

P300_Speller

HW_03

Alireza Shirzad_Alireza Kazemi_Mahdi Sadighi | Computational NeuroScience | May 2018

Part 1

QUESTION 1

یکی از مهم ترین دستاوردهای بشر در حوزه Neuro_Science و بالخصوص Neuro_Engineering طراحی سیستم های BCI برای تعامل مستقیم ادراک انسان با کامپیوتر بوده . با مرور زندگی دانشمند فقید ، استیون هاوکینگ ، که از بیماری ALS رنج می برد در می یابیم که اگر چنین سیستم هایی تعبیه نشده بودند ، خیلی زود باید این مرد بزرگ را فراموش می کردیم . یکی از مهم ترین بخش های حوزه BCI ، ساختن Speller هاست ، زیرا انتقال مفهوم و اطلاعات از مغز به کامپیوتر از مهم ترین چالش های بشری است . هدف کلی این پژوهش طراحی سیستم BCI جهت پردازش Real_Time سیگنال های EEG و Spell کردن کلمات با دقت بالا می باشد . پیش از این سیستم های دیگری نیز در این حوزه به کار گفته شدند مانند Motor_Imagery اما به نسبت دقت بسیار پایین تری داشتند . به عنوان مثال در پروژه Motor_Imagery تنها 19% از افراد به دقت 80%-100% رسیدند اما در این پژوهش 89% از افرادی که از مدل RC استفاده کردند توانستند به این دقت برسند . تفاوت دیگر در نحوه ی پردازش اطلاعات است . سیستم های BCI معمولاً به چندین نحو کار میکنند که عبارتند از :

1 _ Slow Cortical Potentials (SCPs) : که تغییرات بلند مدت سیگنال های EEG را مانیتور می کند و بر اساس آن train می شود .

2 _ oscillations in alpha and beta range : که تغییرات در توان باند های آلفا و بتا را به عنوان feature های مهم مورد مطالعه قرار می دهد .

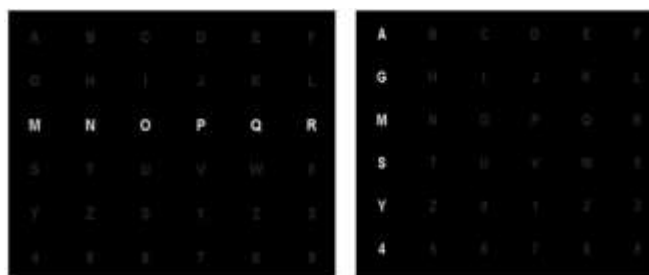
3 _ steady-state visual evoked potentials (SSVEP) : که بر اساس نوسانات هماهنگ سیگنال های مغزی با نوسانات تحریک کار میکند .

4 _ P300 event-related potential : و در آخر تغییر دامنه ی سیگنال های EEG که معمولاً در peak سوم و بین زمان های حدود 300 الی 400 میلی ثانیه اتفاق می افتد که به آن ها P300 می گویند . این تغییر ناگهانی در دامنه به خاطر پردازش های مغز در لایه های بالاتر ادراکی صورت می گیرد و فاکتور مهمی در جدا کردن تحریک deviant از سایر تحریک هاست .

در این پژوهش بر خلاف چندین پژوهش قبلی از P300 event-related potential استفاده شده زیرا هدف ما در این مقاله جدا کردن target ها از nontarget هاست که مستقیماً روی دامنه ی P300 تاثیر می گذارد .

QUESTION 2 :

در این پژوهش برای نمایش حروف از دو پروتکل کلی RC و SC استفاده شده . در پروتکل RC (Row_Column) سطر ها و ستون ها به صورت random روشن می شوند اما در پروتکل SC (Single_Cell) هر خانه به صورت random می شود .





شکل 1:

مانیتور نشان داده شده به سابجکت ها

آزمایش بدین صورت است که فرد جلوی مانیتور می نشیند و 8 کانال EEG از مغز او گرفته می شود. سیگنال ها عموماً از بخش Parietal گرفته شده که به اعتقاد من برای کاهش artifact های ناشی از حرکت چشم ها می باشد. البته از شرکت کننده ها در تست خواسته شده که ریلکس باشند و حداقل امکان تکان نخورند. دیتای EEG توسط g.USBamp که یک amplifier پر استفاده در حوزه های Bio است با فرکانس نمونه برداری 256 Hz در اختیار ما قرار گرفته. جزئیات عملیاتی نصب و راه اندازی را توضیح نمی دهیم. عده ای از داوطلبان در پروتکل RC، عده ای در SC و عده ای در هر دو شرکت کردند. سابجکت ها باید به حرفی که قصد دارند spell کنند خیره شوند و با دقت نگاه کنند. برای افزایش تمرکز به آن ها گفته شده تعداد روشن شدن target را بشمارند. در ابتدا برای Calibration و training سیستم، به آن ها گفته شده کلمه ی "WATER" را Spell کنند. در عرض 5 دقیقه مدل LDA، train می شود و سابجکت برای ارزیابی نهایی و استفاده از دستگاه کلمه ی "LUCAS" را Spell می کند. پس از 15 بار روشن شدن هر خانه، یا سطر و ستون (یعنی sweep شدن کل صفحه برای 15 بار)، مدل LDA روی آن اعمال می شود. پس از مشخص شدن کاراکتر، تصمیم ماشین روی صفحه، نمایش داده می شود. جزئیات دقیق این آزمایش را به صورت دقیق در سئوالات بعدی توضیح می دهیم.

QUESTION 3 :

فرق مهم یک target و nontarget در دامنه ی P300 آن است. لذا بهترین راه برای classification، استفاده از Feature های حوزه زمان است. بدین منظور پس از نمایش هر سطر/ستون یا حرف سیگنال EEG به مدت 800 میلی ثانیه توسط یک Buffer، Latch می شود. این بازه ی 800 میلی ثانیه ای با trial های دیگر نیز Overlap دارد اما عموماً trial ها شامل target دیگری نمی شود لذا می توانیم رفتار P300 را به خوبی مشاهده کنیم. همزمان شماره ی چیزی که نمایش داده شده نیز ذخیره می شود. ابتدا با فیلتر میان گذر، باند 0.5 Hz تا 30 Hz را فیلتر می کنیم تا نویز های فرکانس بالا و بسیار پایین حذف شوند و در حوزه فرکانسی مغز کار کنیم. سپس داده ها را با ضریب 4 Down_Sample می کنیم. از دلایل DownSampling می توان به دو مورد اشاره کرد. یک اینکه اطلاعات ما در باند محدودی است. دوماً که ابعاد دیتای ما کمی کاهش یابد. حال با داشتن 8 کانال EEG و feature های زمانی و label های آن ها می توانیم ماتریس داده هایمان را بسازیم. در آخر مدل LDA را روی این ماتریس دیتا پیاده سازی می کنیم تا معادله خط جدا کننده ی دو دسته مشخص شود.

QUESTION 4 :

در این سؤال مختصراً نتایج را بیان می کنیم اما در بخش های بعد به تفصیل آن ها را آنالیز می کنیم.

1_ برخلاف باور اشتباه نویسندگان مقاله در ابتدا، دقت پروتکل RC از SC بیشتر است.

2_ جنسیت ، مصرف قهوه ، نوشیدنی های الکلی ، مواد مخدر و سیگار ، سطح سواد و ساعت کاری تاثیر خاصی روی عملکرد افراد ندارد .

3_ افرادی که ساعت خواب کمتری داشتند به مراتب بهتر از افراد با ساعت خواب نرمال عمل کردند .

4_ دقت های به دست آمده در دو پروتکل به صورت زیر می باشد :

Classification accuracy in %	Row-column speller: percentage of sessions (N= 81)	Single character speller: percentage in sessions (N= 38)
100	72.8	55.3
80-100	88.9	76.3
60-79	6.2	10.6
40-59	3.7	7.9
20-39	0.0	2.6
0-19	1.2	2.6
Mean accuracy of all subjects	91.0	82.0
Spelling time [s]	28.8	54
Mean accuracy of subjects who participated in RC and SC (N= 19)	85.3	77.9

جدول 1 :

دقت های حاصل شده در پارادایم های RC و SC

Part 2 :

QUESTION 1 :

فایل هر سابجکت شامل یک struct ، حاوی دیتاهای دو آزمایش test و train است . هر کدام از آن ها ماتریسی حاوی اطلاعات زیر است :

ردیف اول : زمان آزمایش با دقت 1/256 s .

ردیف دوم تا نهم : سیگنال های 8 کانال EEG

ردیف دهم : شماره ی سطر/ستون در پروتکل RC و شماره خانه در پروتکل SC

ردیف یازدهم : در صورت target بودن یک ، و در غیر این صورت صفر است .

