باسمه تعالی

دانشگاه صنعتی شریف

دانشکده مهندسی برق

هوش محاسباتی

فاز اول پروژه درس

علی محرابیان96102331

استاد: دکتر رضایی

بهار1399

بخش اول:آشنایی با بیماری ها

بیماری های ناشی از قلب یکی از رایج ترین علل مرگ در سراسر جهان بوده،به طوری که حدود 31% از از مرگ و میرها به دلیل آن است.بررسی ها نشان می دهد که بیماری های قلبی که ارتباط تنگاتنگی با cardiac arrhythmias دارند،سلامت مردم را در مخاطره قرار داده است.یکی از راه هایی که برای شناسایی arrhythmias استفاده می شود،سیگنال های ECG هستند.هدف این پروژه،شناسایی یک سری از بیماری های قلبی به کمک این سیگنال ها است.در شکل زیر به طور موجز،نحوه ضبط این سیگنال ها نشان داده شده است.

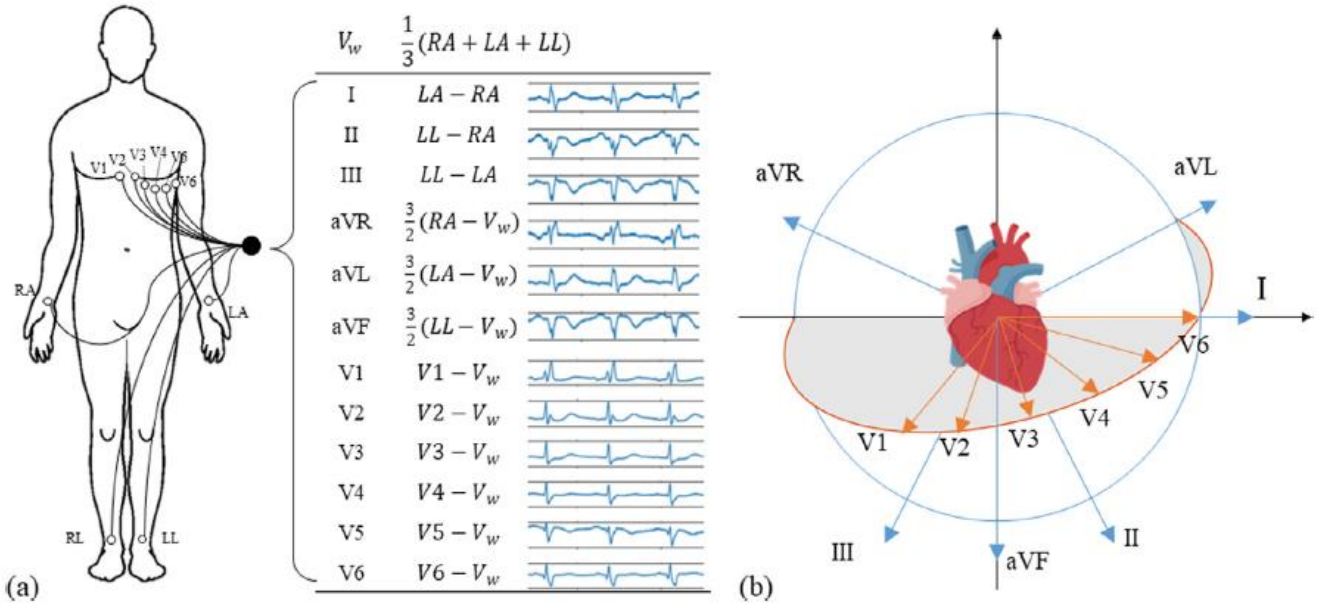
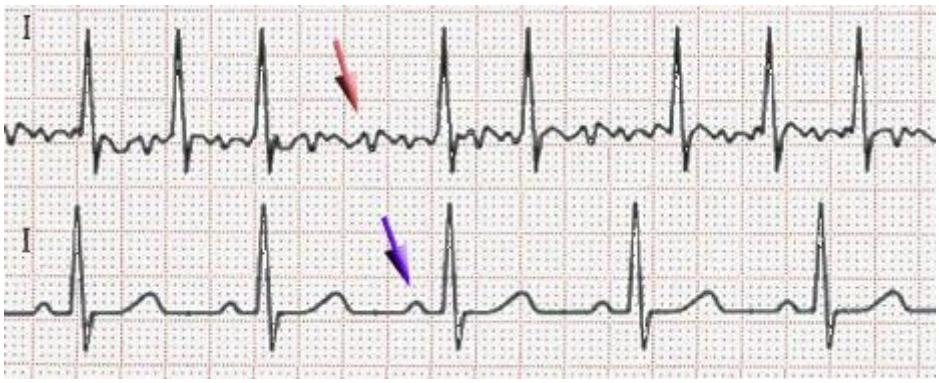


Fig. 1. Illustration for 12-lead ECG system. a) Spatial placement of 10 electrodes used by 12-lead ECG system. Three limb leads and three augmented limb leads as well as six precordial(chest) leads were formed between physical electrodes and a virtual electrode known as Wilson's central terminal. b) Representing electrical potential differences between electrodes placed on human skin, ECG signals from 12 leads reflect electrical activity of heart from different spatial angles.

در ادامه به طور مختصر به بررسی بیماری ها می پردازیم.

:Atrial fibrillation(AF)

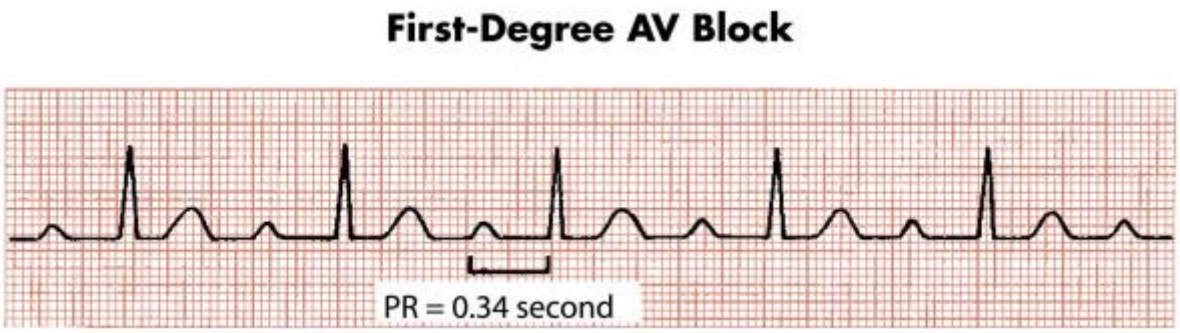
این بیماری ناشی از فقدان پیک های P و جایگزین شدن آن با سیگنال های نویزی است.همچنین فاصله نامنظم بین پیک های R-R هم از دلایل آن است.ولی به دلیل تفاوت های جزئی در شکل سیگنال ها،تشخیص آن به کمک چشم دشوار است.



پیکان بنفش نشان دهنده پیک P است که در سیگنال بالایی به دلیل نویز قابل تشخیص نیست.

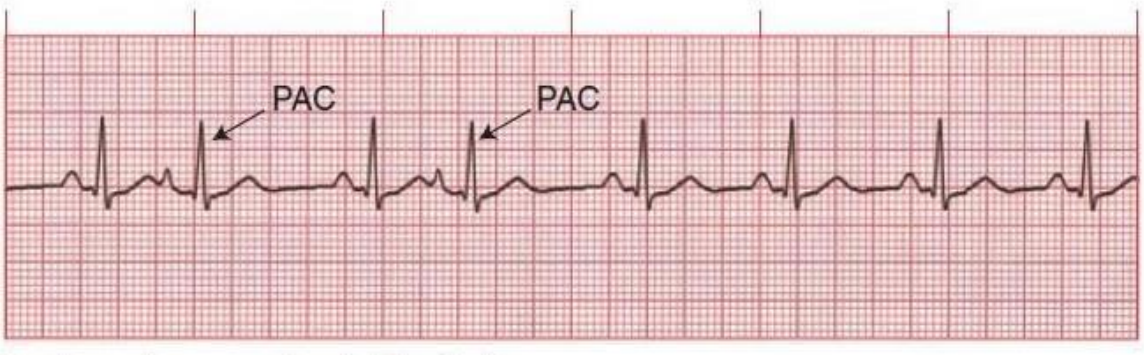
:First degree atrioventricular block

علت آن طولانی شدن فاصله بین پیک های R و P و رسیدن به بیش از 200ms است در حالی که در بزرگسالان این فاصله باید بین 120 تا 200 باشد.



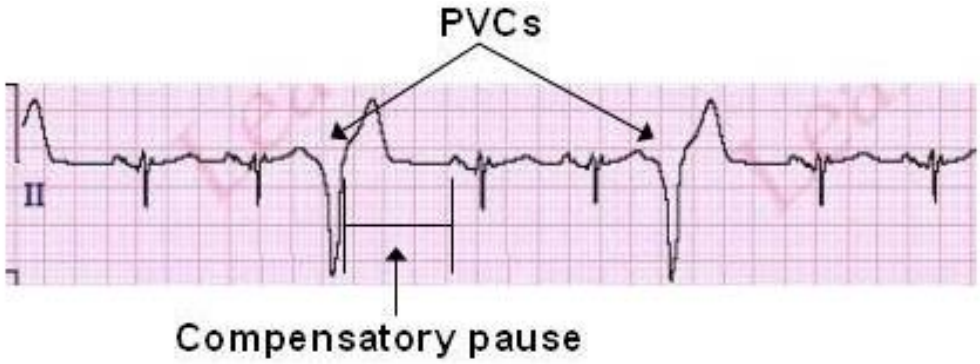
:Premature atrial contractions(PAC)

به طور کلی بر اساس premature heartbeats که از ناحیه atria قلب نشأت می گیرد،به وجود می آید.شاخصه آن حالت غیرعادی پیک P است.



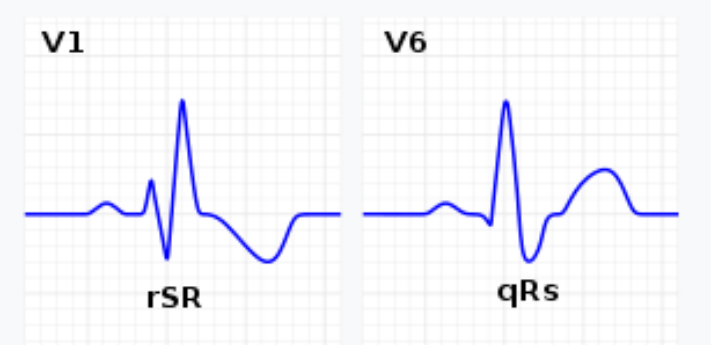
:Premature ventricular contractions(PVC)

ضربان قلب اضافی که در یکی از بطن ها شروع می شود و ریتم طبیعی قلب را به هم می ریزد.به طور کلی فرم QRS آن نسبت به PAC فرق می کند و compensatory pause آن طولانی تر است.



