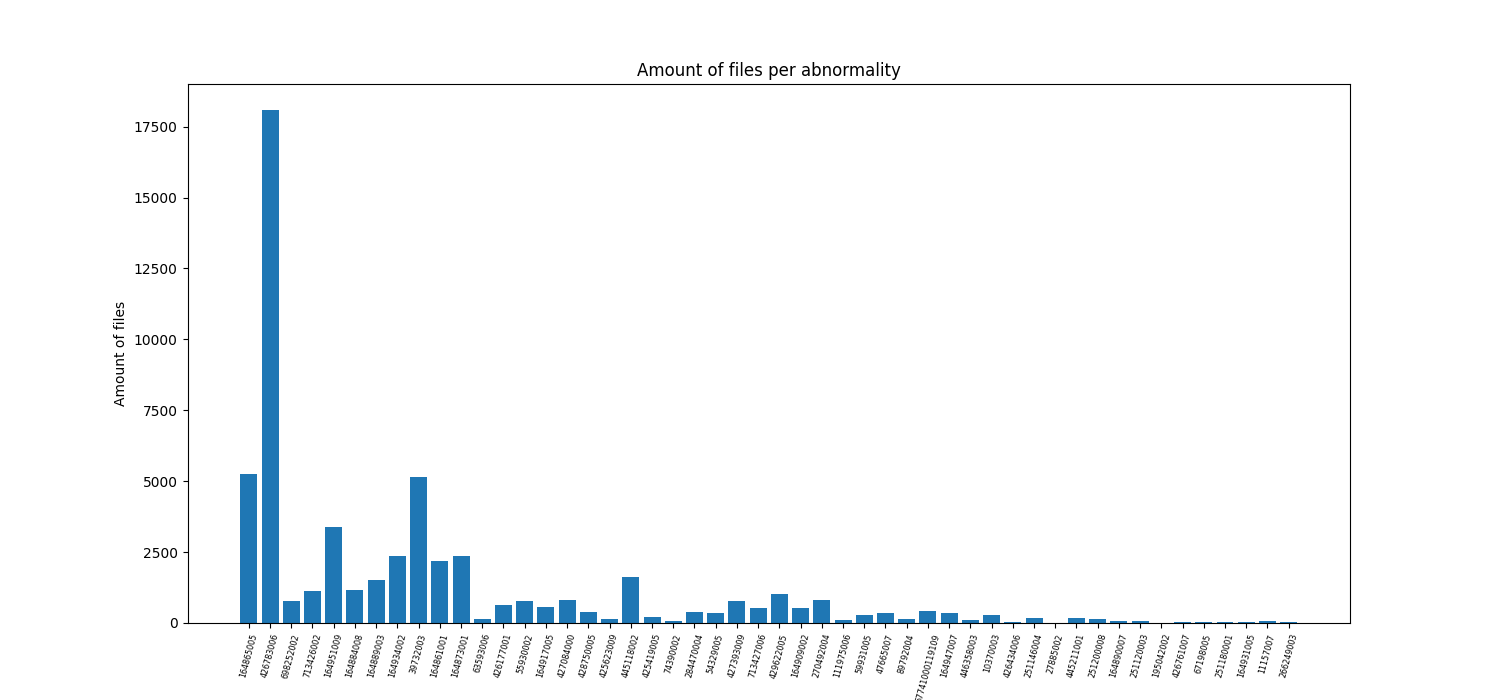
**פרויקט סיום הולכה חשמלית בתאים**

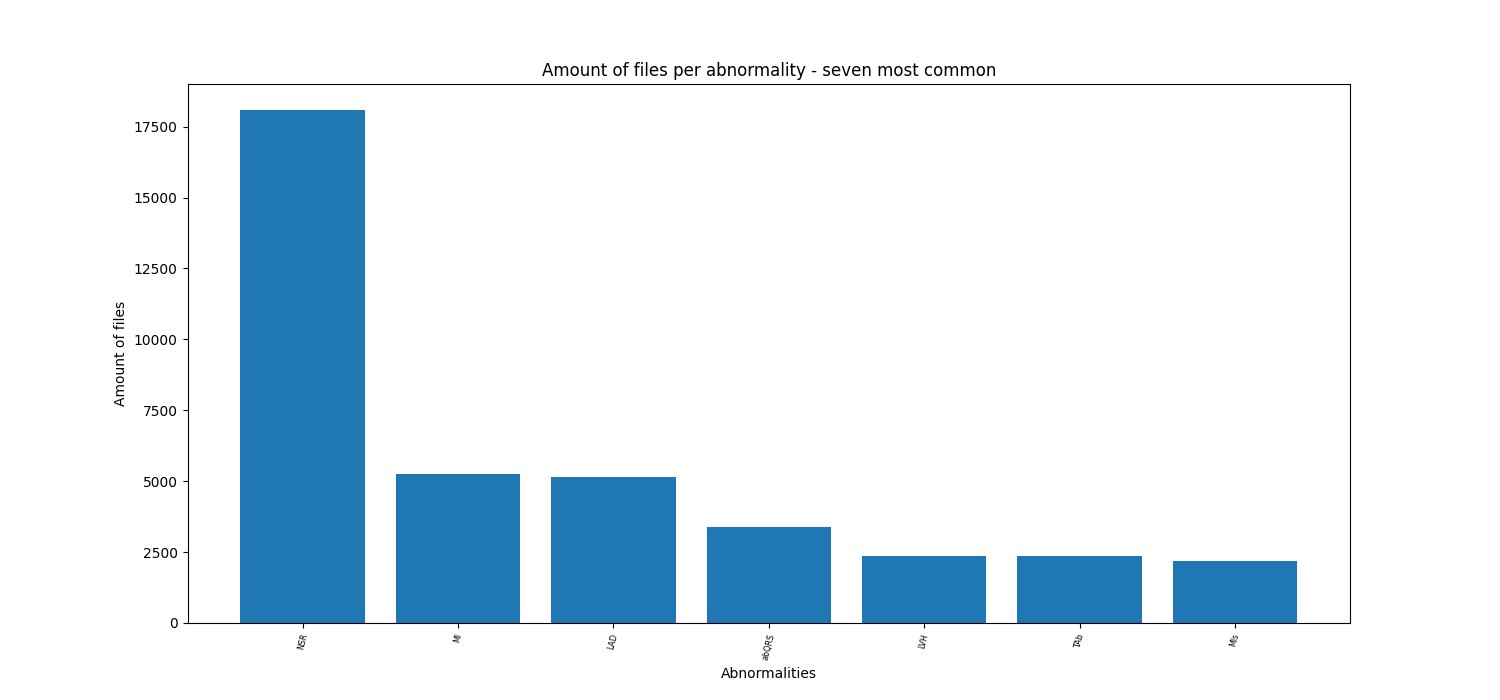
דור מסאס 318218625, אברהם חרינביצקי 313133324

**חלוקת דאטה:**

1. תחילה כתבנו קוד שעובר על כל קבצי ה-.hea הנתונים ומוצא את שבעת האבחונים הכי נפוצים – אדם בריא ו-6 המחלות הנפוצות ביותר.
2. לאחר שיש לנו את המידע נחלק את הנתונים לתיקיות train, validation, test לפי יחס 70-15-15. בשלב זה נרצה לשמור על פילוג אחיד של המידע, לכן נחלק כל אחד מהקבצים באותו פילוג לכל תיקייה, תוך כדי שמירה על כמות מספקת של נתונים עבור על אבחון ומבלי ליצור bias.