RC On_Time	RC Off_Time	SC On_Time	SC Off_Time
15.625 milisecond	15.625 milisecond	15.625 milisecond	62.5 milisecond

جدول 2 :

Timing پارادایم های SC و RC طبق Dataset

QUESTION 2 :

برای فهمیدن اینکه پروتکل استفاده شده ، SC است یا RC ماکزیمم درایه های سطر 10 را پیدا میکنیم . اگر 26 بود یعنی SC است و اگر 12 بود یعنی RC است .

Subject Num	Protocol
1	SC
2	SC
3	RC
4	RC
5	RC
6	RC
7	RC
8	RC
9	RC
10	RC

جدول 3 :

مشخصات سابجکت ها از نظر نوع پروتکل مورد آزمایش

QUESTION 3 :

با دستور find ، اندیس ابتدای درایه هایی که سطر یازدهم آن ها ، یک باشد را در target قرار می دهیم . سپس اندیس ابتدای درایه هایی که سطر دهم آن ها غیر صفر و سطر یازدهم آن ها صفر باشد را در nontarget قرار می دهیم .

QUESTION 4 :

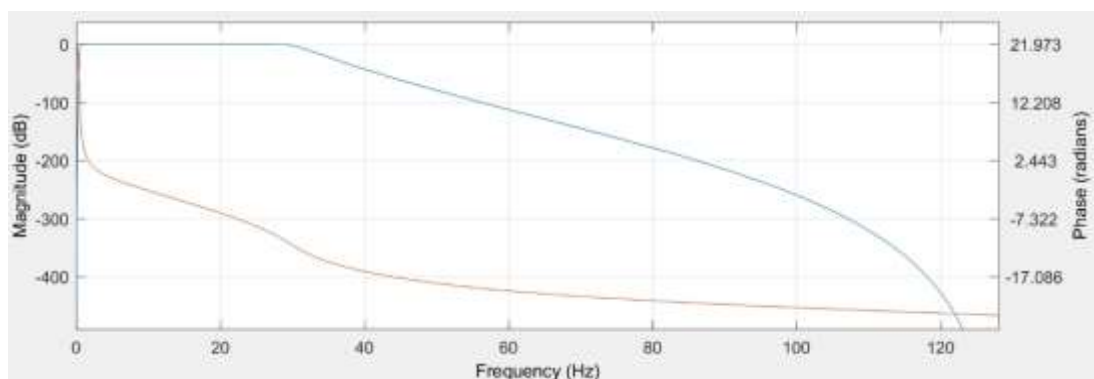
با استفاده از اندیس هایی که در بالا استخراج کردیم ، 800 میلی ثانیه معادل 205 درایه بعد از هر روشن شدن صفحه را جدا می کنیم و ماتریس های سه بعدی در صورت سؤال را را به دست می آوریم .

نکته (توضیحات درباره ی الگوریتم توابع IndexExtraction و TrialExtraction را به صورت comment در متن کد آورديم .

Part 3 :

QUESTION 1 :

در خود تابع TrialExtraction ابتدا سیگنال ها را از فیلتر میان گذر با باند 0.5 Hz تا 30 Hz عبور می دهیم . باید توجه داشته باشیم که سیگنال ها باید از همان اول فیلتر شوند و نمیتوانیم سیگنال های 800 میلی ثانیه ای را فیلتر کنیم . این کار به و دلیل غلط است . یک اینکه اثرات لبه ی بازه نادیده گرفته می شود . دوم اینکه چون بازه ی سیگنال محدود است مولفه های فرکانس پایین به خوبی فیلتر نمی شوند . برای ساخت فیلتر از signal Processing Toolbox متلب استفاده می کنیم . پاسخ فرکانسی فیلتر بدین صورت می باشد :

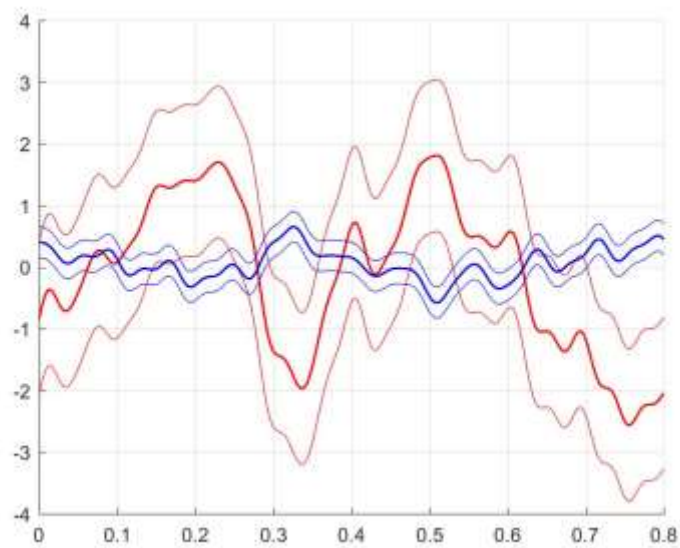


شکل 2 :

پاسخ فرکانسی فیلتر میان گذر استفاده شده در پروژه

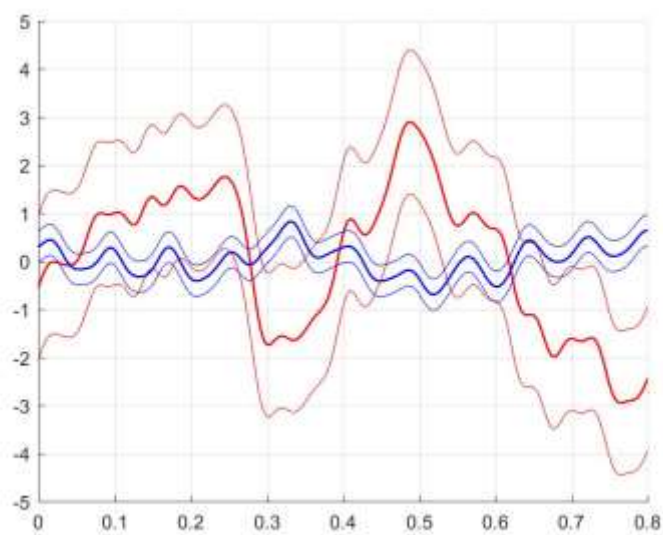
QUESTION 2 :

سیگنال های EEG را در trial های مختلف برای هر کانال محاسبه می کنیم . سپس انحراف معیار هر درایه را در طول trial ها محاسبه می کنیم و به همان اندازه در بالا و پایین هر سیگنال نمودار جدید رسم می کنیم .



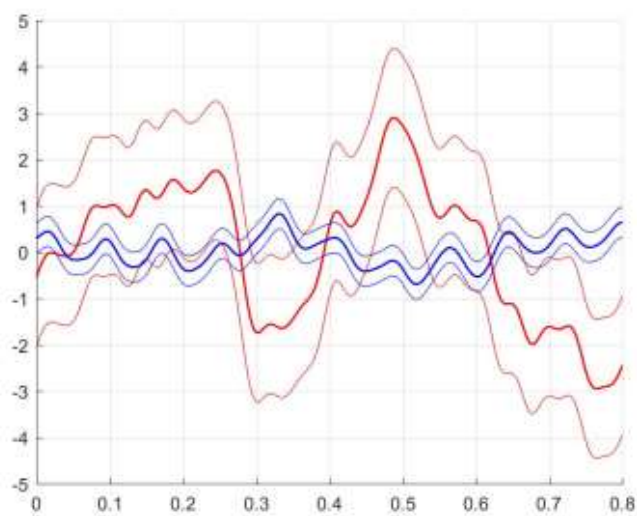
شکل 3:

سیگنال ERP کانال 1 EEG
Subject 8



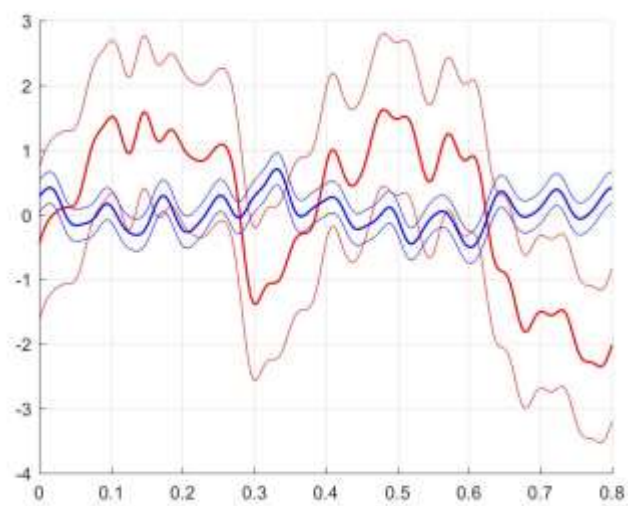
شکل 4:

