

הקדמה

טכניקת EEG (Electroencephalography) היא שיטת רישום של פעילות חשמלית מאיזורים שונים במוח, באמצעות קליטת רישומים מאלקטרודות המונחות על-פני הקרקפת. בעבודה זו ביצענו מספר ניתוחי נתוני EEG של 19 אלקטרודות על בסיס נתוניהם של 4 נבדקים שנאספו כ-100 דקות לפני שחוו התקף אפילפטי ועד תחילת ההתקף עצמו, תוך שימוש במטלב, במטרה לזהות האם קיים שינוי ויזואלי בפעילות החשמלית של הלוקים באפילפסיה טרם ההתקף.

אפילפסיה היא הפרעה בפעילות החשמלית של נוירונים במוח הלוקה בהפרעה, ללא גורמים מעוררים חיצוניים, הגורמת לפרכוסים חוזרים ולא נשלטים. פעילות מוחית תקינה מאופיינת בסנכרון נמוך יחסית בין איזורים מוחיים שונים, בעוד שפעילות מסונכרנת מידי מאפיינת אפילפסיה. הפעילות החשמלית הלא תקינה עשויה להיות במקורה במקום בודד במוח (focal) – במקרה זה, מקור ההתרחשות ישפיע על הסימפטום הנראה (למשל, אם מקורה באיזורי העיבוד המוטורים, נצפה לביטוי נראה בתנועות גפיים מהירות) או פעילות חשמלית לא תקינה שמתחילה ומאפיינת בבת אחת את כל המוח (general). ניתן לראות את התפרצות ההתקף בפעילות החשמלית המוחית באמצעות שימוש ב-EEG באמצעות ירידה בתדר ועליה חדה בעוצמת הפעילות המוחית בפרק זמן נתון. בתחילה, נראה כיצד חילצנו את 342 המאפיינים השונים של עוצמת התדרים לכל חלון זמן ולכל אלקטרודה. לאחר מכן, נראה כיצד ביצענו PCA (Principal Component Analysis) מ-342 מאפיינים ל-3 מאפיינים. הליך ה-PCA מבצע טרנספורמציה ליניארית אורתוגונלית לנתוני המאפיינים, על מנת להעבירם למערכת קוארדינטות בה המאפיינים בעלי השונות הגדולה ביותר מהווים את מערכות הצירים השונות. שיטה זו מובילה להפחתה במספר המימדים שבהם באים לידי ביטוי נתונים, כך שהנתונים יהיו קלים יותר ולרב ויזואלים יותר לפרשנות (ויזואלים יותר - במקרים בו מתבצע מעבר ל-2-3 מימדים בלבד), תוך שמירה על אובדן מידע מינימלי בעקבות שמירה על הנתונים מהמאפיינים בעלי השונות הגדולה ביותר בלבד.

שיטה

בראש ובראשונה טענו את נתוני הקבצים באמצעות פונקציית dir. לאחר מכן, באמצעות פונקציית cellfun (שמפעילה פונקציות על מערכי תאים) isempty (שמחזירה האם תא הוא ריק או לא) ו-regexp (שמחזירה את המיקום ההתחלתי של ביטוי מסוים במחרוזת מסוימת), בדקנו את כל שמות הקבצים והאם הם עומדים בקריטריון הצורה הנדרש. לבסוף חילצנו את מספרי הנבדקים ומספר ההתקף שלהם בעזרת פונקציית regexp עם הביטוי '/d*' ועם התכונה 'match' כך שתחזיר במערך תאים לא את המיקום ההתחלתי של המספרים, אלא את ערכם.

```
fileNames = dir('..\DATA_DIR\*');
filenames = {fileNames.name};
not_match = cellfun(@isempty, regexp(filenames, 'p\d+_s\d+.mat'));
filenames = filenames(~not_match);
nums_Pati_Seiz = regexp(filenames, '\d*', 'match');
```