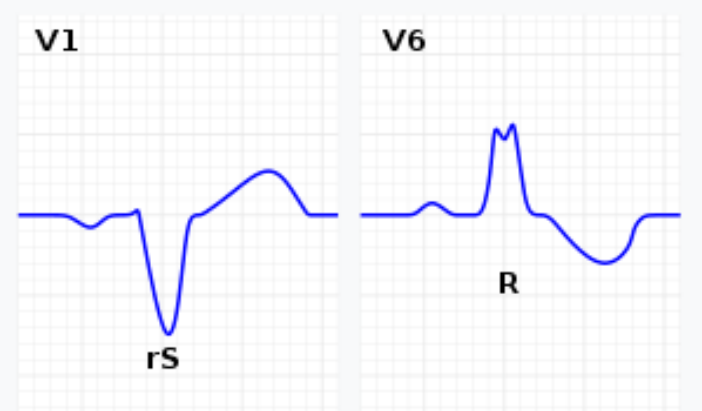
:Right bundle branch block(RBBB)

به طور کلی بطن راست از طریق سیگنال های ضربه ای که توسط right bundle branch منتقل می شود،فعال نمی شود.از علایم آن می توان به وجود یک terminal R wave در lead 1 و یک تاخیر در wave S در Lead 1,6 اشاره کرد.



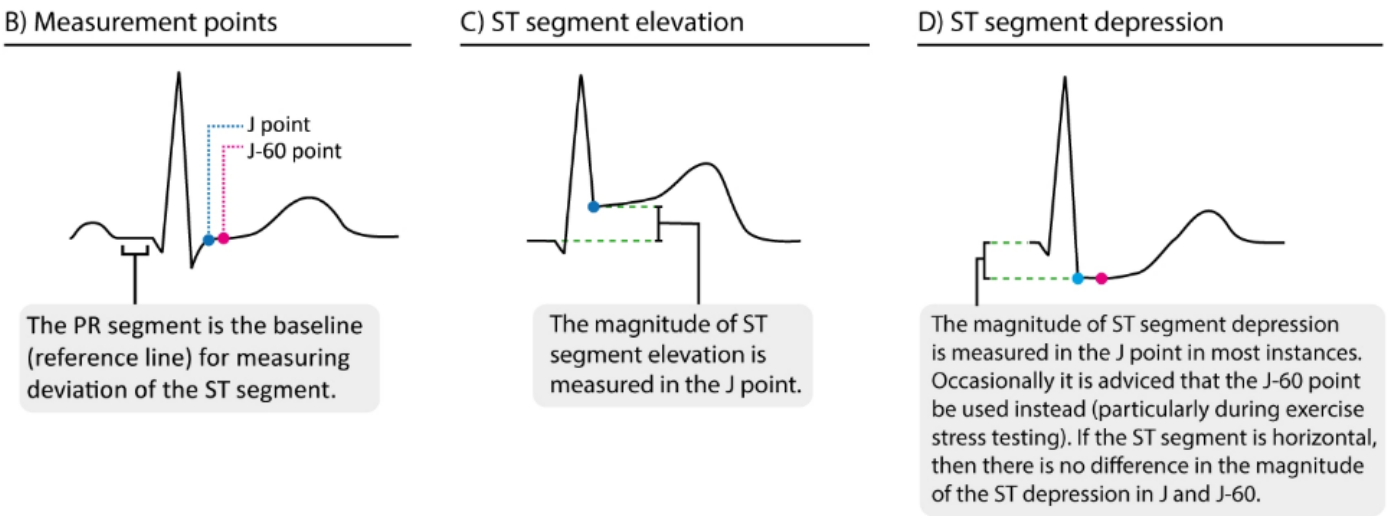
:Left bundle branch block(LBBB)

به طور کلی بطن چپ نسبت به بطن راست دیرتر منقبض می شود.از علایم آن این است که طول بازه زمانی QRS بیشتر از 120ms باشد.پیک R در lead 6 به صورت شکل 'M' مشاهده می شود.



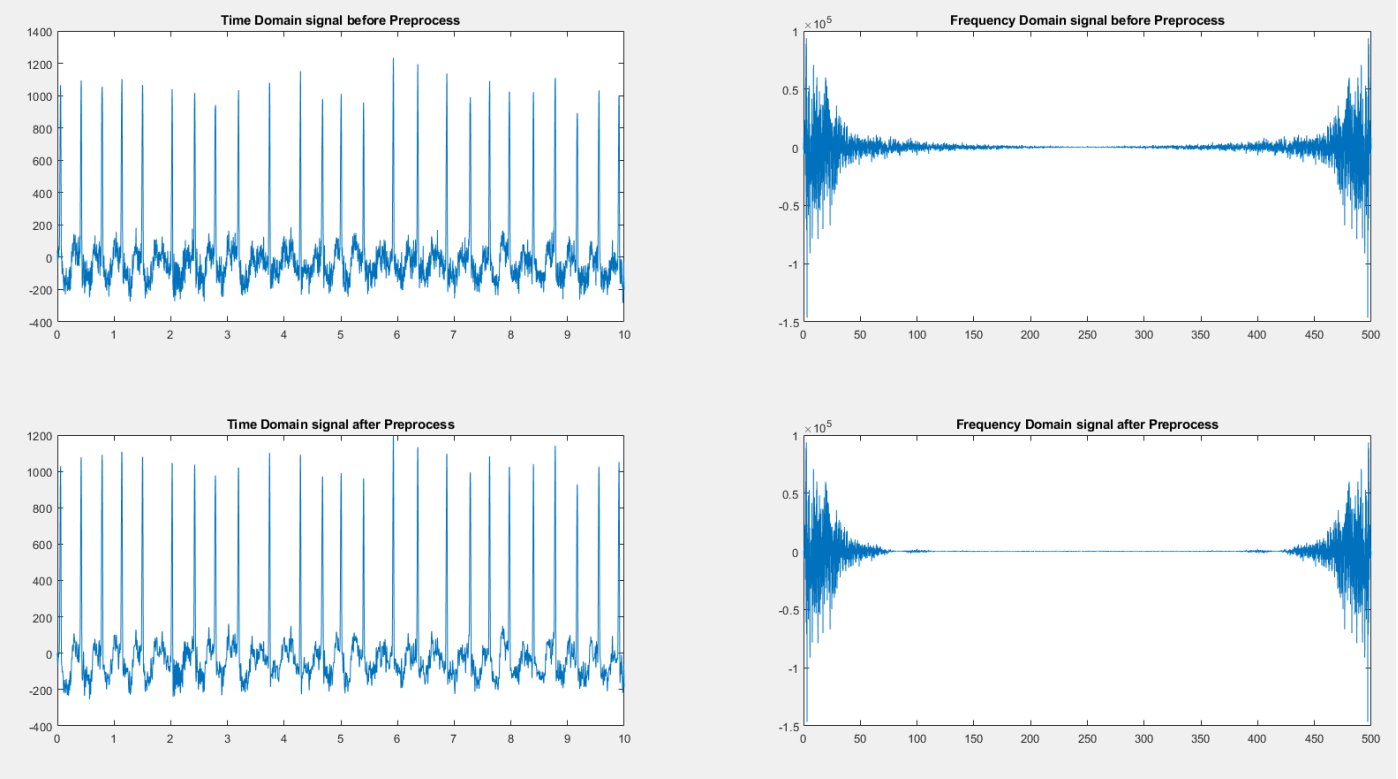
:STD & STE

در STE،بازه ST به طور غیرنرمال زیر baseline بوده و در STE،بالای این خط است.



بخش دوم:پیش پردازش داده ها

برای حذف فرکانس های بالا از فیلتر Savitzky-Golay استفاده می کنیم.این فیلتر مقدار در هر نقطه را با میانگین محلی نقاط اطراف ان جایگزین می کند که باعث نرم شدن سیگنال می شود.برای حذف فرکانس های پایین،یک چند جمله ای با درجه کم به سیگنال خود فیت کرده،در نتیجه تغیرات آهسته سیگنال دنبال می شود. حال تفاضل این چندجمله ای از سیگنال را محاسبه می کنیم.این عمل باعث حذف مقادیر و نویز های نامطلوب می شود.در تمامی مراحل فرکانس نمونه برداری را برابر با 500Hz در نظر گرفتیم.سیگنال های lead 1 نفر سوم آزمایش در زیر قبل و بعد از پیش پردازش رسم شده اند.



یکی دیگر از مواردی که در بررسی لیبل های بیماران مشاهده شد،این بود که بعضی از بیماران دارای چند لیبل خروجی بودند.برای این بیماران بدون بایاس ذهنی،یکی از لیبل ها را به دلخواه انتخاب کردیم.بنابراین با توجه به این مشاهده،ممکن است به هنگام به دست آوردن خروجی داده های تست،بعضی افراد بیشتر از یک بیماری داشته باشند که موضوع چندان عجیبی نیست.

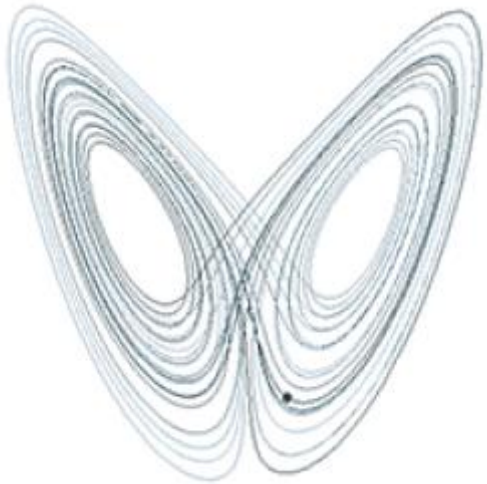
بخش سوم:انتخاب ویژگی ها

در این پروژه سعی کرده ایم که علاوه بر استفاده از ویژگی های تیپیکال در مطالعه سیگنال های ECG،برای اولین بار از ویژگی های آشوبی(chaos) استفاده کنیم.