سیگنال ERP کانال 2 EEG
Subject 8



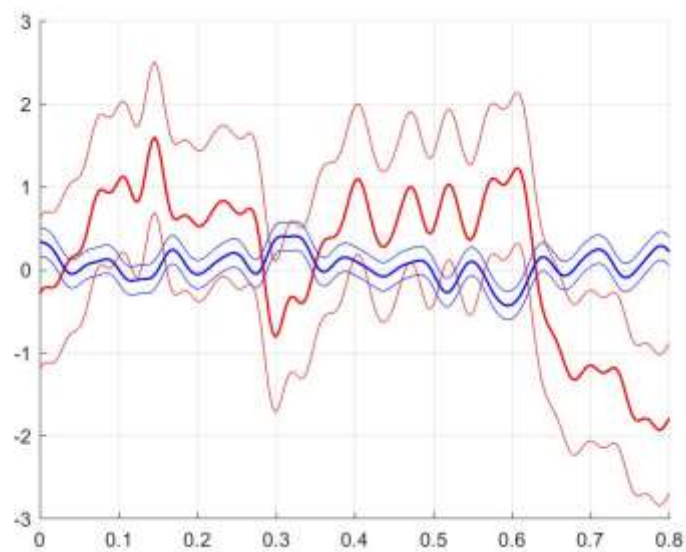
شکل 5:

سیگنال ERP کانال 3 EEG
Subject 8



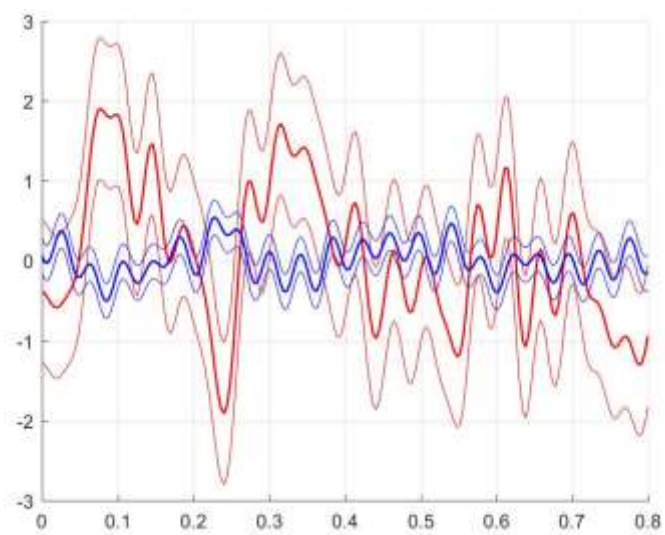
شکل 6:

سیگنال ERP کانال 4 EEG
Subject 8



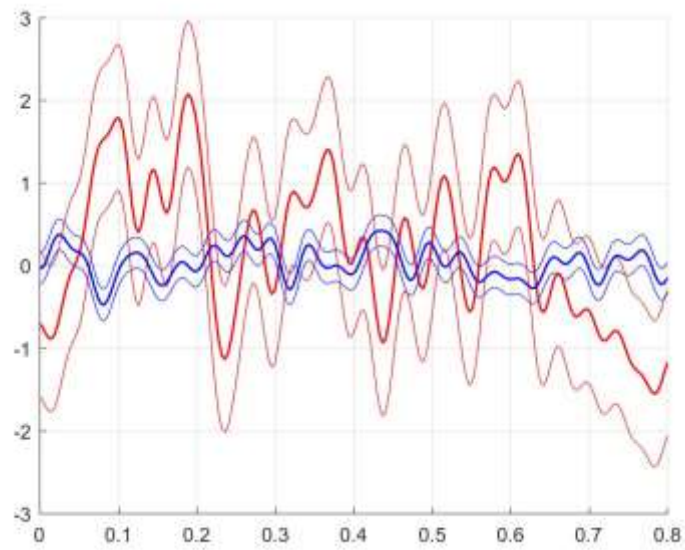
شکل 7:

سیگنال ERP کانال 5 EEG
Subject 8



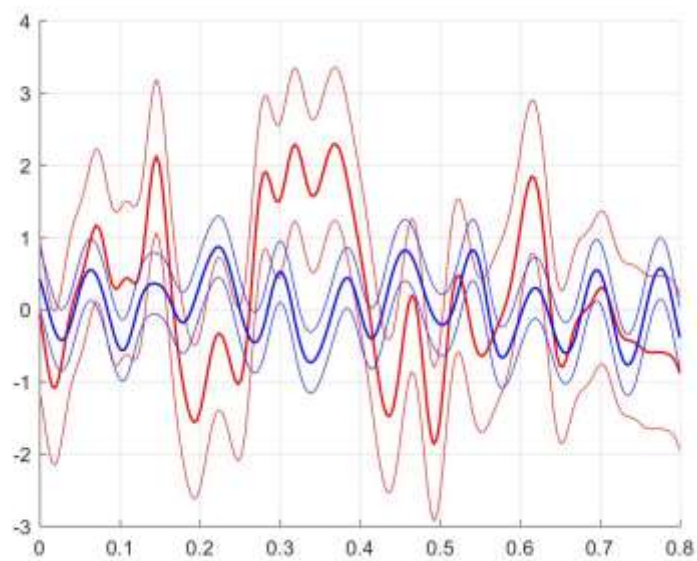
شکل 8:

سیگنال ERP کانال 6 EEG
Subject 8



شکل 9:

سیگنال ERP کانال 7 EEG
Subject 8



شکل 10:

سیگنال ERP کانال 8 EEG
Subject 8

QUESTION 3 :

در trial های nontarget ، شکل موج نوسانی و بدون تغییر فرم می باشد زیرا مغز در زمانی که تحریک Deviant نیامده به عملکرد عادی خود می پردازد و ورودی پیش بینی شده ای دارد . لذا برای ورودی های نزدیک پردازشی انجام نمی دهد و الگویی که تا به حال دریافت کرده را برای آن ها متصور می شود . به عنوان مثال با نگاه به شکل می توان دریافت که مغز در باند فرکانسی آلفا قرار دارد و دامنه ی آن هم تغییر چندانی ندارد . با بررسی دقیق تر طول موج سیگنال ها مثلا برای کانال 8 متوجه می شویم که فرکانس مشاهده شده با فرکانس تحریک ورودی (هرچند nontarget) برابر است . این موضوع جالب است زیرا ما بر روی خانه های روشن تمرکز نمی کنیم منتها مغز ما با تحریک هم فرکانس است . برداشت ما از این اتفاق این است که سلول های rod بدون توجه به نقطه ی تمرکز ما ورودی دریافت کرده و بخشی از مغز که نقطه ی خیره شدن و تمرکز (تنظیم عدسی) است ورودی دلخواه دریافت نکرده . اما ما چون برآیند spiking تمام نوروها را در EEG مشاهده می کنیم ، اثر نوسانات هماهنگ ورودی ها غالب است . اما در trial های target ، شکل موج پیش بینی نشده و غیر نوسانی است . دلیل این اتفاق این است که مغز تحریک Deviant دریافت کرده و مشغول پردازش هایی در لایه های مختلف مغز است . دامنه ی P300 که در شکل ها مشخص است پردازش در لایه های نسبتا بالای مغز را نشان می دهد . این تفاوت در target ها و nontarget ها تقریبا برای همه ی الکترود ها برقرار است .

QUESTION 4 :

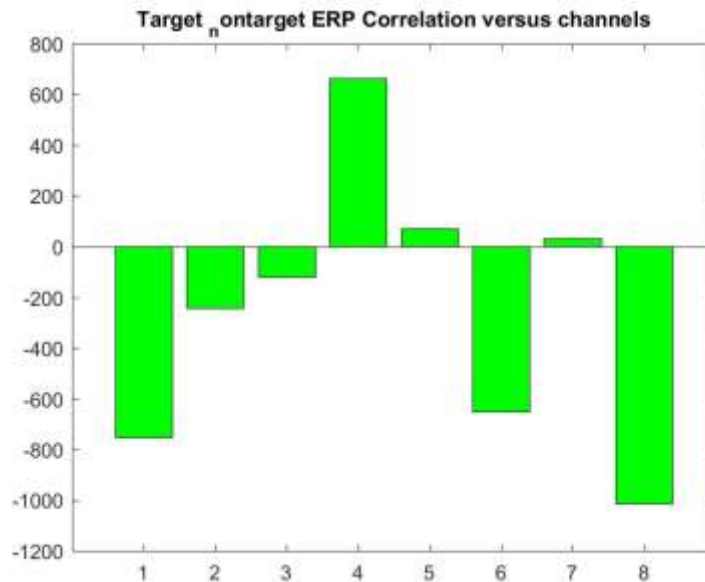
حال به دنبال الکترودی می گردیم که بیشترین تفاوت را در target و nontarget منعکس کند . با نگاه به شکل موج های هر 8 کانال می بینیم که تفاوت ها در همه ی کانال ها به خوبی مشاهده می شود و واقعا انتخاب یک الکترود کار بسیار سختی است . از جنبه های مختلفی می توان به این موضوع نگاه کرد . مثلا می توان گفت کانالی که سیگنال های ERP در target و nontarget ، correlation کمتری دارند ، بهتر است . با این تفکر مقدار C(target_ERP , nontarget_ERP) را در نقطه ی صفر برای هر 8 کانال محاسبه می کنیم . این مقدار در واقع به ما نشان می دهد که دو سیگنال در فضای نمونه ها چه مقدار بر یکدیگر تصویر دارند . هر چه این مقدار بیشتر باشد انتظار ما این است که کانال مربوطه کمتر تفاوت target و nontarget را منعکس می کند . برای مثال برای ساجکت شماره 8 داریم :