כעת עברנו להגדרת פרמטרי הניסוי. חילצנו את מספר הנבדקים, קבענו את תדר הדגימה (250), קבענו את אורך החלונות הרצוי והחפיפה ביניהם בשניות (לטובת חלוקת הסיגנל לסגמנטים) ואת אורך החלונות הרצוי והחפיפה ביניהם בשניות (לטובת מדידת עוצמת התדרים לכל סגמנט) ולאחר מכן כפלנו כל אחד מהמשתנים הללו בתדר הדגימה כדי שיהיו לנו הערכים המייצגים את מספר השניות הרצוי. יצרנו משתנה סף לבדיקת המאפיין של spectral edge ומשתנה הקובע את המימד אליו אנו רוצים לדחוס את המאפיינים בניתוח ה-PCA. יצרנו משתנה המכיל את טווח התדרים בהם נעבוד (1-40) ומבנה המכיל את אותם תדרים, כאשר הם מחולקים לפי רצועות מוסכמות. לבסוף יצרנו מערך תאים שיכיל את מטריצות הקוואריאנס לניתוח ה-PCA הסופי.

```
Hz = 1:0.5:40;
freqs = struct('delta', [1 4.5], ...
               'theta', [4.5 8], ...
               'Lalpha', [8 11.5], ...
               'Halpha', [11.5 15], ...
               'beta', [15 30], ...
               'gamma', [30 40]);
```

בשלב זה התחילה לולאת עיבוד המידע והצגתו. עבור כל התקף (בניסוי זה, מאחר וכל סיגנל קדם-התקף מוקלט מנבדק אחר, ניתן להגיד גם עבור כל נבדק) אנו מייבאים את קובץ הנתונים בעזרת הפונקציה load ושימוש בשם הקובץ אותו ייבאנו קודם לכן. השלב הבא הינו שלב עיבוד המידע, בו יש לנו שני פלטים (ממוצעי המאפיינים ומטריצת המאפיינים – על המאפיינים יורחב בהמשך), הנשמרים במבנים לפי מספר הנבדק, בעבור הפונקציה Signal אותה כתבנו ונרחיב עליה בהמשך.

```
Data.(sprintf('Sub_%d',str2double(nums_Pati_Seiz{n}{1,1}))) =
load(['..\DATA_DIR\' filenames{n}]);

All_Data.(sprintf('Sub_%d',str2double(nums_Pati_Seiz{n}{1,1}))) = ...
Signal(Data.(sprintf('Sub_%d',str2double(nums_Pati_Seiz{n}{1,1}))).data,
window, overlap, Wwindow, Woverlap, fs, Hz, freqs);
```

את חישוב ה-PCA התחלנו בחילוץ מטריצת המאפיינים מתוך המבנה למערך. לאחר מכן, שמרנו במבנה את ממוצעי המאפיינים והפחתנו אותם בערכי המאפיינים עצמם על מנת לנרמל. אחר כך יצרנו את מטריצת הקוואריאנס על ידי הכפלת מטריצת המאפיינים המנורמלת בשחלוף (טרנספוז) שלה וחילוק במספר החלונות פחות אחד – כפי הנלמד בשיעור. אז, באמצעות הפונקציה eig, המקבלת את מטריצת הקוואריאנס והמספר 3 (אחת התכונות של הפונקציה היא קבלת מספר הקובע כמה ערכים עצמיים (הגבוהים ביותר) ווקטורים עצמיים המתאימים לאותם ערכים עצמיים יוחזרו מהפונקציה), קיבלנו את הערכים העצמיים המתאימים (מסודרים לפי גודל) ואת הוקטורים

העצמיים המתאימים להם (מסודרים בהתאם). את הוקטורים העצמיים כפלו במטריצת המאפיינים בצורה כזו שכל המאפיינים של כל חלון ייוצגו על ידי שלושה מספרים (נוצרה מטריצה בגודל 3×299 – שלושה מספרים עבור 299 חלונות), המהווים את ההיטל בעל השונות הגבוהה ביותר של אותם מאפיינים, כך שנשמר את המידע הרב ביותר שביכולתנו לשמר עם דחיסת המידע והפחתת הממדים.

```
subjMat = All_Data.(sprintf('Sub_%d',str2double(nums_Pati_Seiz{n}{1,1})));
meanSubj.(sprintf('Sub_%d',str2double(nums_Pati_Seiz{n}{1,1}))) =
    mean(subjMat,2);
subjMat = subjMat -
    meanSubj.(sprintf('Sub_%d',str2double(nums_Pati_Seiz{n}{1,1})));
covMat{n} = (subjMat * subjMat')/(size(subjMat,2) - 1);
[eigVec{n}, eigVal{n}] = eigs(covMat{n}, 3);
VisualFeatures = eigVec{n}'*subjMat;
```