سیستم های دینامیکی را می توان به دو دسته کلی تقسیم کرد:خطی و غیرخطی.

به طورکلی تمام سیستم های واقعی غیرخطی هستند ولی اکثر اوقات به عنوان اولین تخمین برای این دینامیک ها سعی بر این است که از مدل های خطی برای آن ها استفاده شود.تنها سیستم های غیرخطی قادرند رفتارهای غیرقابل پیش بینی از خود نشان دهند.در سال 1963 ،ادوارد لورنز نتایج تحقیقاتش را تحت عنوان "جریان های یقینی غیرمتناوب" منتشر کرد که رفتار یک مدل ریاضی ساده شده از آب و هوا را نشان می داد.

او نشان داد که چه طور یک مدل نسبتا ساده ریاضی می تواند رفتارهای غیرقابل پیش بینی از خود نشان دهد.خروجی های مدل حتی بعد از گذشت زمان های طولانی هیچ الگوی تکرارشونده ای نداشتند.همین رفتار غیرپریودیک و غیرقابل پیش بینی امروزه تحت عنوان آشوب(chaos)شناخته می شود.آشوب یک خاصیت ریاضی از پاسخ زمانی ،مجموعه ای از معادلات و پارامترها است که تنها در سیستم های با دینامیک غیرخطی یافت می شود.



دینامیک آشوبی،دینامیکی است که به شرایط اولیه حساسیت داشته باشد که به اثر پروانه ای(butterfly effect) مشهور است به شرطی که کراندار باشد.در سیستم های عادی،اگر تفاوت شرایط اولیه بسیارکم باشد،حالت نهایی سیستم به ازای آن دو شرط اولیه متفاوت،تقریبا مشابه است.ادر سیستم های آشوبی،پاسخ نهایی سیستم به ازای این شرایط اولیه،دورتر و دورتر می شود.معادلات دیفرانسیل آشوب در دینامیک پیوسته به صورت زیر است.

$$\begin{cases} \frac{dx}{dt} = f(x,y,z) \\ \frac{dx}{dt} = f(x,y,z) \\ \frac{dz}{dt} = f(x,y,z) \end{cases}$$

در ادامه چند مورد از ویژگی هایی که استفاده کرده ایم را توضیح می دهیم.چون ویژگی های آشوبی جدید و در حال توسعه هستند، در انتهای گزارش،لیست مقالات و کتاب های مربوط و استفاده شده را جهت تفحص بیشتر قرار داده ایم.

1.نمای لیاپانوف(Lyapunov Exponent)

برای سری زمانی یک بعدی$(x_0, x_1, x_2, \dots)$ اگر فاصله بین نمونه های ما τ باشد$(t_n - t_0 = n\tau)$ نمای لیاپانوف به صورت زیر تعریف می شود.

اگر یک مقدار از سری زمانی انتخاب کنیم مانند x_i و از سری یک مقدار دیگر مانند x_j انتخاب کنیم به گونه ای به مقدار اول نزدیک باشد: $d_0 = |x_i - x_j|,\ d_1 = |x_{i+1} - x_{j+1}|,\ d_2 = |x_{i+2} - x_{j+2}|, \dots$

حال اگر فواصل به طور میانگین با افزایش n به طور نمایی افزایش یابد.$(d_n = d_0 e^{\lambda n})$

مقدار λ را نمای لیاپانوف می گویند.اگر این مقدار مثبت باشد،رفتار سیستم آشوبی است.

2.نمای هرست(Hurst exponent)

به طور کلی این ویژگی معیاری برای سنجش حافظه بلندمدت سری های زمانی است و autocorrelation سری ها را به هم مربوط می کند.برای تخمین نمای هرست،سری زمانی به سری های جزئی تری تقسیم شده و وابستگی آن ها سنجیده می شود.برای یک زیر سری به طول n،به صورت زیر محاسبه می شود.$(x_0, x_1, \dots, x_{n-1}, x_n)$

$$m = mean(x_0, \dots, x_n) \rightarrow y_t = x_t - m \rightarrow z_t = n * mean(y_1, \dots, y_n)$$

$$R(n) = \max(z_1, \dots, z_n) - \min(z_1, \dots, z_n) \quad S(n) = \sqrt{var(x_1, \dots, x_n} \rightarrow \quad H = \frac{R(n)}{S(n)}$$

با گرفتن Expected روی مقادیر به دست آمده،مقدار مطلوب به دست می آید.الگوریتم فوق را در پایتون پیاده سازی کردیم که فایل آن به طور جداگانه پیوست شده است.

Higuchi.3

یکی از الگوریتم های مورد استفاده برای محاسبه fractal dimension می باشد که معیاری برای سنجش Complexity سیگنال در حوزه زمان است.پارامتر این الگوریتم k_{max} می باشد که با مطالعه مقاله های متفاوت،مقدار آن را برابر با 3 در نظر گرفتیم.مقدار این پارامتر به صورت زیر محاسبه می شود.

$$L_m(k) = \frac{1}{k} * \left(\left(\sum_{i=1}^{i=int\left(\frac{n-m}{k}\right)} |x(m+ik) - x(m+(i-1)k)| \right) * \frac{n-1}{k * int\left(\frac{n-m}{k}\right)} \right)$$

$$L(k) = \frac{\sum_{m=1}^{k} L_m(k)}{m} \quad \rightarrow \quad HDF = -\frac{\ln(L(k))}{\ln(k)}$$

الگوریتم زیر را پیاده سازی کرده و فایل متلب را پیوست کرده ایم.

sample entropy.4

یکی از معیار هایی است که برای سنجش میزان نوسانات و غیرقابل پیش بینی بودن سیگنال استفاده می شود.

برای سیگنال به طول N، و پارامتر های m,r،مقدار آن به صورت زیر محاسبه می شود.

$$x_N = (x_1, \dots, x_n) \quad x_m(i) = (x_i, \dots, x_{i+m-1}) \quad d(x,y) = \max(|x-y|)$$

A را به صورت تعداد زوج مرتب هایی که $d\left(x_{m+1}(i), x_{m+1}(j)\right) < r$ باشد تعریف می کنیم.

B را به صورت تعداد زوج مرتب هایی که $d\left(x_m(i), x_m(j)\right) < r$ باشد تعریف می کنیم.

نسبت $-\log\frac{A}{B}$ آنتروپی نمونه تعریف می کنیم.معمولا m=2 و r را برابر با 0.2 واریانس سیگنال در نظر می گیرند. کد این قسمت به زبان پایتون نوشته شده است که در فایل های پیوست شده می باشد.

Energy.5

در این قسمت به کمک تبدیل wavelet،محتواهای سیگنال در فرکانس های مختلف را به دست می آوریم و انرژی سیگنال را در بازه های متفاوت فرکانسی حساب می کنیم.سیگنال های detailed،محتوای فرکانس بالا را دارا می باشند و approximation،محتوای فرکانس های میانی و پایین را دارا هستند.

petrosian .6

یکی دیگر از الگوریتم های محاسبه fractal dimension می باشد که به کمک رابطه زیر محاسبه می شود.

$$p = \frac{log_{10}^{N}}{log_{10}^{N} + log_{10}^{\left(\frac{N}{N+0.4N_{\delta}}\right)}}$$

که Nطول سیگنال و N_{δ} تعداد تغییر علامت های سیگنال مشتق است.

katz.7

از دیگر الگوریتم های محاسبه fractal dimension است که از رابطه زیر محاسبه می شود.

$$D = \frac{LogN}{LogN + Log(\frac{d}{L})}$$

N برابر با طول سیگنال است و بقیه مقادیر از رابطه زیر محاسبه می شوند.

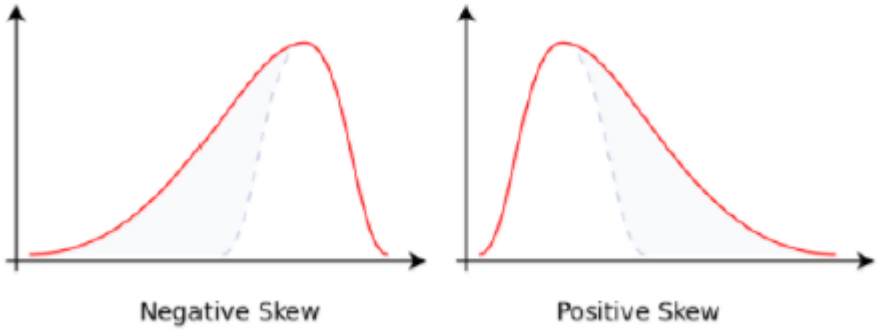
$$d = \max(|x_i - x_1|)\ for\ i \epsilon 1, \dots, n \qquad L = \sum_{I=1}^{N-1} |x_{i+1} - x_i|$$

حال از ویژگی های تیپیکالی که در کارهای پردازش سیگنال استفاده می شود نیز بهره می گیریم.

Skewness.8

چولیدگی یک ویژگی آماری است که نشان گر عدم تقارن تابع چگالی احتمال یک متغیر تصادفی است که تعریف آن به صورت زیر است.

$$skew = \frac{\mu_3}{\sigma^3} \quad \rightarrow \quad \mu_3 = \sum_n (X_{lead}[n] - mean(X_{lead}))^3$$

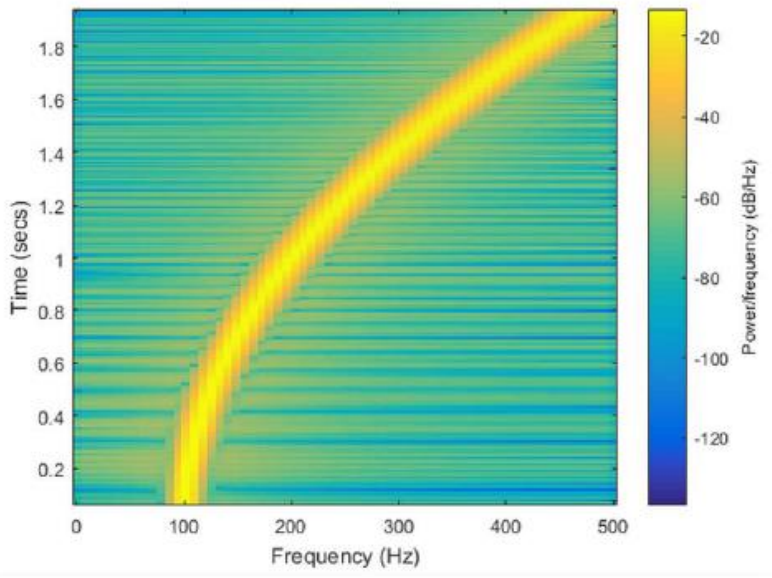


mean frequency & median frequency.9

با استفاده از توابع medfreq و meanfreq در متلب محاسبه می شوند.فرکانس میانگین،میانگین وزن دار فرکانس های یک سیگنال است که وزن هر فرکانس،چگالی دامنه آن در آن فرکانس می باشد.فرکانس میانه نیز با توجه به تعریف میانه مشهود است.