Rank	Channel Num	Correlation
1	7	18779324600
2	8	14019239362
3	6	15007
4	5	9214
5	4	180
6	3	136
7	2	0.52
8	1	-0.94

جدول 4 :

مقدار correlation سیگنال های target ERP و nontarget ERP

همان طور که ملاحظه می کنید با این معیار ، کانال 1 و 2 با اختلاف معنی داری کانال های بهتری برای تشخیص target بودن و nontarget بودن است . همچنین کانال های انتهایی نیز اطلاعات زیادی به ما نمی دهد . البته با مشاهده ی تفاوت بسیار زیاد اعداد جدول می توان به یک مشکل پی برد . این که برخی کانال ها به خودی خود دامنه ی کمی دارند و مقدار correlation معیار خوبی برای تفاوت شکل این دو نیست . لذا بر میانگین دو سیگنال آن ها را تقسیم میکنیم تا بتوان مقادیر را با هم مقایسه کرد .



شکل 11 :

مقدار Normalized Correlation سیگنال های target ERP و nontarget ERP کانال های مختلف

با مشاهده ی این نمودار متوجه می شویم که الکتروود های 1 و 8 کمترین میزان Correlation را در میان سیگنال های Target و nontarget دارند .

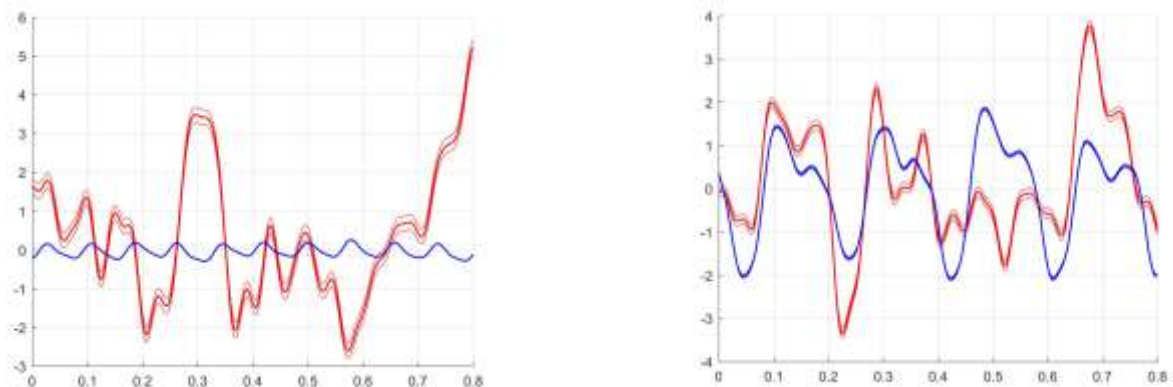
همچنین از منظر تحلیل واریانسی داده ها نیز می توان به موضوع نگاه کرد . از آنجاییکه مدل LDA براساس minimum کردن واریانس درون دسته ای شکل می گیرد پس انتظار داریم که شکل موج با واریانس کمتر ، درواقع با بازه ی اطمینان کوچک تر ، بهتر به Classification کمک کند . که با مشاهده ی شکل موج ها می بینیم که اکثرا شبیه همد و واقعا نمی توان تشخیص داد .

برای انتخاب بهترین زمان به بازه هایی نگاه می کنیم که بیشترین تفاوت را میان target و nontarget را منعکس کند . در واقع زمانی که مغز ورودی Deviant را هضم می کند . با نگاه به شکل ها عموما در بازه های 0 تا 200 ، 300 تا 400 و 600 تا 800 ما تفاوت معنی دار داریم اما نمی توان با قطعیت سخنی گفت .

QUESTION 5 :

. A

تقریباً تمام سیگنال های nontarget شکل نوسانی دارند برای مثال به شکل های زیر توجه فرمایید :



شکل 12 :

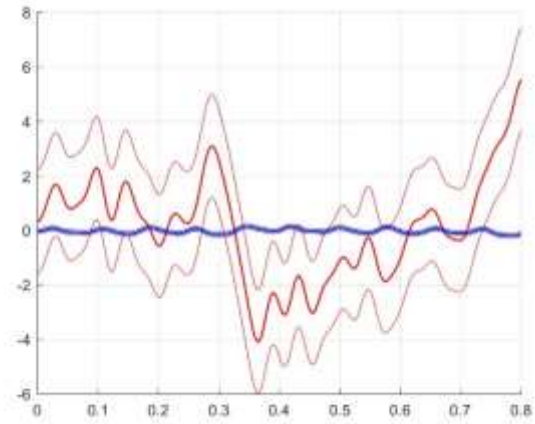
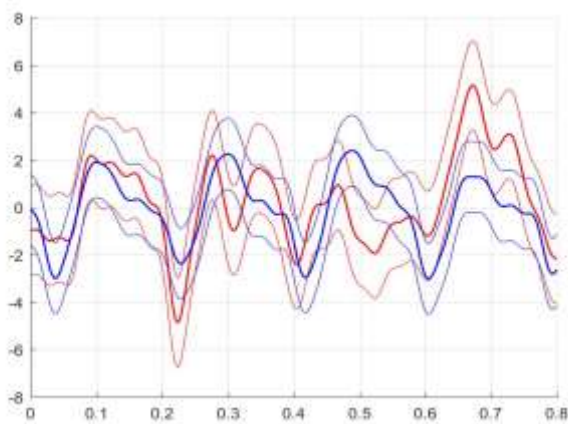
نوسانی بودن شکل موج ERP nontarget در هر دو پروتکل RC و SC

. b

برای اکثر سابجکت ها مانند قسمت قبل بهترین زمان ، در انتهای بازه و در میانه ی بازه حدود 400 میلی ثانیه است . البته در سابجکت 10 شکل موج ها بسیار شبیه به همند و کلاً با سابجکت های دیگر فرق دارد . در اینجا به نظر می رسد الکترو د 2 و 3 بهتر عمل می کنند .

. C

با توجه به شکل موج های ERP پروتکل RC و SC ، تفاوت عمده ای در دو دسته وجود دارد . تفاوت دامنه در پروتکل SC به شدت بیشتر از پروتکل RC است . زیرا در پروتکل SC احتمال روشن شدن هر خانه کمتر از RC است . لذا سورپرایز بیشتری به همراه دارد پس دامنه ی ولتاژ بیشتری نسبت به حالت RC دارد . همچنین در حالت nontarget نیز نوسانات در SC بسیار همگون تر و منظم تر است و در بعضی موارد یک سینوسی صاف و تمیز می باشد . این بدین علت می باشد که مغز شخص مدت بیشتری را صرف سینک کردن خود با نوسانات تصویر می کند .



شکل 13:

تفاوت شکل موج ها در پارادایم های RC و SC

. D

انتظار داریم که پروتکل SC بهتر از RC کار کند زیرا تفاوت عمده ی بین سیگنال های target و nontarget باعث بیشتر شدن فاصله ی دو دسته می شود و مدل LDA با خطای کم تری حاصل می شود . همچنین واریانس درون دسته ی nontarget در حالت SC بیشتر است . اما به دلایلی نامعلوم (البته paper به دلایلی اعم از خستگی چشم اشاره کرده که از نظر بنده قانع کننده نبود) پروتکل RC بهتر جواب می دهد .

QUESTION 6 :

نتایج ما تا به اینجای کار این بوده که مشخصه ای در سیگنال های target وجود دارد که باعث تمایز آنها با nontarget شده . اتفاقی در این trial ها می افتد که از آن می توان در دسته بندی دو اتفاق بصری استفاده کرد . پردازشی که در این اتفاقات در عمق مغز رخ میدهد باعث تفاوت ERP آن ها می شود . همچنین می توان نتیجه گرفت با دادن وردی نوسانی حتی بدون نیاز به تمرکز روی آن ها می توان سیگنال های مغزی را به Oscillation وادار کرد که از این حیث نیز می توان در فناوری های نوین BCI از آن استفاده کرد .