קצת על הבחירה בין הפונקציות eig ו-eigs. למעשה הן דומות כמעט לחלוטין, שתיהן מקבלות מטריצה ריבועית ומחזירות ע"ע ו/או ו"ע – תלוי בכמות הפלטים שמייעדים לפונקציה. עם זאת, ישנם הבדלים קטנים. בעוד שהפונקציה eig מקבלת מטריצה ריבועית אחת (או שניים) ומחזירה את כל הע"ע והו"ע של אותה מטריצה, הפונקציה eigs (ללא הגדרה נוספת) מחזירה רק את **ששת** הע"ע הגדולים ביותר, מסודרים לפי גודלם, ואת הו"ע המתאימים להם, מסודרים בהתאם. אם מוסיפים לפונקציה eigs מספר K, בנוסף למטריצה הריבועית, הפונקציה תחזיר את K הע"ע הגדולים ביותר ואת הו"ע המתאימים להם. מאחר והפונקציה eigs מקצרת ומייעלת את זמן העבודה, בחרנו להשתמש בפונקציה זו.

בסוף יצרנו חלון תצוגה בעבור כל התקף (נבדק) כך שכותרת החלון תהיה מספר הנבדק ומספר ההתקף שלו. את וקטור הזמן (מאה דקות לפני ההתקף ועד להתקף עצמו) – שישמש אותנו גם כוקטור הצבע בפונקציה scatter ו-scatter3 – יצרנו באמצעות הפונקציה linspace אשר מקבלת שלושה ערכים ומחזירה וקטור אשר ערכיו מתחילים בערך הקלט הראשון, נגמרים בערך הקלט השני, ומספר האיברים שבו הוא ערך הקלט השלישי. את צבעי הגרפים שינינו באמצעות פונקציית colormap לתצורת hot. השתמשנו בפונקציות scatter ו-scatter3 על מנת ליצור פיזור של הנקודות הדו-ממדיות והתלת-ממדיות המבטאות את ההיטל בעל השונות הגדולה ביותר של 342 המאפיינים שאספנו על שני או שלושה ממדים. את מד הצבע יצרנו על ידי הפונקציה colorbar ואת התיג שלה יצרנו על ידי התכונה של ה-handle שלה Label.String. בגודל הגופן שלטנו בעזרת התכונה 'fontSize{%d} text'.

```
figure('units', 'normalized', 'Position', [0 0.2 1 0.6]); hold on;
sgtitle(['Patient num. - ' nums_Pati_Seiz{n}{1,1} ', Seizure num. - '
nums_Pati_Seiz{n}{1,2}], 'FontSize', 18);
```

```

c = linspace(-100,0,length(VisualFeatures(1,:)));
colormap(hot);
subplot(1,2,1); scatter(VisualFeatures(1,:),VisualFeatures(2,:), 20, c,
'filled');
title('2D projection', 'FontSize', 16);
xlabel('PC 1');
ylabel('PC 2');
grid ON;
subplot(1,2,2); scatter3(VisualFeatures(1,:),VisualFeatures(2,:),
VisualFeatures(3,:), 20, c, 'filled');
title('3D projection', 'FontSize', 16);
xlabel('PC 1');
ylabel('PC 2');
zlabel('PC 3');
c = colorbar;
c.Label.String = '\fontsize{15} Time to seizure [min]';

```

הפונקציה Signal – מקבלת מערך נתונים של סיגנל EEG כאשר כל שורה מייצגת קלט מאלקטרודה אחרת. בנוסף הפונקציה מקבלת אורך חלון רצוי לחלוקת הסיגנל לסגמנטים, אורך חפיפה רצוי של החלוקה לסגמנטים, אורך חלון רצוי לבדיקת עוצמת התדרים בשיטת וולש, אורך חפיפה רצוי לבדיקת עוצמת התדרים, תדר דגימה, טווח התדרים הרצוי ומבנה המכיל את שמות רצועות התדרים הרצויות וטווחם. בתמורה היא מחזירה מערך דו-ממדי המכיל את מספר האלקטרודות כפול מספר רצועות התדרים כפול שתיים ועוד 6 (בניסוי שלנו, $19 \times (6 + 2 \times 6) = 342$) מאפיינים עבור כל חלון זמן שמתקבל מחלוקת הסיגנל לסגמנטים.

