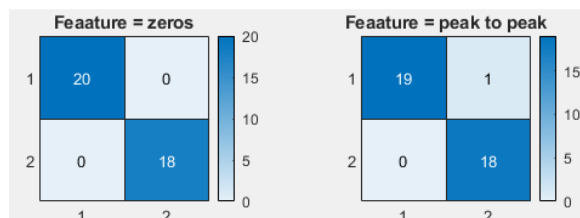
ویژگی های تعداد عبور از صفر و دامنه peak to peak را می توان در نظر گرفت.

```
elseif feature == "zeros"
    f_val = sum(seg == 0);
    if f_val > 10
        alarm(i) = 1;
    else
        alarm(i) = 0;
    end
elseif feature == "peak-to-peak"
    f_val = max(sig) - min(sig);
    if f_val > 350
        alarm(i) = 0;
    else
        alarm(i) = 1;
    end
end
```

```
Feature = zeros
accuracy = 1.000000 , sensitivity = 1.000000 , specificity = 1.000000
```

```
Feature = peak to peak
accuracy = 0.973684 , sensitivity = 1.000000 , specificity = 0.947368
```

```
>>
```



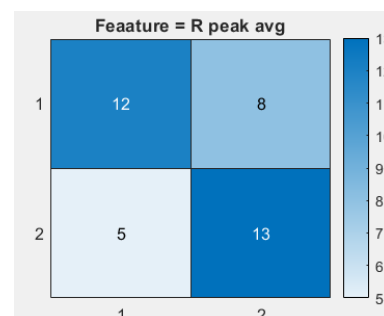
## قسمت ز

با در نظر گرفتن تمام آشکارساز ها چه در حوزه زمان و چه در حوزه فرکانس برای دو داده، می توان گفت برای داده اول بهترین ویژگی برای طبقه بندی هر دو ویژگی فرکانسی انتخاب شده ( میانه و توان باند 40 – 80 Hz ) و R peak avg می باشد. برای داده دوم بهترین ویژگی تعداد عبور از صفر می باشد.

## قسمت ژ

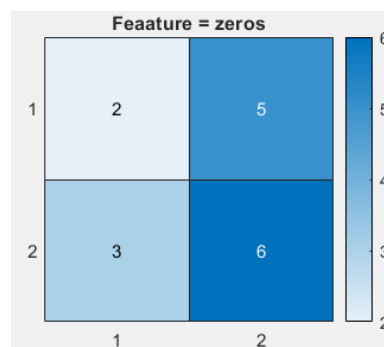
ابتدا بهترین آشکارساز به دست آمده برای داده اول را بر روی داده دوم اعمال می کنیم. از ویژگی R peak avg استفاده می کنیم.

```
Data 2 - Feature = R peak avg
accuracy = 0.657895 , sensitivity = 0.705882 , specificity = 0.619048
fx >>
```



حال بالعکس

```
Data 1 - Feature = zeros
accuracy = 0.500000 , sensitivity = 0.400000 , specificity = 0.545455
fx >>
```

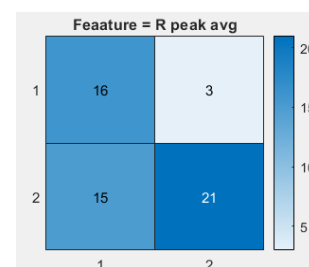


مشاهده می شود عملکرد طبقه بند ها به شدت افت می کند که کاملاً منطقی است چرا که مسخره ترین روش برای طبقه بندی و به صورت دستی (نه هوشمند و اتوماتیک (به ویژه در انتخاب آستانه)) استفاده شده است. روش های استفاده شده اصلاً بر پایه هوش مصنوعی نیستند!!!

## قسمت س

نتایج را برای داده "n\_426.mat" دوباره به دست می آوریم. کدهای این قسمت در فایل مجزا است. از ویژگی R peak avg استفاده می کنیم.

```
Data 2 - Feature = R peak avg
accuracy = 0.672727 , sensitivity = 0.516129 , specificity = 0.875000
fx >>
```



مشاهده می کنیم با وجود دقت نسبتاً مناسب، هشدار کاذب و تشخیص از دست رفته نیز مشاهده می شود.