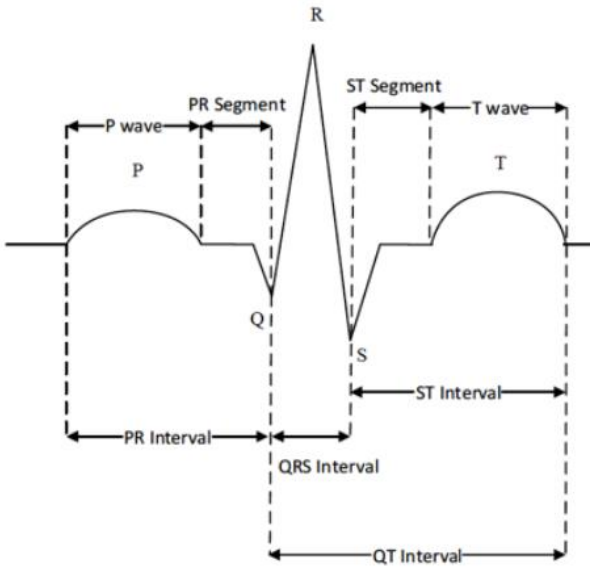
short time fourier transform(STFT).10

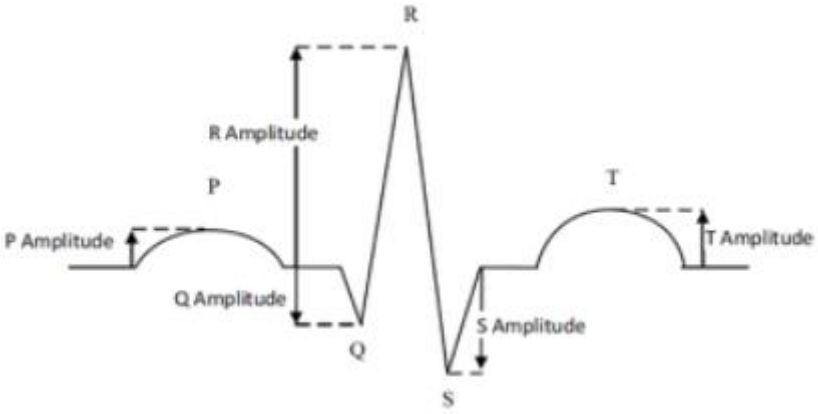
تبدیل فوریه زمان کوتاه،ابزاری است برای برقراری ارتباط بین حوزه زمان و فرکانس که در آن،محتوای فرکانسی را در طول زمان بررسی می کنیم.در واقع،این تبدیل،سیگنال را به پنجره های زمانی کوچک تقسیم می کند و در هر کدام از این پنجره های زمانی،به محاسبه تبدیل فوریه می پردازد.بنابراین،می توان تغییرات انرژی در فرکانس های مختلف را بررسی کرد.شکل زیر،نمونه ای از این تبدیل است که به شکل نمودار 2 بعدی زمانی_ فرکانسی رسم شده است.



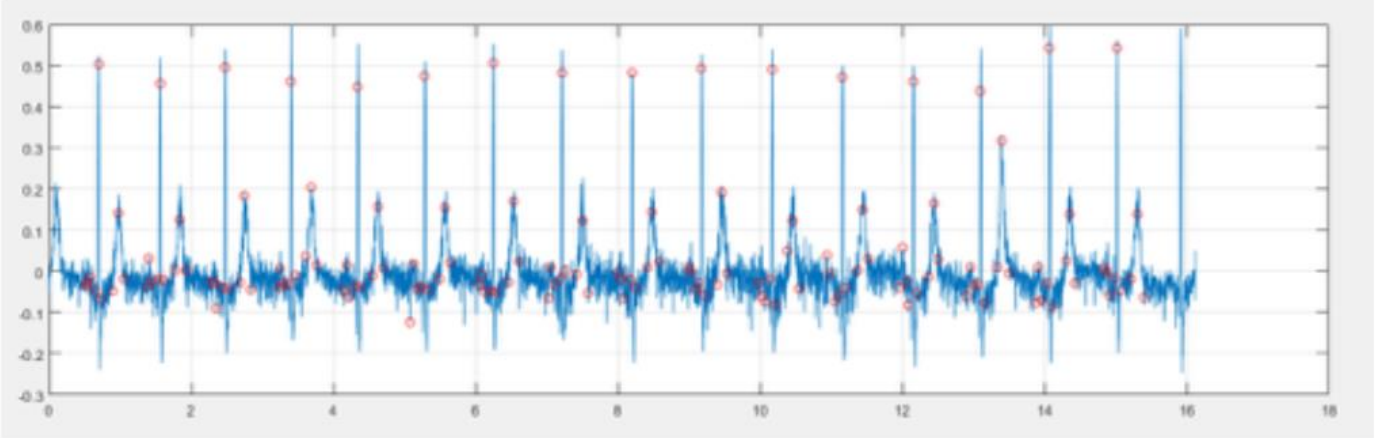
QRS datas.11

از دیگر ویژگی هایی که به طور معمول در مطالعه سیگنال های ECG استفاده می شود،نقاط بحرانی و بازه های زمانی این سیگنال ها هستند.





به کمک wfdb toolbox و تابع findpeaks نقاط بحرانی را به دست می آوریم.16 ویژگی موثری که شامل میانگین و واریانس دامنه ها،میانگین و واریانس مقادیر بازه ها و میانگین و واریانس طول بازه ها می باشد را حساب کردیم.در ادامه مشاهده خواهید کرد که این ویژگی ها،تاثیر به سزایی در جداسازی کلاس های خروجی خواهند داشت.شکل زیر نمونه ای از خروجی تابع برای lead 1 یکی از بیمار ها است.



حال به کمک ویژگی هایی که به دست آورده ایم،ماتریس feature را می سازیم.به دلیل بزرگ بودن سایز ماتریس، باید به دنبال راهی باشیم تا زمان training را کم کنیم.برای این کار،از چند روش استفاده خواهیم کرد که ادامه کار توضیح داده خواهد شد.

بخش چهارم:انتخاب ویژگی های موثر و پیش پردازش ماتریس ویژگی

(4.1

از کارهای مرسوم در مقاله های learning،بحث feature selection و pre-process ماتریس ویژگی است. در این پروژه از الگوریتم ReliefF برای انتخاب ویژگی استفاده کردیم.بررسی کامل الگوریتم در مقاله ای که در انتهای گزارش آورده ایم،انجام شده است.در این جا شرح مختصری از آن را انجام می دهیم.

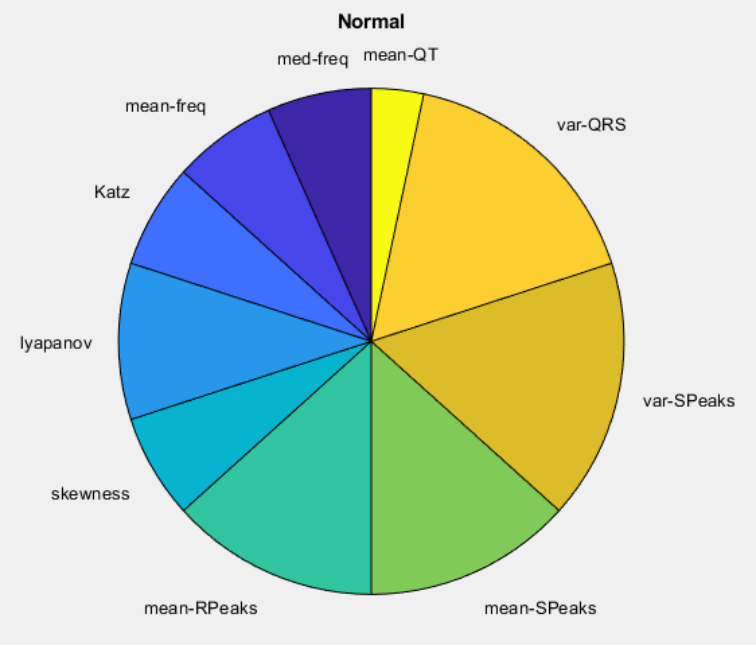
در این روش برای هر ویژگی در ابتدا یک وزن 0 در نظر گرفته می شود.در هر بار تکرار،یک نمونه به طور تصادفی انتخاب می شود و به کمک الگوریتم KNN،بردار های ویژگی از تک تک کلاس ها که نزدیک به نمونه انتخاب شده هستند را پیدا می کنیم.اگر هر دو از یک کلاس بودند،امتیاز آن کاهش و اگر از کلاس متفاوت بودند،امتیاز آن افزایش می یابد.امتیاز هم به صورت فاصله اقلیدسی تعریف می شود.

$$w_i = w_i - (x_i - nearheat_i)^2 + (x_i - nearmiss_i)^2$$

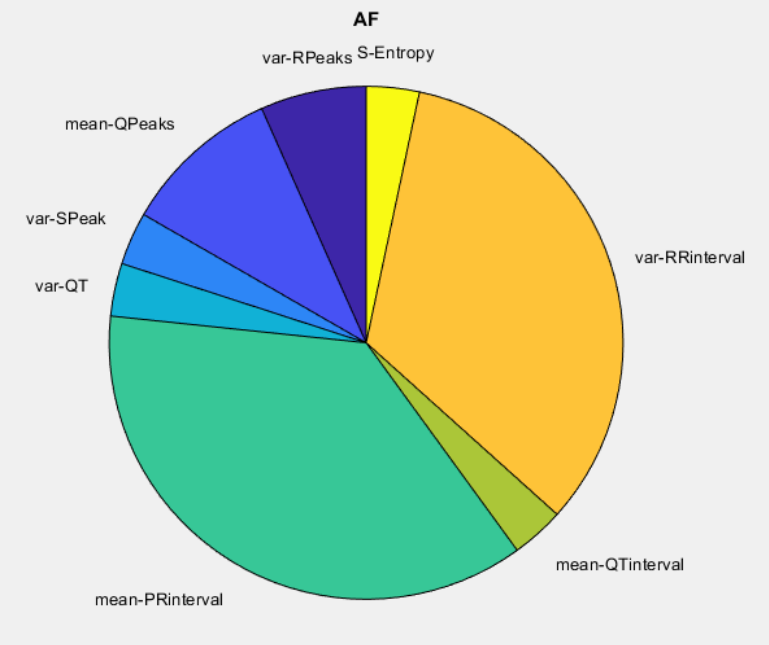
به کمک کران چبی شف ثابت می شود که می توان یک upperbound خوب برای خطا پیدا کرد.
برای پردازش ماتریس ویژگی ها،با مطالعه مقالات مختلف،تصمیم گرفتیم که تمام feature ها را به بازه 1 تا 1- نگاشت کنیم.توضیح بیشتر این مورد در بخش چالش ها داده خواهد شد.