לאחר שיצרנו משתנה All_Data ריק, שיכיל את המערך אשר יחזור מהפונקציה, פתחנו לולאה בעבור כל אלקטרודה. דבר ראשון, חילקנו את הסיגנל לסגמנטים באמצעות פונקציית buffer, אשר מקבלת את הסיגנל, את אורכי החלון והחפיפה הרצויים. באמצעות התכונה nodelay אשר גורמת ל zero padding להיות בחלון האחרון בלבד ובאמצעות כתיבת שני פלטים (השני כתוב כטילדה מפני שאנו לא משתמשים בו) אנו מקבלים את כל החלונות לתוך המשתנה הראשון (PowerSpectra) למעט החלון האחרון שנכנס למשתנה השני (אותו חלון מרופד באפסים). לאחר מכן, באמצעות הפונקציה pwelch קיבלנו את המקדמים עבור טווח התדרים אותו הפונקציה קיבלה בעבור כל חלון זמן (סגמנט).

פונקציית Signal:

```

[PowerSpectra, ~] = buffer(Data(n,:), window, overlap, 'nodelay');

```

```
PowerSpectra = pwelch(PowerSpectra, wind, op , Hz, fs);
```

לאחר מכן יצרנו את העוצמה המנורמלת באמצעות הפונקציה normPower שמקבלת את מטריצת מקדמי התדרים לכל חלון זמן ובאמצעות לולאה עוברת על כל אחת מהשורות ומחלקת אותה בסכום העמודות.

פונקציית Signal:

```
NormalizedPower = normPower(PowerSpectra);
```

פונקציה:

```
function NormalizedPower = normPower(Data)
NormalizedPower = zeros(size(Data));
for n = 1:length(Data(:,1))
    NormalizedPower(n,:) = Data(n,:)./sum(Data,1);
end
end
```

שני המאפיינים הראשונים (כאשר למעשה כל מאפיין בעצם מכיל שישה מאפיינים, אחד בעבור כל רצועת תדר. סה"כ 12 מאפיינים) אותם יצרנו הם העוצמה היחסית והעוצמה היחסית לוגריתמית של כל רצועת תדרים באמצעות הפונקציה CalcRelative, אשר מקבלת את מטריצת מקדמי התדרים לכל חלון זמן, את וקטור טווח התדרים ואת המבנה המכיל את שמות וטווחי רצועות התדרים. קודם כל חילצנו את שמות רצועות התדרים באמצעות פונקציית fieldnames ושמרנו את מספר הרצועות. לאחר מכן יצרנו משתנים ריקים שיכילו את מטריצות העוצמה היחסיות ואת לוגריתם העוצמה הכללית. השתמשנו בנוסחה שהובאה במטלה על מנת לבטא את סכום הלוגריתם של עוצמות התדרים בכל חלון זמן. צורת הנוסחה מאפשרת לנו לדעת כי לא נכנס לתחום המספרים השליליים (יוסבר בסוף הפסקה). בשלב זה פתחנו לולאה בעבור כל אחת מרצועות התדרים. חילצנו את טווח התדר של כל רצועה ואת העוצמות של כל רצועה על ידי שימוש בפקודה find על השוואת וקטור טווח התדרים הכללי לערכי טווח תדרי הרצועות. יצרנו משתני מונה (numerator - סכום רצועות התדרים לכל חלון שמן) ומכנה (denominator - סכום העוצמה הכללית של כל חלון זמן). חלוקה ביניהן מביאה לנו את העוצמה היחסית של כל רצועת תדרים וחלוקת סכום הנוסחה הלוגריתמית של רצועת התדרים בסכום הנוסחה הלוגריתמית של כל התדר מביאה לנו את העוצמה הלוגריתמית היחסית.

$$\text{Log}(\exp(1) * \text{power} ./ \min(\text{power})) = \log(\exp(1)) + \log(\text{power} ./ \min(\text{power}))$$

לוגריתם של אקספוננט שווה לאחד, והמספר הכי נמוך שיופיע במונה הוא המספר שמופיע במכנה מה שגורם לכך שהמספר הכי נמוך שיופיע בלוגריתם השני הוא 1 שתוצאתו תהיה שווה ל-0. לכן, הערך הנמוך ביותר שאנו מקבלים מהמשוואה הזו הוא 1. חשוב שלא יהיו לנו תוצאות שליליות מפני שאנו מדברים על עוצמות תדרים והיחס ביניהם, וערכים אלו אינם יכולים להיות שליליים על כן חשוב שהתוצאות לא יצאו שליליות.