PART 4:

QUESTION 1 :

همانطور که گفته شد ، در این بخش سیگنال ها را DownSample میکنیم . پنی از فرکانس 256 Hz به 64Hz می رسانیم . با این کار تعداد feature های ما به $8 \times 51 = 408$ می رسد .

QUESTION 2 :

در این بخش یک ماتریس 2700×408 برای پروتکل SC و یک ماتریس 900×408 برای پروتکل RC به مدل LDA تحویل می دهیم . برای این کار از دستور fittediscr استفاده می کنیم . این دستور به طور دیفالت یک مدل LDA را

برای Classification پیاده می کند . برای پیاده سازی 5_Fold_CrossValidation یک تابع نوشتیم به نام cross_LDA که ماتریس دیتا و label را با هم و تعداد fold ها را می گیرد . سپس ماتریس را shuffle میکند . هر بار یک بخش از دیتا (مثلا در این مسئله یک پنجم) را نگه می دارد و LDA را روی بقیه train میکند و روی بقیه تست می کند . در نهایت عددی به عنوان درصد صحت به عنوان خروجی به ما می دهد . حال به تحلیل Performance Measure های به دست آمده می پردازیم . در سابجکت مورد مطالعه (سابجکت شماره 8) Confusion Matrix روی داده های train را بدست می آوریم .

143	7
12	738

حال با استفاده از این ماتریس درصد صحت هایی که می خواهیم را بدست می آوریم .

درصد صحت روی کل داده های train = 97.89 %

درصد صحت روی داده های target دیتای train = 92.26 %

درصد صحت CrossValidation روی دیتای train = 78.67 %

همان طور که ملاحظه می کنید با اعمال CrossValidation درصد صحت روی داده ها کاهش یافت زیرا بخشی از داده های training را از دست می دهیم .

نکته ی مهم این است که داده های ما balance نیستند . یعنی تعداد target ها از تعداد nontarget ها به مراتب کم تر است لذا نگاه مطلق به درصد صحت روی کل داده ها کار اشتباهی است . چون ممکن است کل target ها اشتباه دسته بندی شوند اما باز هم درصد صحت خوبی بگیریم . لذا به ملاک های خود درصد صحت روی داده های target را نیز اضافه می کنیم . درصد صحت روی داده های تست نیز به صورت زیر است :

81	69
108	642

درصد صحت روی کل داده های test : 80.33 %

درصد صحت روی target های دیتای test : 42.86 %

می بینیم که درصد صحت روی داده های تست کاهش چشم گیری پیدا کرده . همچنین با وجود درصد صحت بالا روی کل داده ها ، درصد صحت پایینی روی داده های target داریم .

QUESTION 3 :

برای تشخیص کلمه ای که شخص قصد Spell کردن آن را دارد ابتدا داده ها را بر اساس زمان sort می کنیم . حال برای مشخص شدن هر حرف ، 15 بار seep شدن کل صفحه را capture میکنیم و آن را آنالیز می کنیم . برای پروتکل SC باید $15 \times 36 = 540$ trial و برای پروتکل RC ، $15 \times 12 = 180$ trial . حال با دیدن هر target کلمه ی نشان داده شده روی تصویر را با استفاده از ردیف 10 ماتریس داده شده می یابیم و به آن حرف یک رای می دهیم . در انتهای 15 trial حرفی که بیشترین رای را داشت به عنوان خروجی نشان می دهیم . برای پروتکل RC اما با مشاهده ی target به تمام درایه های آن سطر یا ستون رای می دهیم . در انتهای 15 trial تمام درایه ها را میخوانیم و آن را که بیشترین تعداد رای را داشت به عنوان خروجی می دهیم . خروجی ماتریسی است به همان فرمت ماتریسی که در مقاله آمده بود .

نکته مهم : بر خلاف حرفی که paper زده بود ، بعضی از سابجکت ها روی LUCAS ، Train و روی WATER تست شده بودند .

نحوه ی کار را برای subject 8 مشاهده می کنیم :

5	6	6	8	14	8
5	6	6	8	14	8
7	8	8	10	10	10
6	7	7	9	15	9
5	6	6	8	14	8
4	5	5	7	13	7

13	15	12	12	13	12
15	7	4	4	5	4
19	11	8	8	9	8
22	14	11	11	12	11
16	8	5	5	6	5
16	8	5	5	6	5

24	16	10	0	14	13
16	8	3	0	6	5
20	12	4	0	10	9
22	14	4	0	12	11
17	9	5	0	7	6
17	9	5	0	7	6

8	6	8	7	14	7
3	1	3	2	9	2
2	0	2	1	8	1
3	1	3	2	9	2
3	1	3	2	9	2
2	0	2	1	8	1

4	2	6	3	6	12
6	4	8	5	8	14
9	7	11	8	11	17
5	3	7	4	7	13
4	2	6	3	6	12
3	1	5	2	5	11

جدول ۵ :

از چپ بالا تا راست پایین
تعداد رای های به هر حرف Spell شده برای سابجکت

با مقایسه این جداول با شکل متوجه میشویم که سابجکت می خواسته کلمه ی WATER را Spell کند که با خطا در یک حرف موفق به این کار شده . خروجی ابتدایی کد برای این سابجکت جدولی به شکل زیر است .

2	0	0	0	4	0
0	0	0	0	0	0
0	0	0	0	1	5
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

جدول ۶ :

خروجی ابتدایی کد که ترتیب لغات را در جدول حروف مشخص می کند

سپس ماتریسی تعریف کردیم که خانه های آن با کد ascii حروف و شماره ها به ترتیب شکل پر شده . حال با پیدا کردن خانه های مشخص شده در ماتریس ascii کلمه را Spell می کنیم . خروجی نهایی کد بدین شکل است :

“A*ER”

QUESTION 4 :

برای بدست آوردن Coeffs مدلی که در قسمت Classification Learner متلب ساخته بودیم را زیر و رو کردیم منتها چنین قسمتی پیدا نکردیم . لذا مجبور شدیم برای این بخش کار دوباره ای انجام دهیم و باز از fitdiscr استفاده کنیم . حال بردار Coeffs مدل LDA را بررسی می کنیم . ابتدا بردار حاصله را sort میکنیم تا تاثیر گذارترین مولفه ها را بدست

آوریم . سپس با بدست آوردن شماره ی feature ها به بررسی این می پردازیم که کدام الکتروود و کدام زمان از همه تاثیر گذار تر است .

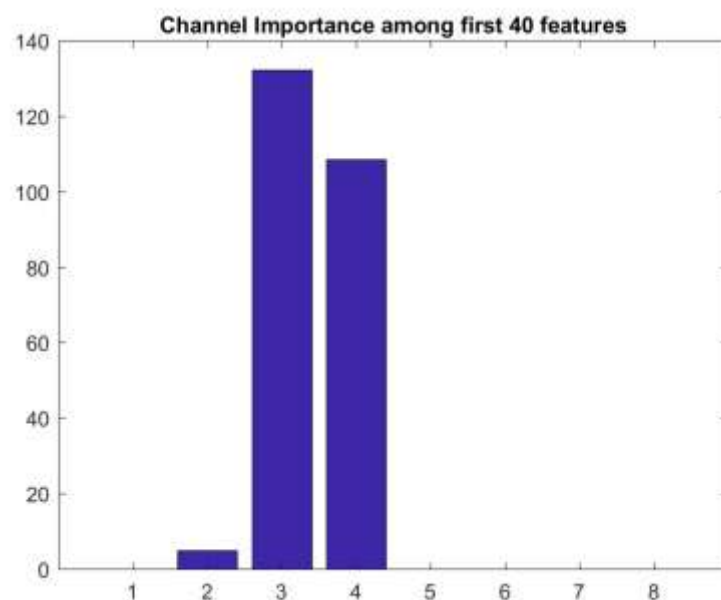
317	178	192	321	216	107	222	117	227	58	329	65
74	62	221	108	347	156	134	111	109	113	68	125
139	234	204	155	112	235	318	320	66	79	183	250
241	230	114	73	218	67	190	78	135	140	208	186
173	243	249	229	116	247	115	237	185	228	236	223
210	177	171	220	214	215	211	191	219	248	246	157
166	170	176	158	172	244	245	175	213	209	212	242
		167	168	169	163	164	162	161	165	160	159

جدول ۷ :

ترتیب تاثیر گذاری feature ها روی جدا سازی target ها و nontarget ها به ترتیب از بالا راست تا چپ پایین از بی تاثیر ترین تا تاثیر گذار ترین