(4.2

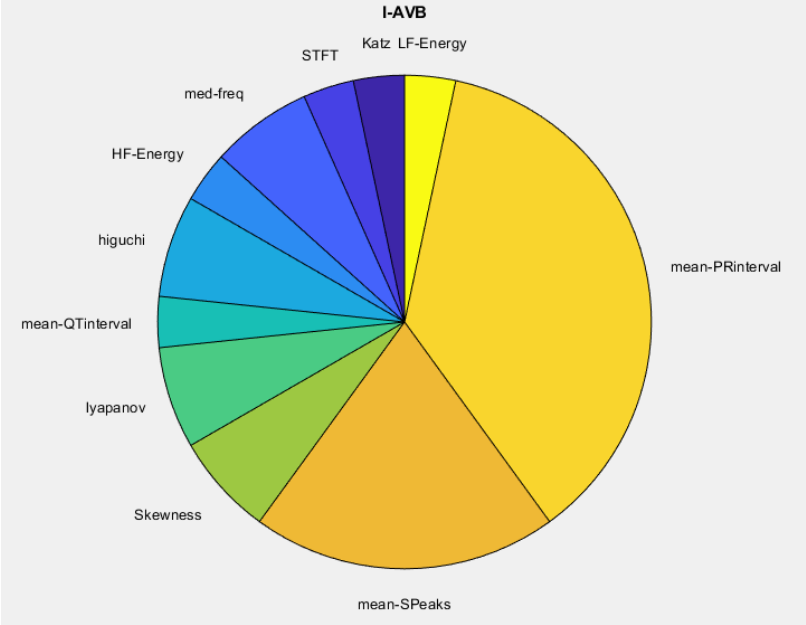
با توجه به این که کلاس های خروجی ما دارای 9 لیبل مختلف می باشد از one-hot encoding استفاده می کنیم.یعنی برای هر بیماری به صورت جداگانه لیبل ها را جدا می کنیم.این کار زمان آموزش را کاهش داده و و مقایسه خروجی ها را ساده می کند.در این بخش،می خواهیم بهترین ویژگی هایی که برای هر بیماری استخراج کردیم را بیان کنیم.چارت های زیر نشان می دهند که برای هر بیماری،کدام ویژگی ها موثرتر بوده اند.در این چارت ها،پراکندگی 30 ویژگی اول موثر نشان داده شده است.



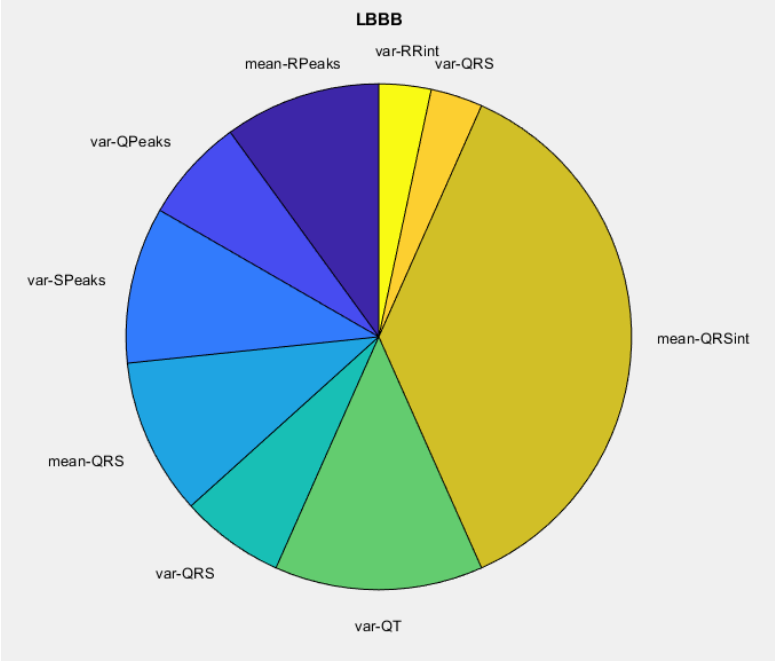
برای حالت نرمال،واریانس پیک S،واریانس QRS،میانگین پیک S ویژگی های مهم و تاثیر گذار برای جدا کردن بیمار ها بودند.



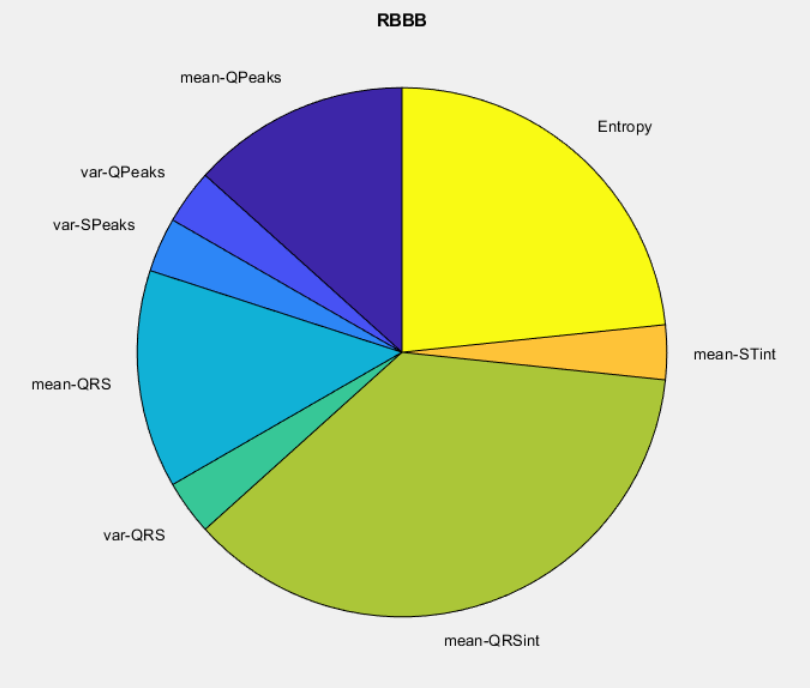
برای AF،مشاهده شد که میانگین بازه RP،نقش بسیار تاثیرگذاری در انتخاب این بیماری دارد به طوری که ویژگی های دوم تا 10 موثر،همگی مربوط به این مورد بودند.همین طور Entropy،ویژگی مهمی بود.



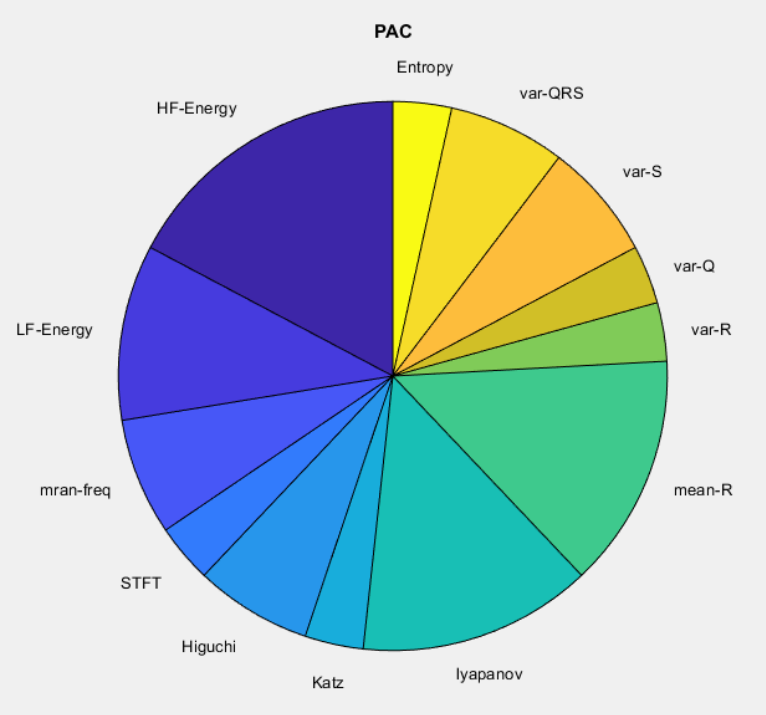
برای I-AVB نیز میانگین مقادیر بازه RP،نقش مهمی در تشخیص این بیماری داشت.



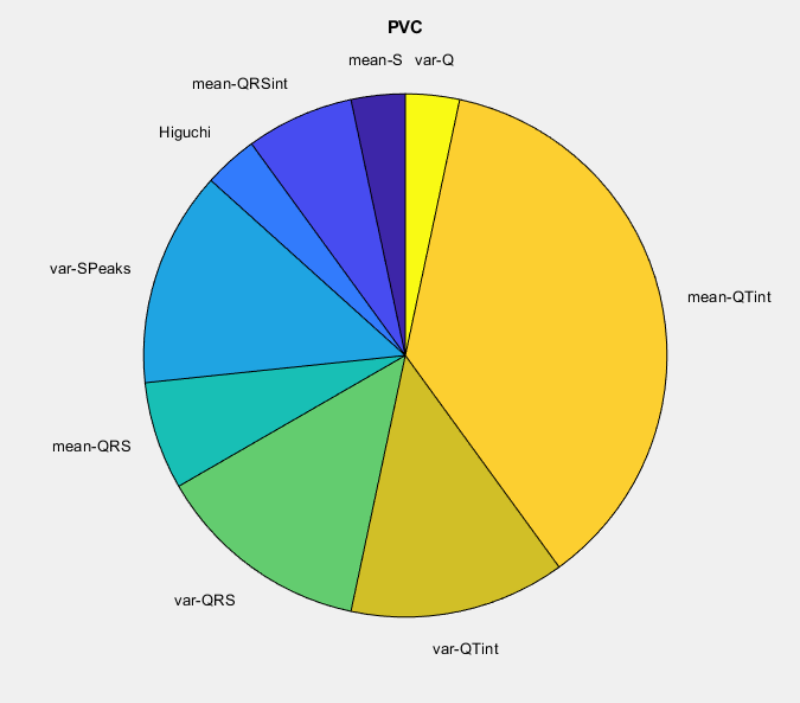
برای LBBB،واریانس بازه QT،واریانس پیک S،واریانس QRS ویژگی های مهم برای تشخیص بودند.