פונקציית Signal:

```
[RelativePower, RelativeLogPower] = CalcRelative(PowerSpectra, Hz, freqs);
```

פונקציה:

```
function [RelativePower, RelativeLogPower] = CalcRelative(Data, Hz, freqs)

fields = fieldnames(freqs);
numFreq = size(fields,1);

RelativePower = zeros(numFreq,size(Data,2));
RelativeLogPower = zeros(numFreq,size(Data,2));

LogTotPow = sum(log(exp(1).*Data./min(Data)));

for n = 1:numFreq
temp = freqs.(fields{n});
tempData = Data(find(Hz == temp(1)):find(Hz == temp(end)),:);

numerator = sum(tempData, 1);
denominator = sum(Data, 1);

RelativePower(n,:) = numerator./denominator;
RelativeLogPower(n,:) =sum(log(exp(1).*tempData./min(tempData)))./LogTotPow;
end
end
```

את המאפיין הבא, השלושה עשר במספר, סטיית התקן של עוצמת התדרים, חישבנו באמצעות שורש הסכום של עוצמת התדרים לכל חלון זמן.

פונקציית Signal:

```
stdPower = sqrt(sum(PowerSpectra));
```

שני המאפיינים הבאים חושבו באמצעות פונקציית `calcCoeff` שמקבלת את מטריצת מקדמי התדרים לכל חלון זמן, לקטור טווח התדרים ומספר האלקטרודה (האחרון בשימוש כשרוצים להציג את פעולת כל אלקטרודה בעבור כל נבדק, סה"כ 76 גרפים (4 נבדקים כפול 19 אלקטרודות)). לאחר שהפונקציה יוצרת מערך דו ממדי בעל שתי שורות ומספר עמודות כמספר חלונות הזמן, באמצעות לולאה (שרצה על חלונות הזמן) והשימוש בפונקציית `polyfit` (המקבלת את לוגריתם וקטור טווח התדרים, לוגריתם העוצמה בכל חלון והספרה אחד – המייצגת את סדר הפולינום הרצוי) אנו מקבלים את שני המקדמים הלינאריים של כל לוגריתם העוצמה כפונקציה של לוגריתם התדר. (הגרפים מסומנים ב-comment, על מנת להציג את הגרפים יש לעשות uncomment).

סקריפט Signal:

```
coeff = calcCoeff(PowerSpectra, Hz, n);
```

פונקציה:

```
function coeff = calcCoeff(Data, Hz, a)
coeff = zeros(2, size(Data, 2));
for n = 1:size(Data, 2)
    coeff(:,n) = polyfit(log(Hz), log(Data(:,n)), 1)';
end
end
```

את המאפיין ה-16, `spectralMoment`, יצרנו על ידי מכפלה וקטורית בין וקטור טווח התדרים לבין מטריצת העוצמה המנורמלת.

סקריפט Signal:

```
spectralMoment = Hz*NormalizedPower;
```

שני המאפיינים האחרונים, `spectralEdge` ו-`spectralEntropy` חושבו באמצעות פונקציות שנכתבו עבורן, `specEdge` ו-`specEntropy`, כאשר הקלט היחיד הינו מטריצת העוצמה המנורמלת. `specEdge` מוצאת את התדר שעד אליו סכימת העוצמה המנורמלת שווה ל-0.9. במקרה שלנו הערך הכי קרוב ל-0.9 מלמטה (0.8...). הפונקציה מחשבת זאת תחילה בעזרת הפקודה `cumsum` שמחשבת את הסכום המצטבר של כל עמודה (חלון זמן). לאחר מכן הפכנו את וקטור הסכום המצטבר לוקטור בינארי כאשר הסף הוא 0.9. בעזרת הפונקציה `diff` יצרנו וקטור הפרשים ומצאנו את באמצעות `find` והשמת שני פלטים את כל מיקומי השורות (= התדרים) שבהם יש הפרש בין הערכים בוקטור הבינארי. כלומר, תדר הסף. `specEntropy` מבצע סכימה של כל עוצמה של כל תדר בכל חלון זמן

307963157
311603476

כפול הלוג על בסיס 2 של אותה העוצמה. לשם כך נעזרנו בלולאה ובפקודה \log_2 שמחשבת לוג על בסיס 2. לבסוף עשינו מינוס על תוצר הסכימה.