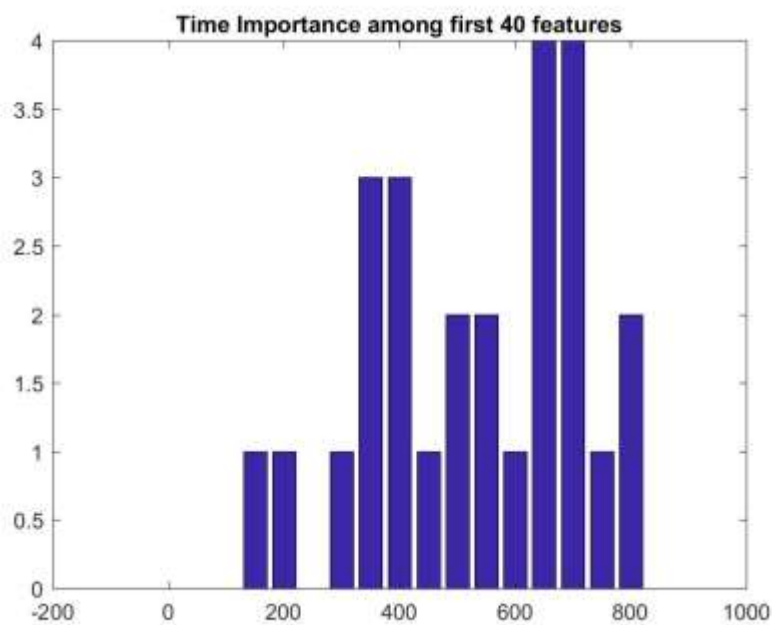
همان طور که می بینید 15 مولفه ی تاثیر گذار اول و تنها Coefficient های بالای 2 در بازه ی feature های شماره 153 تا 204 هستند که متعلق به الکتروود شماره 3 می باشند . همچنین اکثر آن ها در بازه ی اعداد 160 تا 170 هستند یعنی حدودا از زمان 100 milisecond تا 266 millisecond زمانی است که تحریک Deviant بیشترین تاثیر را روی جداسازی target از nontarget دارد . پس از آن نیز اکثر اعداد در بازه ی 204 تا 255 قرار دارند که یعنی الکتروود شماره 4 پس از الکتروود 3 از تاثیر گذار ترین الکتروود ها برای جداسازی target از nontarget است .

برای بررسی دقیق و علمی تر این موضوع ، 40 مولفه ی اول تاثیر گذار را از Coeffs استخراج می کنیم . می دانیم حاصل تقسیم این اعداد بر طول هر سیگنال ، شماره ی کانال را به ما می دهد . لذا حاصل تقسیم آن ها را حساب می کنیم . حال این اعداد که نمایانگر شماره ی کانال ها ست را هیستوگرام می کنیم . اندازه ی هر میله نمایانگر میزان اهمیت آن کانال است . برای دقیق تر شدن و دخیل شدن رتبه و اهمیت آن ها ، طول میله ها را در Coeffs مربوطه ضرب می کنیم یعنی هر کانال به اندازه Coefficients خود در بزرگ شدن اندازه میله Contribution داشته باشد . نمودار بدست آمده شهود بسیار خوبی از اهمیت کانال ها به ما می دهد . برای پیدا کردن زمان مهم نیز کافی است باقی مانده ی اعداد را بر طول سیگنال بگیریم . باقی کار ها مانند پیدا کردن کانال خوب است



شکل 14 :

کانال های مهم در Subject 8



شکل 15 :

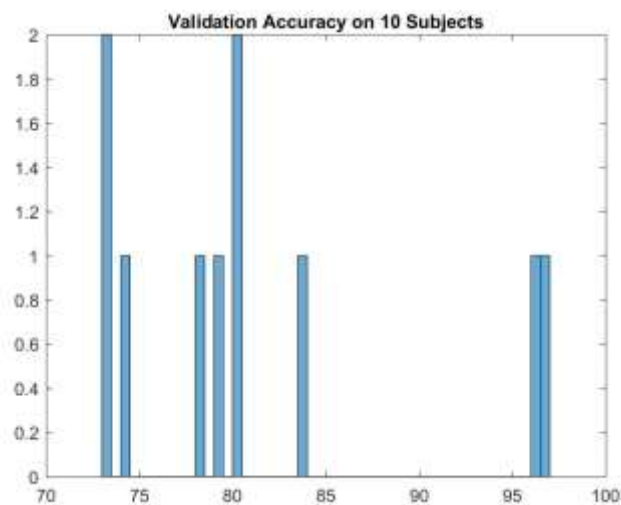
کانال های مهم در Subject 8

طبق این نمودار ها ، کانال 3 و 4 کانال های مهمی هستند . همچنین زمان های 600 تا 800 میلی ثانیه و 300 تا 400 میلی ثانیه نیز زمان های مهم هستند . نظر قاطع تر را بعد از بررسی تمام سابجکت ها ارائه می دهیم .

QUESTION 5 :

A .

هیستوگرام درصد صحت روی سابجکت ها به شکل زیر می باشد .



شکل 16 :

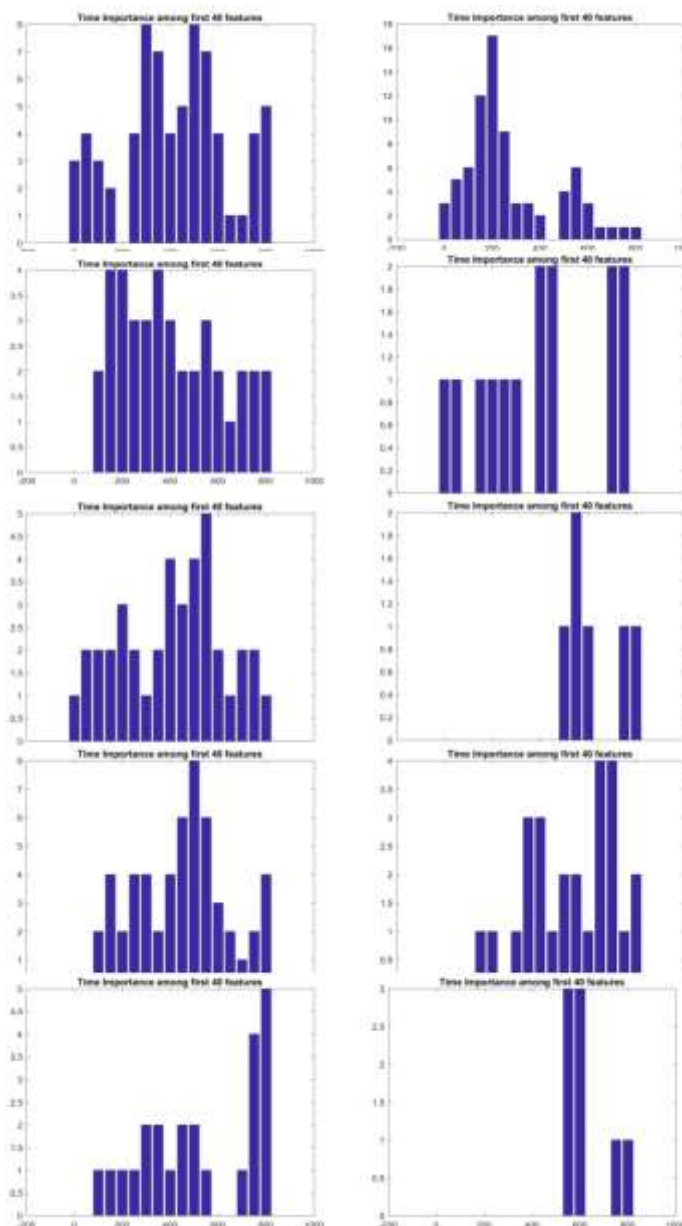
هیستوگرام درصد صحت subject های مختلف

B .

سیستم برای هیچ یک از افراد موفق نشده که لغت را به طور کامل تایپ کند و دقت تایپ اکثرا حول بازه ی 70 الی 80 درصد بوده .

C .