איור 1 - התפלגות האבחונים בקבצים השונים



איור 2 - התפלגות 7 האבחונים הנפוצים

**טעינת דאטה:**

1. לאחר שהסיגנלים הנתונים חולקו לתיקיות, נרצה לטעון אותן אל תוך המודל. ניעזר בחבילה wfdb המאפשרת קריאה של הסיגנלים וקריאת האבחונים והפרמטרים הנתונים בקובץ ה-hea.
2. מאחר ויכולים להיות מספר אבחונים לכל סיגנל נתייג את הדאטה באמצעות תגית one-hot (או כמה, בהתאם לאבחון הסיגנל), כלומר מילון/וקטור המייצג את האבחונים המתאימים לסיגנל, כך שהרשת תוכל להתאמן תוך התייחסות לכל התסמינים המתאימים במקביל.
3. את הדאטה נקרא לתוך מילון כאשר מספר האבחון הוא המפתח, והאיברים עבור כל מפתח הם שמות הקבצים המכילים את אותה הפרעה, כאשר אותו קובץ יכול להופיע תחת מספר מפתחות עבור ריבוי הפרעות. על מנת למנוע הטייה של הלמידה נגביל את מספר הקבצים הקיימים מכל אבחון ל-6000 קבצים לכל היותר.
4. מתוך המילון נבחר את שבע ההפרעות השכיחות ביותר, כאשר הנפוצה תהיה קצב לב רגיל – NSR, והשאר יהיו בסדר הבא - 'MI', 'LAD', 'abQRS', 'LVH', 'TAb', 'MIs'.
5. לאחר שקיבלנו מילון שמכיל את ה7 מחלות הנפוצות עברנו על הדאטה שבתוך המילון ופילטרנו אותו בכדי להעיף רעשים ואותות DC מהסיגנל המקורי, בנוסף לזה נירמלנו את ערכי הסיגנל המפולטר כך שיהיו בין -1 ל1.

**מודל**

בחרנו לבנות ארכיטקטורת מודל משלנו המבוססת על רשת נוירונים מסוגCNN - הרשת מכילה שכבות של קונבולוציה חד מימדית אשר מלוות בשכבת אקטיבציה מסוג Relu וMaxpool וכן שכבות fully connected.

ניתן לראות שבכל שכבות הקונבולוציה ישנו שימוש ב padding על מנת לשמור על אורך סיגנל אחיד – 5000 ערכים המתאימים לחצי דקה של סיגנל.

אל המודל נכנס וקטור בגודל 32x5000x12 כאשר 32 מייצג את גודל ה-batch שאיתו בחרנו לאמן את המודל כאשר batch הינו מספר ה – inputs שנכנסים למודל בכל קריאה לאימון( על כמה קבצים בו זמנית המודל מתאמן), 5000 הוא אורך הסיגנל, ו-12 ערוצים כמספר האלקטרודות.

חשוב להדגיש כי בחרנו בקונבולוציה חד מימדית על 12 ערוצים ולא קונבולוציה דו מימדית, שכן הסיגנלים הנמדדים מאלקטרודות שונות אינם זהים בצורתם ולכן עשויים לגרום ללמידה שגויה על יחס בין האלקטרודות השונות לעומת אימון ערוצים המתאימים לכל אלקטרודה בנפרד.

הבעיה איתה התמודדנו הינה בעיית - Multi label classification אבחון כאשר האבחנה יכולה לכלול יותר מקטגוריה אחת, כלומר הפלט יכול להיות שייך לכמה קטגוריות, ובמקרה שלנו חולה אחד יכול להיות בעל מספק אבחונים. לכן על מנת לאמן את המודל נבחרה פונקציית BCEWithLogitsLoss כפונקציית חישוב ההפסד שלנו שכן פונקציה זו משלבת את פונקציית ה -

BCE (Binary cross entropy) אשר משתמשים בה בבעיות multiclass classification, עם Sigmoid במוצא על מנת להתאים לבעיות של Multi label classification.

משום כך, בנוסף לשכבות המוצגות באיור 1 הצטרפה שכבת אקטיבציה מסוג Sigmoid על מנת לנרמל את ערכי המוצא כך שיהיו בין 0 ל 1 בשלבים שלאחר האימון, שכן באימון ה-Sigmoid ב-BCEWithLogitsLoss.