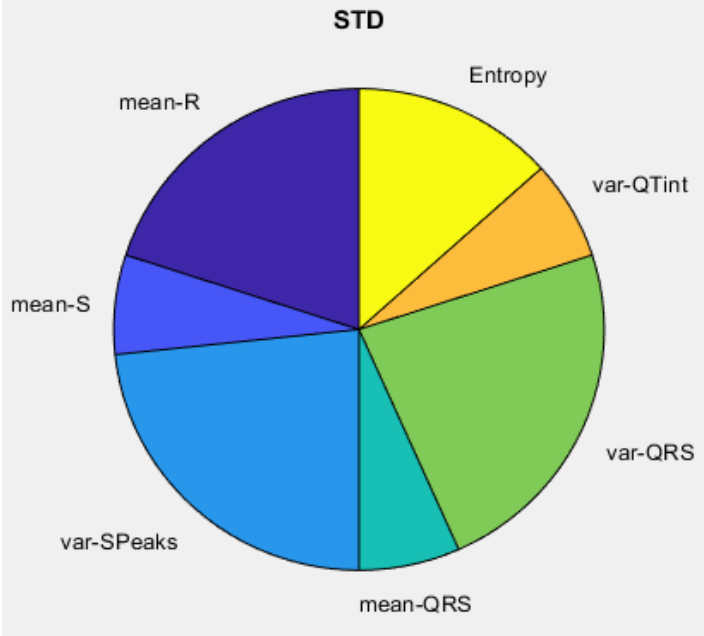
برای RBBB،میانگین مقادیر بازه QTو به خصوص میانگین بازه های زمانی QT بسیار موثر بودند.



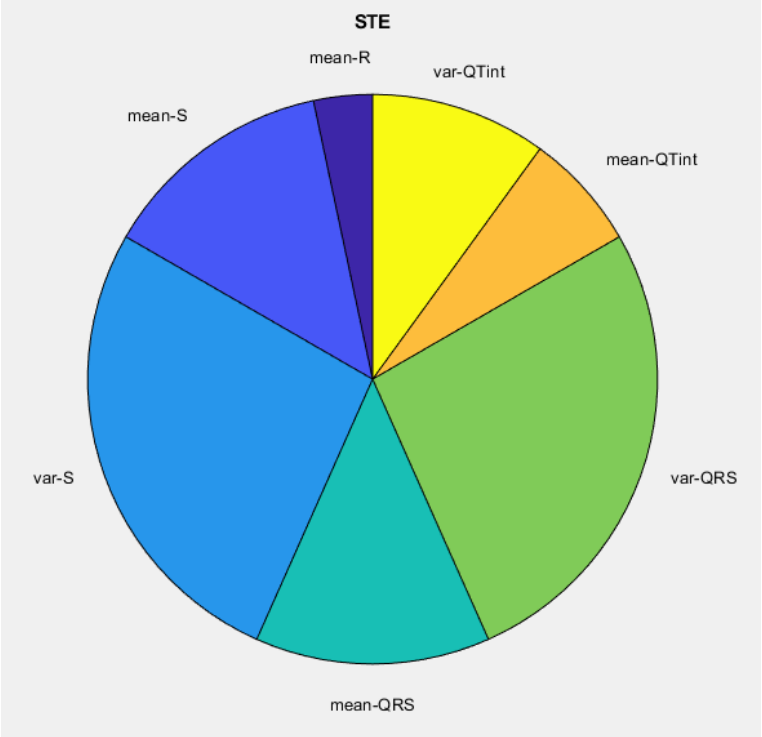
برای PAC،ویژگی های Petrosian،انرژی باندهای فرکانسی مختلف و Katz از ویژگی های مهم بودند.



برای PVC،واریانس بازه QT،واریانس پیک های Q و واریانس پیک S از ویژگی های مهم بودند.



برای STD،واریانس پیک های S و واریانس QRS و Entropy تاثیر گذار بودند.

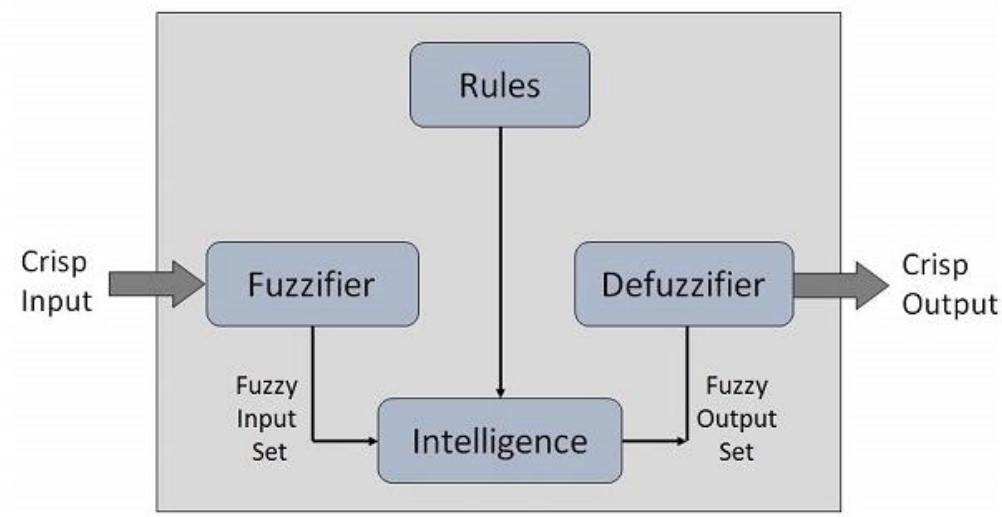


برای STE،واریانس QRS ، واریانس پیک های S و میانگین بازه QT مهم بودند.

بخش پنجم:آموزش و اعتبارسنجی داده ها

(5.1

در این پروژه از با توجه به این که از one-hot encoding استفاده کردیم،بنابراین 9 سیستم فازی مجزا را آموزش خواهیم داد.در این پروژه از دو روش مجزا برای رسیدن به این هدف استفاده شده است.روش اول، استفاده از سیستم فازی خالص است.ما از سیستم فازی سوگنو برای آموزش داده های خود استفاده می کنیم. جزئیات این روش و ساختار آن به تفصیل در درس مورد بحث قرار گرفته است.لذا به طور خیلی مختصر به آن اشاره می کنیم.یک سیستم فازی به طور خیلی ساده،ساختاری شبیه به شکل زیر دارد.



در قسمت fuzzifier،تابع عضویت ورودی تعریف می شود و می توانیم به کمک آن برای هر ورودی،یک مقدار عضویت برگزینیم.سپس در قسمت inteligence،به کمک قوانین تعریف شده و عملگرهای از پیش تعیین شده، مقدار عضویت خروجی تعیین می شود.سپس به کمک defuzzifier،به خروجی مطلوب می رسیم.

کنترل کننده سوگنو از قوانینی مشابه زیر استفاده می کند.

$$if\ x\ is\ A\ and\ y\ is\ B\ then\ z = f(x,y)$$

ایده اصلی این است که تابع متناظر،یک تابع کنترلی محلی خوب برای منطقه فازی باشد.در متلب،برای تابع f دو حالت خطی و ثابت در نظر گرفته شده است که ما از حالت خطی آن استفاده می کنیم.

شاید به جرئت به توان گفت که یکی از مهم ترین قسمت های پروژه،انتخاب تعداد تابع عضویت های ورودی سیستم فازی است که تاثیر به سزایی در خطای آموزش و زمان یادگیری ما دارد.در این مورد به تفصیل در قسمت چالش ها صحبت خواهد شد.

پیاده سازی این قسمت به کمک تابع genfis در متلب انجام شد.این تابع،تعریف قوانین و تابع عضویت ورودی را با 3 الگوریتم متفاوت انجام می دهد.پس از مطالعه بسیار و سعی و خطا،از روش FCMclustering استفاده کردیم.یکی از تفاوت های جالبی که این روش با بقیه روش های clustering دارد،این است که یک نقطه ممکن است با وزن های مختلف به بیش از یک دسته تعلق داشته باشد.این الگوریتم سعی در مینیمم کردن تابع هدف زیر دارد.