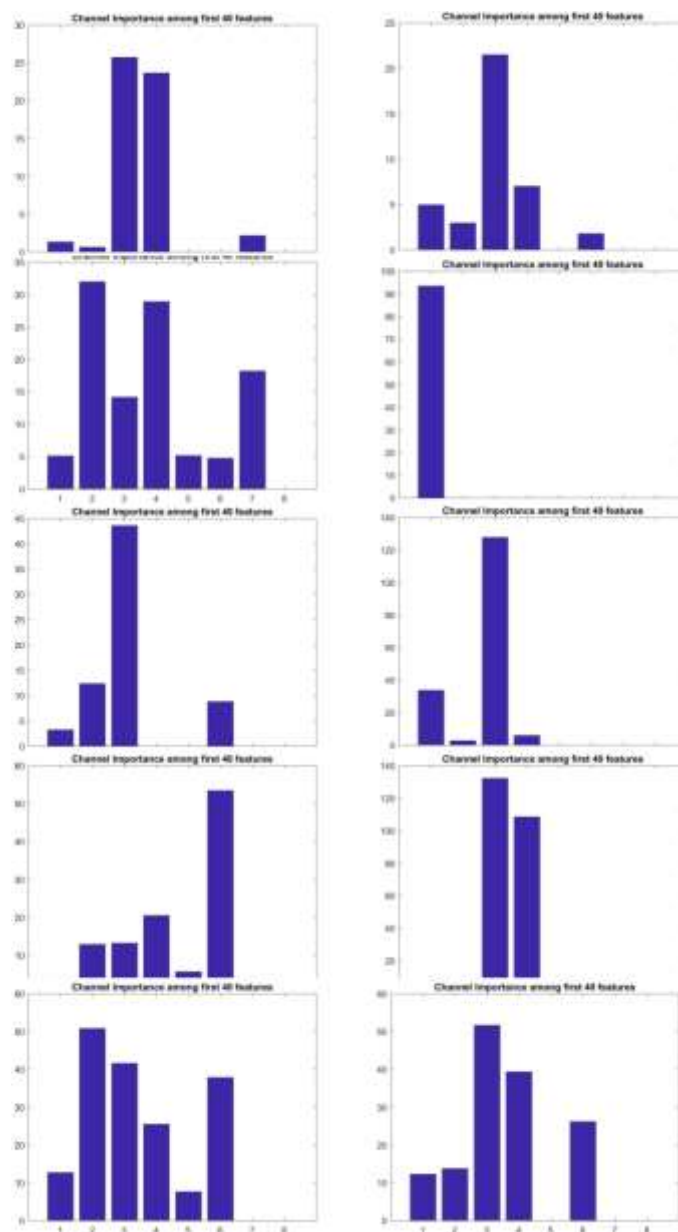
با همان متدی که در قسمت قبل ، الکتروود ها را از لحاظ اهمیت طبقه بندی می کردیم تمام نمودار ها را به دست می آوریم و برای سابجکت های مختلف می کشیم .



شکل 17 :

میزان تاثیر گذاری زمان های مختلف روی Classification

همان طور که مشاهده می کنید ، مکان پیک زدن این نمودار ها با هم متفاوت است یعنی برای سباجکت های مختلف حدود زمانی مختلف است اما به صورت بسیار rough می توان نظراتی را داد . مثلا سباجکت ها 1 و 2 و 3 ، پیک سراسری نمودار خود را در زمانی بین 300 تا 400 میلی ثانیه زدند . همچنین سباجکت های 4 و 6 و 8 و 9 و 10 پیک های سراسری خود را در انتهای بازه یعنی بین 600 تا 800 میلی ثانیه زدند . نظر نهایی این است که اگر مجبور بودیم بازه ی زمانی ای را انتخاب کنیم ، بازه های 300 الی 400 و 600 الی 800 را بازه های مناسبی بودند .



شکل 18 :

میزان تاثیر گذاری کانال های مختلف روی Classification

برای بررسی بهترین الکتروود نیز مانند قبل عمل میکنیم . مشاهده می شود که بری اکثر قریب به اتفاق سابیکت ها الکتروود ۳ و تعدادی نیز الکتروود ۴ ، بهترین الکتروود است .

. D

نظر دادن درباره عملکرد متد RC و SC با مشاهدات محدود ، کاری بسیار مشکل است اما می توان حرف هایی را ولو نادقیق درباره ی عملکرد دو پروتکل زد . میانگین تعداد حروف Spell شده در متد SC ، 1.5 حرف و در متد RC ، 2.25 حرف می باشد که این بدین معناست که به طور میانگین عملکرد در پارادایم RC بهتر از SC بوده . مشاهده می شود که پس از CrossValidation درصد صحت روی داده های train در پارادایم SC بیشتر از RC است که ممکن است این بدین معنا باشد که عملیات training روی SC بیشتر در معرض خطر Overfitting قرار دارد .

. E

در قسمت قبل هم گفتیم که به علت کمبودن احتمال روشن شدن خانه ای در متد SC ، این متد باید نتیجه ی بهتری بدهد که با نتایج نیز سازگار است .

Question 6 :

ابتدا به بررسی سابجکت ها دیگر می پردازیم و در آخر نتایج کلی خود را ثبت می کنیم :

SubjectNum	Confusion Matrix		Total Accuracy	Target Accuracy
S1	2	73	96.37%	7.41%
	25	2600		
S2	13	62	96.93%	93.24%
	21	2604		
S3	59	91	73.33%	28.37%
	149	601		
S4	68	82	79.22%	39.31%
	105	645		
S5	74	76	78.11%	37.95%
	121	629		
S6	48	102	73.44%	25.95%
	137	613		
S7	71	79	74.44%	31.98%
	151	599		
S8	81	69	80.33%	42.86%
	108	642		
S9	83	67	80.00%	42.35%
	311	637		
S10	79	71	83.89%	51.63%
	74	676		

جدول 8 :

Performance Measure از زوایای مختلف برای مدل LDA روی دیتای *test*

SubjectNum	Confusion Matrix		Total Accuracy	Target Accuracy	Total Accuracy (Cross Validation)
S ₁	48	27	98.7%	85.71	95.78%
	8	2617			
S ₂	60	15	99.22%	90.91%	96.33%
	6	2619			
S ₃	120	30	94.56%	86.33%	71.89%
	19	731			
S ₄	135	15	97.67%	95.74%	76.44%
	6	744			
S ₅	135	18	96.78%	92.31%	75.67%
	11	739			
S ₆	120	30	94.78 %	87.59%	70.00%
	17	733			
S ₇	134	16	97.22%	93.71%	75.22%
	9	741			
S ₈	143	7	97.89%	92.26%	78.67%
	12	738			
S ₉	133	17	97.89%	98.52%	79.78%
	2	748			
S ₁₀	130	20	97.11%	95.59%	77.33%
	6	744			

جدول 9:

Performance Measure از زوایای مختلف برای مدل LDA روی دیتای *train*

SubjectNum	Spelled Word	Accuracy
S ₁	*BKWO	0 %
S ₂	MUKAG	60 %
S ₃	*U*AV	40 %
S ₄	*U*AS	60 %
S ₅	*A*E*	40%
S ₆	*****	0%0%
S ₇	*ATE*	60 %
S ₈	*A*ER	60%
S ₉	*A*ER	60 %
S ₁₀	**YER	40 %

جدول 10:

ماتریس تصمیم، کلمه ی *Spell* شده و درصد صحیح بودن حروف برای تمام سابجکت ها

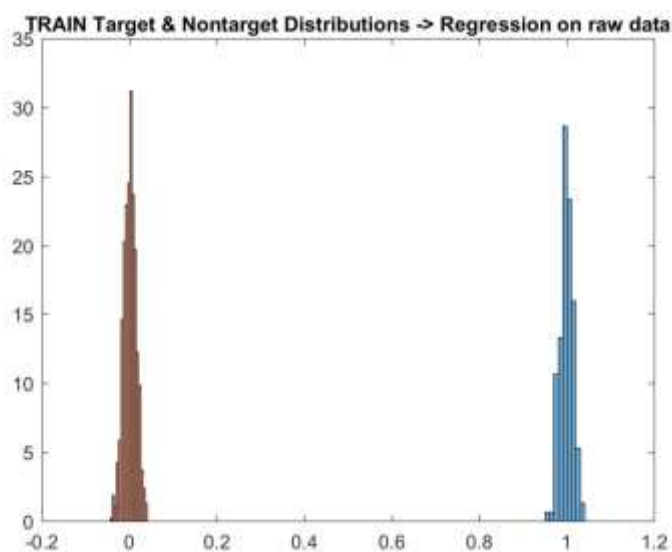
نکته : در تصمیم گیری نهایی بسیار strict عمل کردیم و موارد بسیاری بودند که دو حرف با هم انتخاب شده بودند که یکی از آن ها درست بود اما آن ها را در دسته ی نامشخص (*) قرار دادیم . لذا حروف spell شده ی بالا حروفی هستند که صد درصد درست بودند .

نتیجه گیری نهایی : به نظر می رسد که متد ارائه شده در مقاله با Accuracy نسبتاً خوبی عمل میکند . این اتفاق به نظر من ناشی از انتخاب feature های خوب است . در مسئله ی Oddball یا deviant Stimulus ، بهترین feature همان amplitude سیگنال فیلتر شده ی EEG است و به بهترین نحو ممکن انتظارات را برآورده می کند . گواه این مدعا نه چندان بد مدل روی داده های تست است .