סקריפט Signal:

```
spectralEdge = specEdge(NormalizedPower);
```

פונקציה:

```
function [spectralEdge] = specEdge(Data)
sumIs = cumsum(Data, 1);
sumIs = sumIs < 0.9;
diffSum = diff(sumIs);
[spectralEdge, ~] = find(diffSum == -1);
spectralEdge = spectralEdge';
end
```

סקריפט Signal:

```
spectralEntropy = specEntropy(NormalizedPower);
```

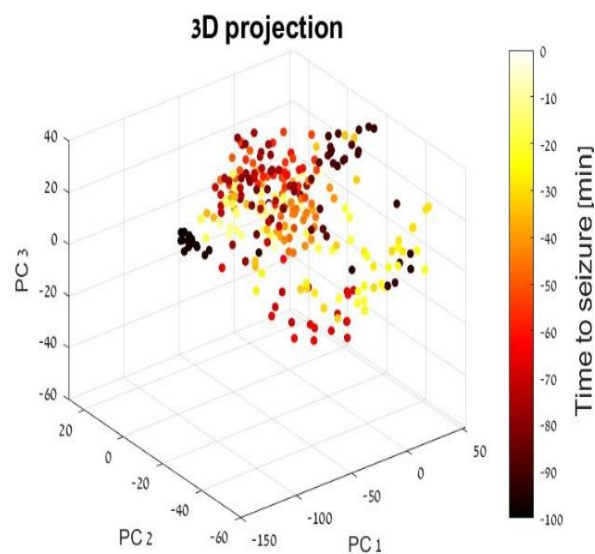
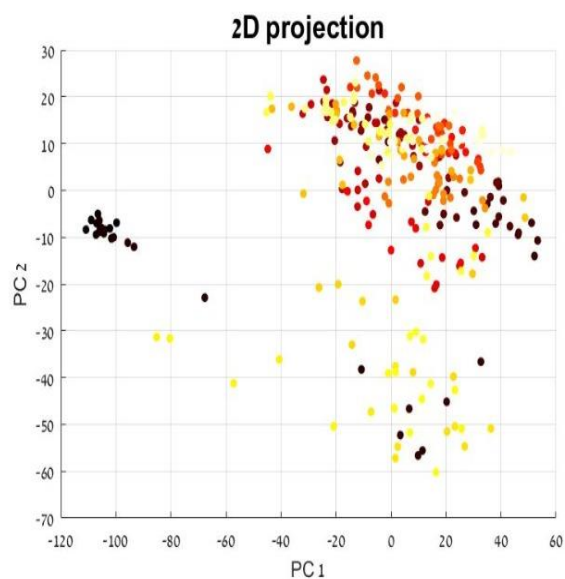
פונקציה:

```
function spectralEntropy = specEntropy(Data)

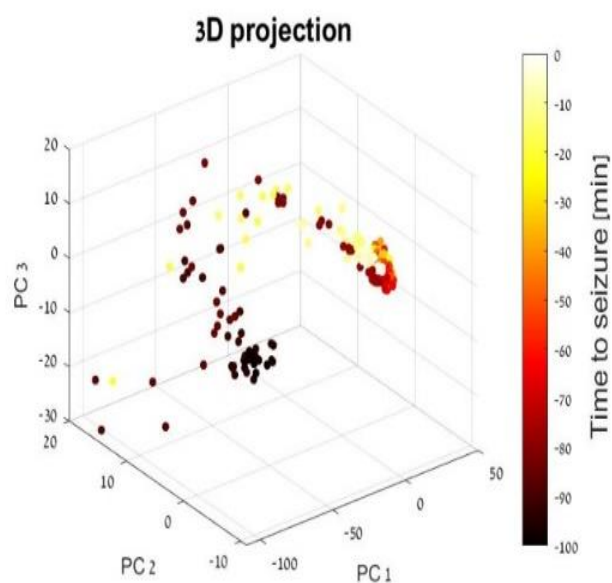
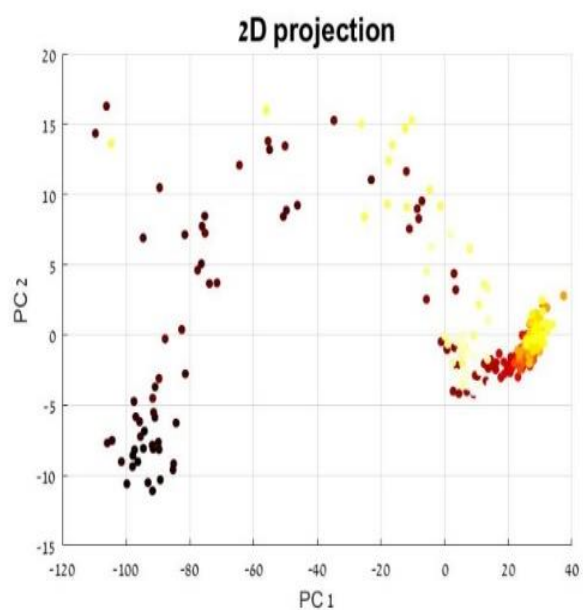
spectralEntropy = Data(1,:) .* log2(Data(1,:));
for n = 2:size(Data,1)
    spectralEntropy = spectralEntropy + Data(n,:) .* log2(Data(n,:));
end
spectralEntropy = -spectralEntropy;
end
```