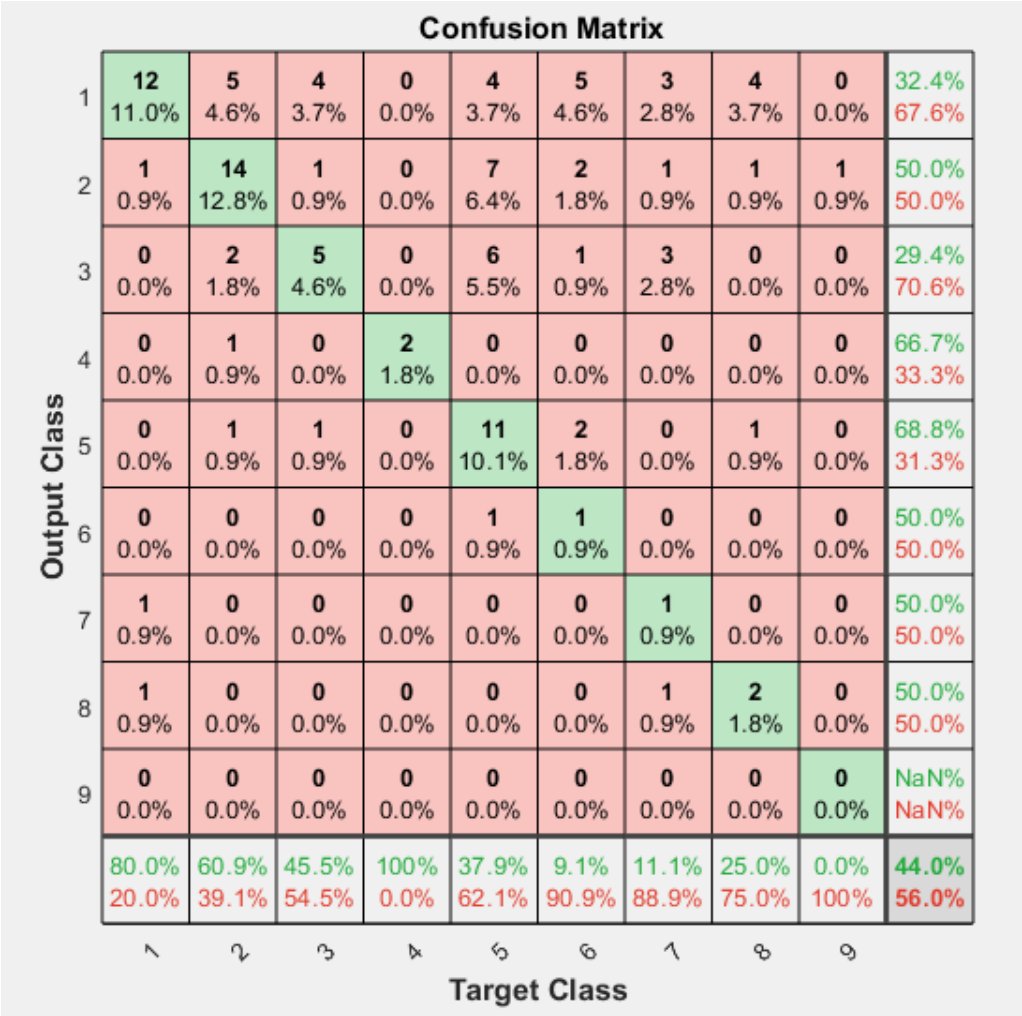
$$argmin(c)\ \sum_{i=1}^{n}\sum_{j=1}^{c} {w_{i,j}}^{m}\left||x_i - c_j|\right|^2 \quad where \quad w_{i,j} = \frac{1}{\sum_{k=1}^{c}\left(\frac{||x_i - c_j||}{||x_i - c_k||}\right)^{\frac{2}{m-1}}}$$

که c_j ها دسته های ما و x_i نقاط ما هستند.پارامتر m،نقش فازی کننده و تعیین مقدار عضویت در یک دسته را برای ما بازی می کند.ما بعد از 1000 تکرار،این الگوریتم را متوقف می کنیم.

در ادامه به کمک تابع tunefis،سعی در بهینه کردن پارامتر های سیستم فازی تولید شده می کنیم.ما برای این کار از الگوریتم بهینه سازی و جستجو genetic استفاده می کنیم.دیگر پارامتر که برای کاهش زمان اجرا استفاده می کنیم،استفاده از parallel computing است.در این مورد به تفصیل در بخش چالش ها توضیح داده خواهد شد.در متلب سری 2020 به بعد،تابع cross validation Kfold به ویژگی های این تابع اضافه شده است.چون یکی از اهداف این پروژه استفاده از این ویژگی بود،از 5fold برای جلوگیری از overfitting استفاده کردیم.برای آموزش از 9 ویژگی اولی که به کمک الگوریتم relief پیدا کردیم،استفاده می کنیم.

زمان آموزش 9 سیستم فازی ما چیزی نزدیک به 5 ساعت طول کشید.

پس از اتمام و ذخیره سازی سیستم های فازی،به کمک تابع evalfis،خروجی داده های تست را محاسبه می کنیم. چون مقادیر به دست آمده،اعدادی بین 0 و 1 هستند،باید به دنبال راهی باشیم تا آن ها را به 0 و 1 تبدیل کنیم. در این جا،دو راه حل پیش روی خود داریم.یا باید مقادیر به دست آمده را گرد کنیم و یا باید با انتخاب کردن یک Threshold،کلاس خروجی را تعیین کنیم.پس از تحقیق و تفحص بسیار،تصمیم گرفتیم که راه دوم را انتخاب کنیم.از عدد 0.2 برای به دست آوردن کلاس خروجی استفاده کردیم.مقادیر کمتر از 0.2،مقدار 0 و مقادیر بیشتر از 0.2،مقدار تخصیص دادیم.حال به کمک confusion matrix،قصد در تحلیل نتایج به دست آمده و مقایسه آن ها داریم.



حال به بررسی جزئیات این ماتریس می پردازیم.سطرها نمایانگر خروجی های سیستم و ستون ها نماینده کلاس های خروجی از پیش دانسته شده هستند.قطر اصلی نشان دهنده مشاهده هایی هست که به درستی طبقه بندی شده اند.خانه های غیرقطر اصلی هم مشاهده هایی هستند که به درستی طبقه بندی نشده اند.آخرین ستون سمت راست در هر کلاس نشان دهنده دو متریک اندازه گیری است.ابتدا 4 معیاری که برای سنجش اعتبار داده در بحث لرنینگ زیاد استفاهد می شود را معرفی می کنیم.

	Has disease	Does not have disease
Identified as having disease	True positive (TP)	False positive (FP)
Identified as not having disease	False negative (FN)	True negative (TN)

TP،معرف کسانی هستند که بیماری را داشته و درست طبق بندی شده اند.FP،کسانی که بیماری را نداشته ولی جز بیماران طبقه بندی شده اند.TN،کسانی که بیماری را ندارند و جز غیر بیماران طبقه بندی شده اند.
FN،کسانی که بیماری را دارند و جز غیر بیماران طبقه بندی شده اند.

حال متریک هایی که برای ستون سمت راست هستند،به صورت زیر تعریف می شوند.

$$Precision = \frac{TP}{TP + FP} \qquad FDR = \frac{FP}{FP + TP}$$

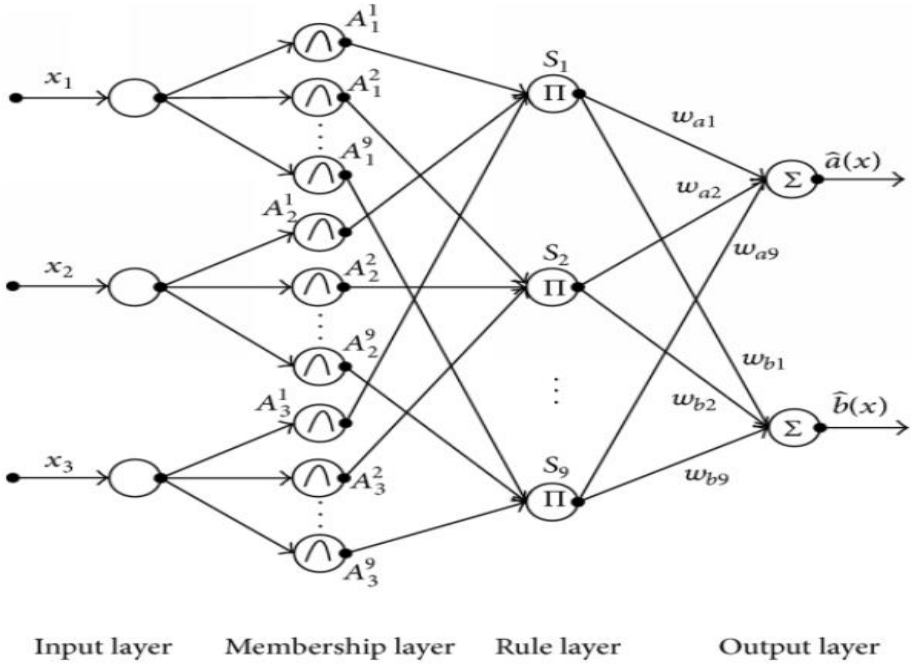
در کل نشان می دهند که نسبت به کل کسانی که بیمار تشخیص داده شده اند،چه تعداد واقعا بیمار بوده و چه اصلا بیمار نبوده اند.

آخرین سطر در این ماتریس،دو متریک دیگر را برای اعتبارسنجی داده ها استفاده می کند.

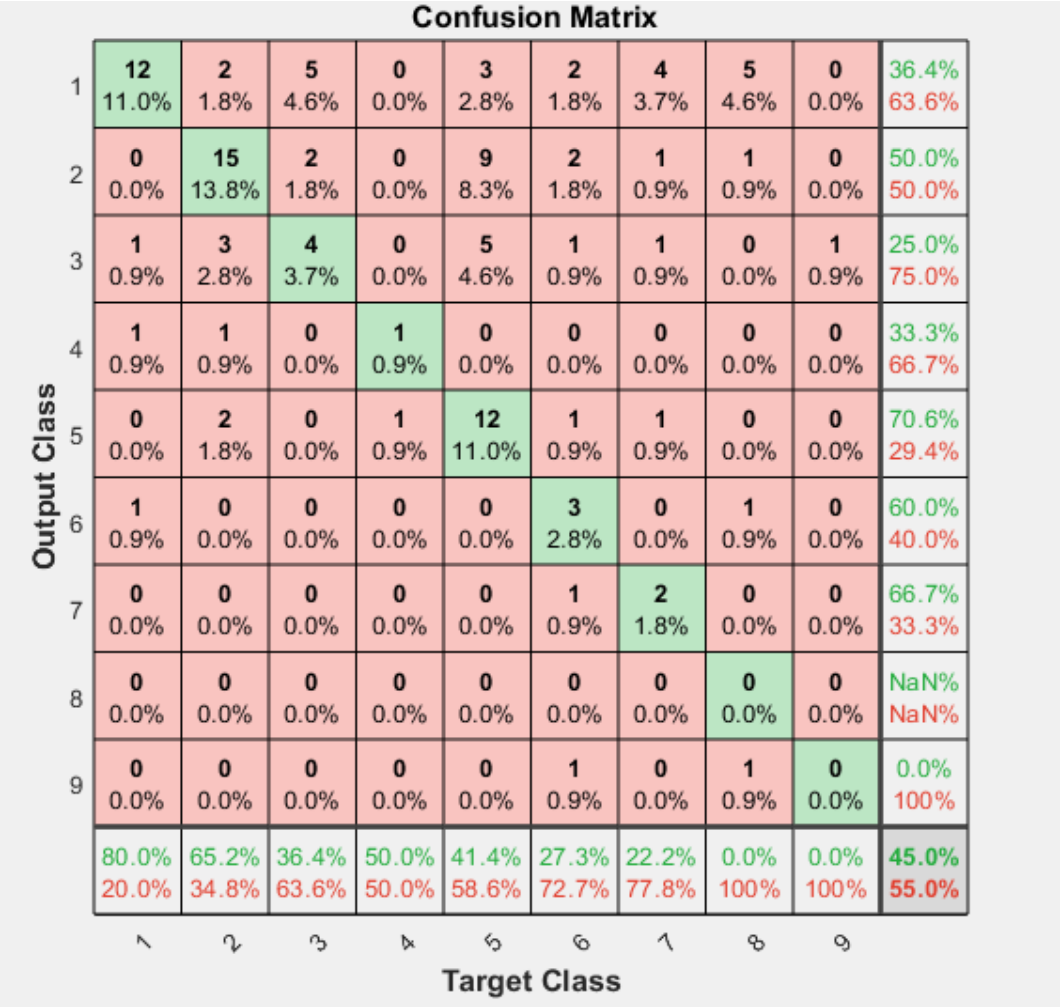
$$recall = \frac{TP}{TP + FN} \qquad FNR = \frac{FN}{FN + TP}$$

این دو مورد درصد صحت را بین لیبل هایی که درست تشخیص داده شده اند،انجام می دهد.
برای چند بیماری اول،می بینیم که نعداد زیادی به اشتباه بیمار تشخیص داده شده اند.دلیل آن تا حدودی به انتخاب Threshold برای طبقه بندی بر می گردد.از دزصد های دو بیماری آخر،می توان فهمید تعداد کسانی که این بیماری را دارند بسیار کم هستند.درصد صحت کلی ما در این قسمت 44% بود.