Part 5 :

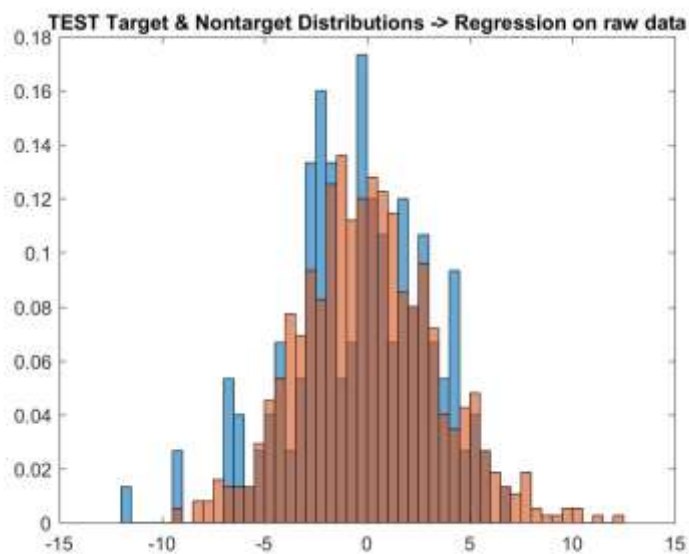
در این قسمت می خواهیم با امتحان متدهای مختلف ، دسته بندی را بهبود ببخشیم .

A . به مانند تمرین قبلی از روش regression استفاده می کنیم . ابتدا داده های خام و بدون downsample شدن را به عنوان feature های regression قرار می دهیم . یعنی ماتریس train_X_prime را که داده های اولیه هستند به همراه label های آن ها را به fitlm می دهیم . مدل را روی داده های TRAIN و داده های TEST امتحان می کنیم و هیستوگرام هر یک را می کشیم .



شکل 19 :

توزیع label های predict شده توسط Regression برای داده های train در حالت Raw

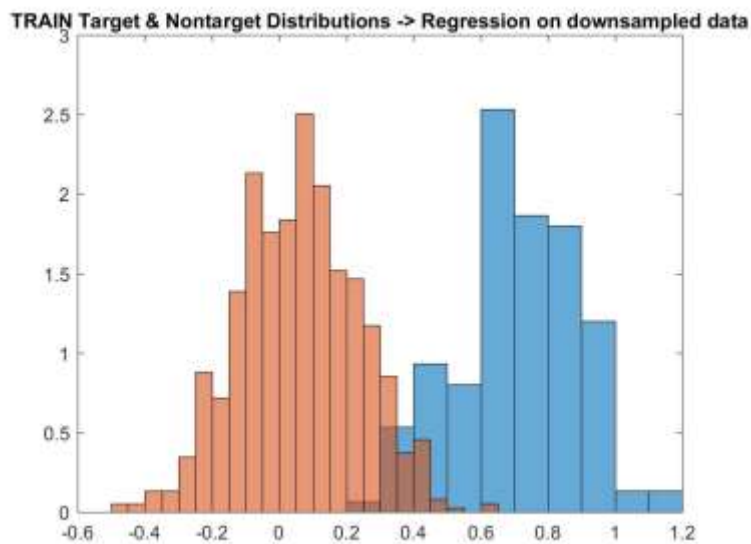


شکل 20 :

توزیع label های predict شده توسط Regression برای داده های test در حالت Raw

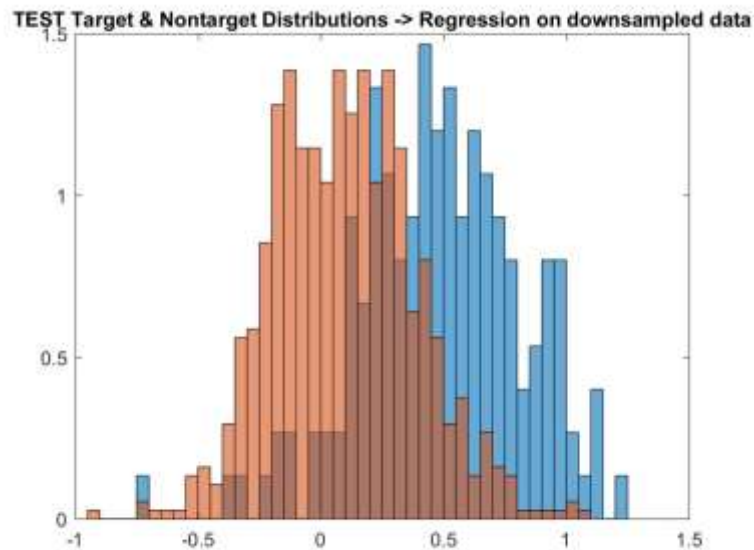
همانطور که می بینیم مدل خطی روی خود داده های train ، target ها و nontarget ها را صد درصد از هم جدا می کند اما در پیش بینی روی داده های test بسیار بد عمل می کند .

حال روی داده های DownSample شده این کار را انجام می دهیم . نتایج به صورت زیر می باشد :



شکل 21 :

توزیع label های predict شده توسط Regression برای داده های train در حالت DownSample شده



شکل 22 :

توزیع label های predict شده توسط Regression برای داده های test در حالت DownSample شده

همانطور که مشاهده می کنید مدلی که روی داده های DownSample شده Fit کردیم روی داده های train به خوبی داده های raw عمل نکرده و کمی overlap دارند اما در داده های test به مراتب بهتر از مدل قبلی عمل کرده .

حال چالش بعدی انتخاب threshold برای انتخاب target و nontarget است . روش پیشنهادی ما این است که عددی به عنوان threshold انتخاب شود که LOSS را در train مینیمم کند . برای پیدا کردن این نقطه از روش Iterative Binary Search استفاده می کنیم . بدین صورت که ابتدا Error را در نقاط آخرین target و آخرین nontarget حساب می کنیم . سپس بازه را به دو قسمت تقسیم میکنیم و Error را در آن نقطه محاسبه می کنیم . با نصف کردن بازه ها این کار را ادامه می دهیم تا آنجا که Error تغییر نکند . Error را بدین شکل تعریف می کنیم :

$$\text{Error} = (\# \text{Missclassified_Target} / \# \text{Total_Target}) + (\# \text{Missclassified_NonTarget} / \# \text{Total_NonTarget})$$

با این تعریف Error ، متأسفانه نتیجه ی خوبی نگرفتیم . سپس تعریف را اصلاح کردیم . Threshold را جایی قرار دادیم که بالاترین درصد صحت را بگیریم . اما باز هم نتیجه ی خوبی نگرفتیم . یعنی درصد صحت بالا معادل با تعداد حروف Spell شده ی زیاد نیست . به طور تجربی threshold را عوض کردیم و یافتیم که نقطه ی 0.4 مقدار معقولی برای threshold است .

SubjectNum	Spelled Word	Accuracy
S ₃	*USX*	20%
S ₄	LU*AS	80%
S ₅	***AS	20%
S ₇	*ASER	60%
S ₈	WASER	80%
S ₉	WA*ER	80%
S ₁₀	W*Y*R	40%

جدول 10 :

لغات Spell شده توسط مدل Regression

B. در این بخش می خواهیم کاری شبیه STA در بحث Encoding را انجام دهیم . با این تفاوت که این کار را به دو روش انجام می دهیم . در ابتدا بردار Target_ERP و Non_Target_ERP را پیدا می کنیم . سپس به جای اینکه نقاط را روی آن ها تصویر کنیم ، فاصله ی نقاط را با این دو نقطه که نماینده ی دسته های target و nontarget هستند می سنجیم و کاری شبیه به کاری که در KNN انجام می دهند را پیاده سازی می کنیم . در ابتدا بدون هیچ امیدی این روش را امتحان کردیم زیرا به علت نویز بالای کانال های EEG ، فاصله معیار مناسبی برای طبقه بندی نیست . اما با چیز جالبی مواجه شدیم . این مدل برای پروتکل SC بسیار خوب عمل می کند اما در پروتکل RC ناکار آمد است . دلیل این امر به نظر ما به همان موضوع واریانس داده ها بر می گردد . میزان پخش شدگی و درهم تنیدگی داده ها در پروتکل SC کمتر از پروتکل RC است . به نحوی کاری که ما کردیم ورژن ابتدایی تر LDA بود که معیار فاصله دسته ها در آن دخیل نشده بود . نتایج کار در برخی از سابجکت های قابل قبول بدین شکل است :

SubjectNum	Confusion Matrix		Total Accuracy	Target Accuracy	Spelled Word	Accuracy
S ₁	41	34	66.48%	4.5%	DVKOS	40%
	871	1754				
S ₂	47	28	78.19%	7.73%	*UKAS	80%
	561	2064				
S ₃	90	60	64.67%	25%	LU*A5	60%
	258	492				
S ₄	76	74	67.44%	28.66%	L3U**	40%
	122	628				

جدول 11 :

کلمات Spell شده با مدل Average