לאחר כל החישובים הנ"ל, אנו שומרים את כל המאפיינים כשורות במערך All_Data שלאחר ריצה על כל האלקטרודות, יחזור לסקריפט הראשי.

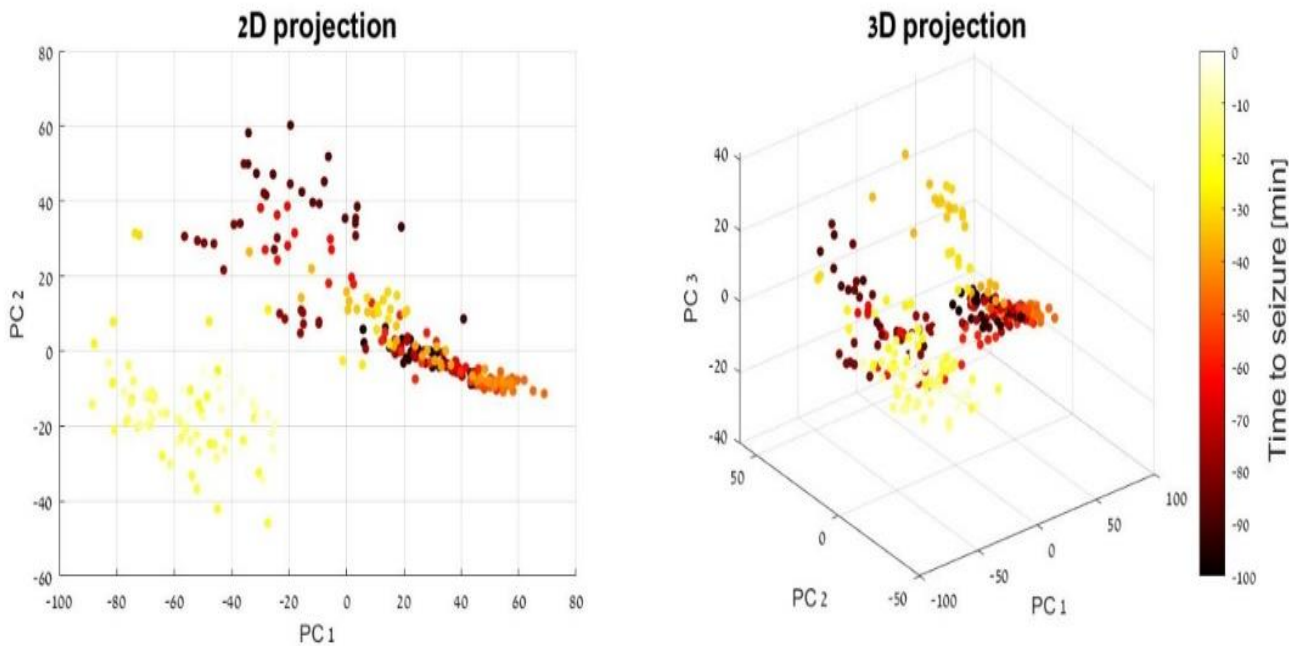
Patient num. - 62, Seizure num. - 3



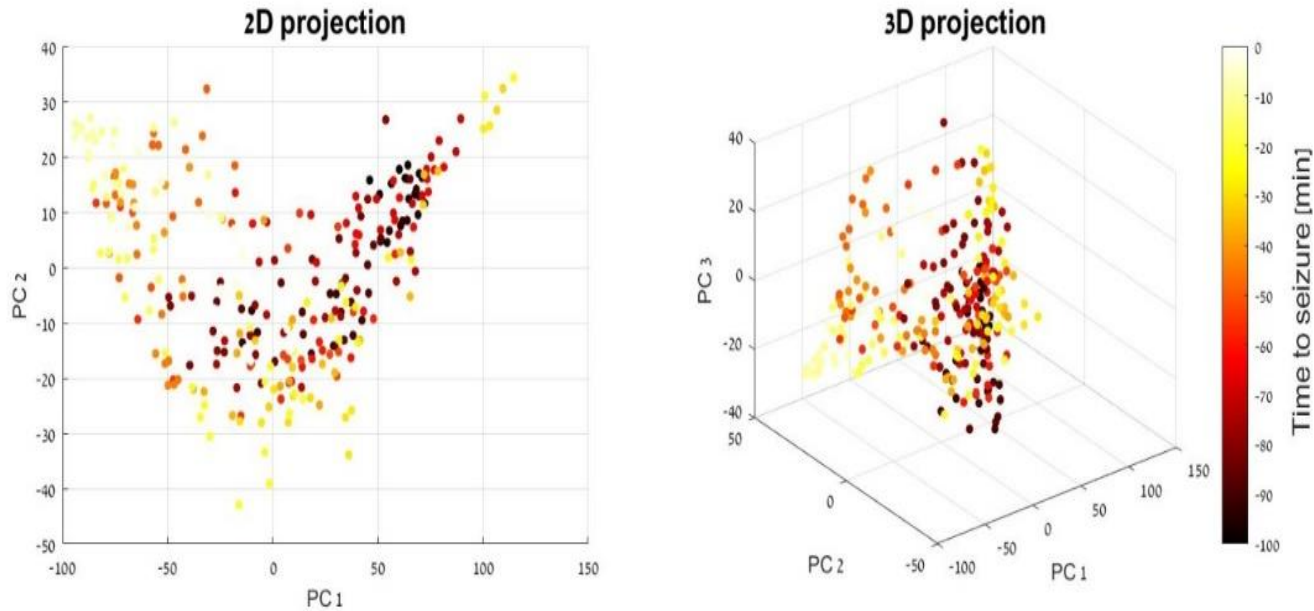
Patient num. - 67, Seizure num. - 3



Patient num. - 44, Seizure num. - 3



Patient num. - 78, Seizure num. - 1



בכל חלון תצוגה ניתן לראות 2 גרפים המייצגים מפה דו מימדית ותלת מימדית של ארבעת הנבדקים כך שכל נקודה מסמלת את היטל מאפייני הפעילות המוחית בנקודות זמן שונות. מימד הזמן מיוצג באמצעות צבע ההולך ומתבהר, מ100 דקות לפני ועד זמן ההתקף.

ניתן לראות הבדל בדפוס הפעילות המוחית בטרם ההתקף אצל נבדקים מספר 67 ו44. בקרב נבדק 44, דפוס הפעילות המוחית המאפיין את 100 דקות לפני ההתקף ועד ל20 דקות לפני ההתקף נשאר בטווח החיובי של הצירים, וכ20 דקות לפני ההתקף דפוס הפעילות משתנה והיטלה נמצאת בטווח השלילי.

בניגוד לכך, בנבדק מספר 67 דפוס הפעילות המוחית המאפיין את 100 דקות לפני ההתקף ועד ל50 דקות לפני ההתקף נשאר בטווח השלילי של הצירים, וכ50 דקות לפני ההתקף דפוס הפעילות משתנה והיטלה נמצאת בטווח החיובי של PC1. לפי הבדלים אלו, ניתן לנבא מדפוס הפעילות השונה של אותו אינדיווידואל האם הוא עשוי לחוות התקף אפילפסיה טרם ההתקף.

בשני הנבדקים האחרים - מספר 62 ו78, לא ניתן לראות דפוס פעילות מוחית מסוימת המאפיינת את הזמנים השונים בטרם ההתקף, ועל כן לא ניתן לנבא מדפוס הפעילות המוחית שלהם דבר. ניתן לראות שאין דפוס פעילות מוחית מסוימת משותפת בין הנבדקים המאפיינת את הלוקים באפילפסיה בטרם ההתקף.