روش دیگری که در این پروژه برای رسیدن به هدف استفاده کردیم،استفاده از سیستم Neuro-fuzzy است.
به طور کلی در این روش،ما سعی در پیدا کردن پارامترهای سیستم فازی به کمک تکنیک هایی که در شبکه عصبی استفاده می شود داریم.در این سیستم برخلاف سیستم فازی معمولی،امکان تعریف کردن قوانین قبل،هنگام یا بعد از فرآیند آموزش وجود دارد.شکل زیر یک سیستم عصبی ـ فازی ساده را نشان می دهد.



این ساختار در متلب به کمک تابع Anfis پیاده سازی می شود.همان طور که در تعریف بیان کردیم،ما پارامترهای سیستم فازی را به کمک شبکه عصبی پیدا می کنیم.برای این قسمت،تعداد epoch ها را برابر با 10 در نظر می گیریم.آموزش داده ها از این روش چیزی نزدیک به 8 ساعت به طول انجامید.confusion matrix به صورت زیر است.



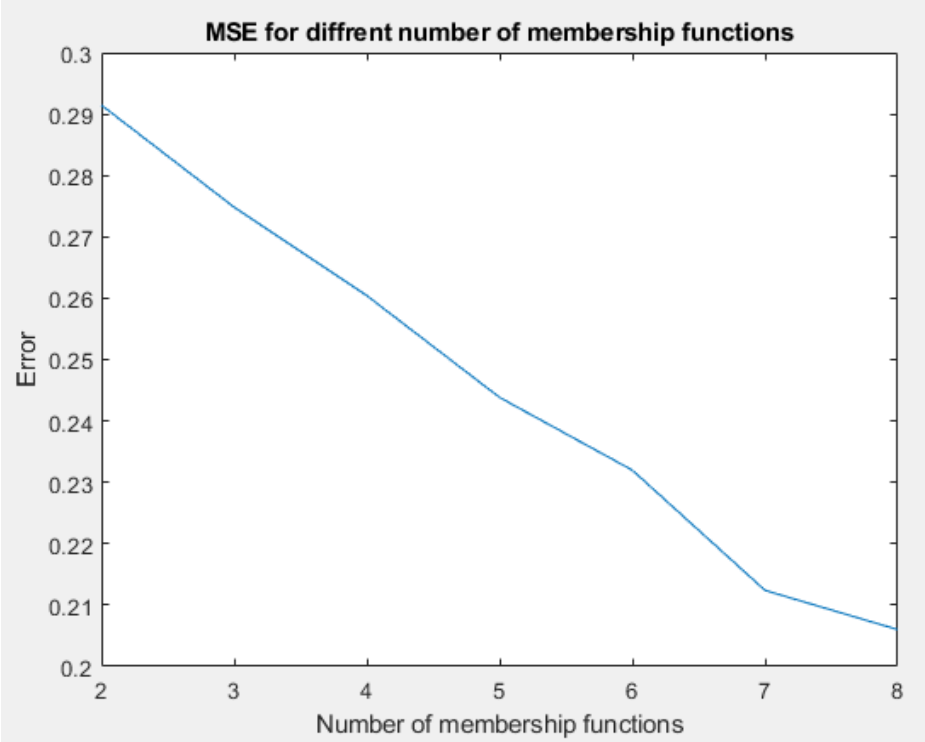
تقریبا می توان دید که خروجی های این قسمت نیز شباهت زیادی به قسمت قبل دارند به طوری که مثلا برای بیماری های اول،تعداد زیادی به اشتباه بیمار تشخیص داده شده اند.درصد صحت کلی در این قسمت 45% بود.تفاوت بزرگ این دو سیستم،زمان اجرای آن ها بود به طوری که به کمک genfis،در مدت زمان بسیار کمتری به خروجی مشابه رسیدیم.

بخش ششم:چالش ها !!!!

در طول این پروژه به چالش های متعددی برخوردیم که سعی کردیم با استفاده از روش های متفاوت با آن ها برخورد کنیم.

1)مهم ترین چالشی که با آن برخورد کردیم،مسئله زمان اجرا و کمبود مموری بود.اولین راه حلی که برای مواجه با این مشکل در نظر گرفتیم،پردازش روی GPU بود.GPU به کمک پردازش موازی و استفاده از کارت گرافیک زمان محاسبه را به طور قابل توجهی کاهش می دهد.یکی دیگر از راه حل های کارامد،استفاده از ویژگی های موازی سازی خود متلب است.از جمله این موارد می توان به parallel pool و ماتریس های distributed اشاره کرد.به طور مثال به کمک parallel pool می توان یک حلقه for را به صورت موازی پردازش کرد تا زمان آن کاهش یابد.یکی دیگر از کارهایی که به طور کلی در کاهش زمان اجرا موثر است،استفاده هرچه بیشتر از feature های خود متلب برای پیاده سازی الگوریتم ها است.مثلاً استفاده از tunefis برای بهینه کردن پارامترهای سیستم این امکان را به ما می دهد که با متود های گوناگون،پارامتر های سیستم را بهینه کنیم.

2)یکی دیگر از چالش هایی که پیش روی ما بود،انتخاب تعداد تابع عضویت ورودی سیستم فازی بود.این مورد تاثیر به سزایی در کاهش خطا و یادگیری ما دارد.نمودار زیر برای یک تعداد epoch ثابت،تغیرات خطای کمینه مربعات را برای تعداد تابع عضویت ورودی متفاوت به ما نشان می دهد.از تابع anfis برای شبیه سازی این قسمت استفاده کردیم.



همان طور که می توان دید،با زیاد شدن تعداد تابع عضویت ورودی،خطای ما به صورت تقریبا خطی کاهش پیدا می کند.

ولی خوب همان طور که می توان حدس زد،افزایش تعداد تابع عضویت ورودی،زمان اجرای برنامه را به صورت قابل ملاحظه ای افزایش می دهد.نمودار زیر،زمان های اجرا به ازای تعداد مختلف تابع عضویت ورودی را به ما نشان می دهد.در این قسمت از تابع anfis استفاده شده است.

همان طور که می توان دید،زمان اجرا به صورت نمایی افزایش می یابد،به طوری که برای 8 عدد تابع عضویت ورودی،زمان اجرا به ازای یک epoch،نزدیک به 2 ساعت طول می کشد!!!

پس یک بده بستان بین زمان اجرا و رسیدن به خطای کمتر وجود دارد.ما در این پروژه از 7 تابع عضویت ورودی برای آموزش شبکه استفاده کردیم.

3)یکی دیگر از چالش هایی که در پروژه داشتیم،این بود که در بعضی موارد،خروجی مطلوبی از سیستم های خود نمی گرفتیم.یعنی خروجی به دست آمده،خارج از بازه مورد نظر بود.علت اصلی آن این است که ورودی داده های تست ممکن است خارج از بازه ورودی های داده آموزش باشند.به همین خاطر قبل از آموزش،تمامی مقادیر ماتریس ویژگی را بین 1 و 1- نگاشت می کنیم.این کار تا حدی مشکل پیش آمده را حل می کند.یکی دیگر از دلایل این موضوع،قدیمی بودن سیستم پیاده شده برای بخش فازی در متلب است.یعنی کلیت این سیستم ها در 20 سال اخیر هرچند با وجود اضافه شدن feature های جدید،تغییر نکرده است.

بخش هفتم:خلاصه و جمع بندی نهایی

فرآیند کلی این پروژه به این صورت بود که ابتدا پیش پردازش دیتاست بیماران انجام دادیم.سپس با مطالعه مقاله های گوناگون،به جستجوی ویژگی های موثر برای بخش استخراج ویژگی ها پرداختیم.برای کاهش زمان اجرا،روش های گوناگون از جمله feature selection و one-hot encoding را به کار بردیم.

سپس از دو راه مجزا به آموزش 9 سیستم فازی برای بیماری ها به صورت جداگانه مشغول شدیم که خروجی های ما به دلیل چالش های به وجود آمده چندان رضایت بخش نبودند.در نهایت هم چالش های به وجود آمده و راه حل های اتخاذ شده برای رسیدن به خروجی مطلوب تر را بیان کردیم.شاید نرسیدن به خروجی مطلوب را بتوان به کمک جمله زیر از دکتر Rudolf Kruse بهتر دریافت کرد.

On the contrary, a fuzzy system demands linguistic rules instead of learning examples as prior knowledge. Furthermore the input and output variables have to be described linguistically. If the knowledge is incomplete, wrong or contradictory, then the fuzzy system must be tuned. Since there is not any formal approach for it, the tuning is performed in a heuristic way. This is usually very time consuming and error-prone.

بخش هشتم:مراجع

1. Signal Pattern Recognition Based on Fractal Features and Machine Learning

Chang-Ting Shi

2. Higuchi Fractal Analysis of Heart Rate Variability is Sensitive during Recovery from Exercise in Physically Active Men

Rayana L. Gomes, Luiz Carlos M. Vanderlei

3. Geometry of chaos,chapter 6,Lyapanouv Exponent Stanislav V. Klimenkov