חיזוי הפלט נקבע בעזרת סף של חצי – ערכים שיצאו אחרי ה Sigmoid כך שהם גדולים מחצי הפכו בוקטור החיזוי ל-1 וערכים קטנים מחצי התאפסו.

**אימון המודל**

אימון המודל נעשה ב epochs (כל epoch הינו מעבר על כל הדאטה של האימון), כמות הepochs שהתחלנו איתה את האימון הייתה 50 וכן הגדרנו את הlearning rate(קצב הלמידה של המודל) להיות 0.001 ושכל 20 epochs יקטן פי 10, דבר זה נועד כדי למנוע המודל להגיע למצב שבו הוא נתקע במינימום מקומי בפונקציית חישוב הפסד ויתקע, לאחר שראינו שהאימון מתבצע בצורה טובה הגדלנו את הכמות ל100.

הכנס גרפים של הפסד

כמו שניתן לראות מהאיור אכן קיבלנו התכנסות להפסד נמוך יותר בסט הולידציה כאשר הגלנו את כמות ה epochs ל100.

לאחר האימון כאשר עברנו לטסט ראינו שאנחנו מקבלים confusion matrix שמכיל ערכים גבוהים בעיקר על דאטה של אנשים בריאים אבל עבור מחלות מסוימות ניתן לראות מהמטריצה שאופן הזיהוי אינו טוב, דבר זה נובע מכך שמחלות אלו כמעט ולא נמצאות בסט הדאטה שקיבלנו ולכן המודל אינו אומן מספיק טוב כדי לזהות אותם.

הכנס confusion matrix

כדי לנסות לפתור בעיה זו הגבלנו את הכמות קבצים שיש להם label של אנשים בריאים שהמודל יכול לקבל כך שלא יהיה כל כך הרבה דאטה של אנשים בריאים, הדבר אכן עזר למודל לפענח בצורה טובה יותר את המחלות אך גרם לכך שהוא ידע לאבחן פחות טוב אנשים בריאים.

הכנס confusion matrix לאחר תיקון הכמות הדאטה

לאחר שראינו שפתרון זה אינו אופטימלי החלטנו ללכת על פתרון אחר, נעזרנו בהזזה של הדאטה של אותם מחלות שהמודל אינו יודע לזהות טוב על מנת לשכפל את מספר הקבצים שמכילים את הlabels של אותם מחלות ובכך לאמן את המודל שקיבלנו על יותר קבצים של מחלות אלו.

הכנס גרפים של הפסד על הדאטה החדש

הכנס confusion matrix לאחר תיקון הכמות הדאטה

**קלט**

המודל מקבל כמה ארגומנטים מהמשתמש על מנת להתחיל את הריצה :

Input – הpath של התיקיית דאטה שנכנסת למודל כinput הdefault – “WFDB”

output – הpath של התיקייה שאליה יודפסו כל הפלטים של הריצה הdefault – “” (ריק – הדפסה באותה תיקייה שבה הקובץ ריצה)

phase – השלב של הריצה האם זה training, test או fine tune הdefault – “training”

epochs – במקרה שאנחנו בphase של אימון על כמה epochs הוא מתבצע הdefault – 50

Model-path – במקרה שאנחנו עושים test או fine tune ונרצה לבדוק רשת מאומנת או לאמן רשת מאומנת אזי אנחנו צריכים להכניס את הpath של אותו קובץ של המודל כדי שהוא יוכל לטעון אותה

**פלט**

תיקיית הפלט של הריצה תכיל קובץ בשם: run\_log\_{date and time of the run}.log שמכיל את כל מהלך הריצה, במקרה של ריצה רק של ‘test’ הקובץ יכיל את הארגומנטים שנכנסו לפונקציה מהמשתמש, את הconfusion matrix שהתקבל במהלך הריצה וכן את ה Test Loss ו Test accuracy.

בנוסף תיקייה זו מכילה קובץ בשם: cm\_{date and time of the run}.png שמכיל ויזואליזציה של confusion matrix של אותה הריצה.



איור 3 - דוגמא לתיקיית פלט