Epileptic Seizure Prediction using EEG Time Series Data

For the partial fulfilment of the course CS F266

Under the able guidance of
Dr Basabdatta Sen Bhattacharya

By -
Fenil Suchak (2016B3A70541G)

Acknowledgement

I would like to express my special thanks of gratitude to my supervisor Dr.Basabdatta Sen Bhattacharya who assisted this project on a weekly basis giving feedbacks and organizing poster presentation. I would also like to thank Dr Levin Kuhlmann and Epilepsyecosystem.org for providing me with the invaluable dataset and ideas without which this project would not have been possible. As this work builds on the work of several other researchers whose research papers have been studied, I'm very grateful to the community for sustaining such a free flow of information. Lastly, I would like to thank the open source community for building fast and efficient tools for scientific computing as most of the operations performed in this study use open source implementations.

Code is open source and available at Code available at -
https://github.com/Fenil3510/Epileptic_Seizure_Prediction

 [1]

[1] www.epilepsyecosystem.org

Abstract

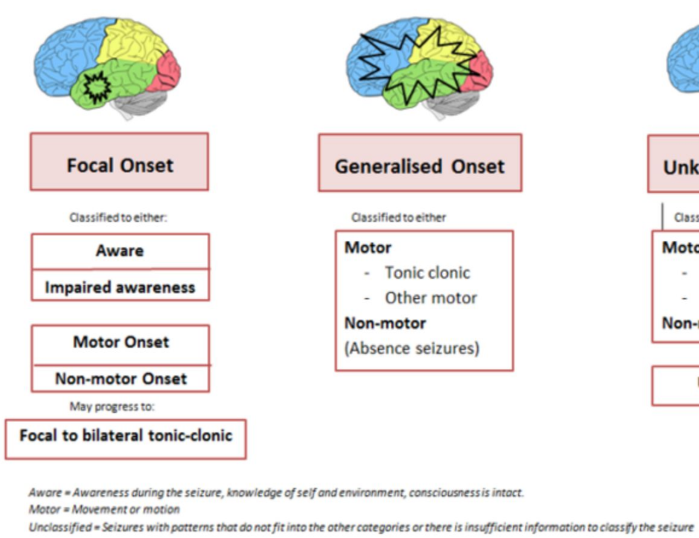
This project builds a predictive model to forecast an onsetting epileptic seizure before 5 minutes. The data used is time series EEG data with 16 channels. The two main models used for prediction are xgboost (gradient based decision tree and convolutional neural networks). This work achieves a roc-auc of 0.79 and the state of the art on this dataset being 0.81. This work is subject to further model tuning which will match or exceed the state of the art.

Introduction

Seizures, abnormal movements or behaviour due to unusual electrical activity in the brain, are a symptom of epilepsy. But not all people who appear to have seizures have epilepsy, a group of related disorders characterized by a tendency for **recurrent** seizures.

Non-Epileptic Seizure

Also called pseudoseizures are not accompanied by abnormal electrical activity in the brain and may be caused by psychological issues or stress. However, non-epileptic seizures look like true seizures, which makes diagnosis more difficult. Normal EEG readings and lack of response to epileptic drugs are two clues they are not true epileptic seizures. These types of seizure may be treated with psychotherapy and psychiatric medications.



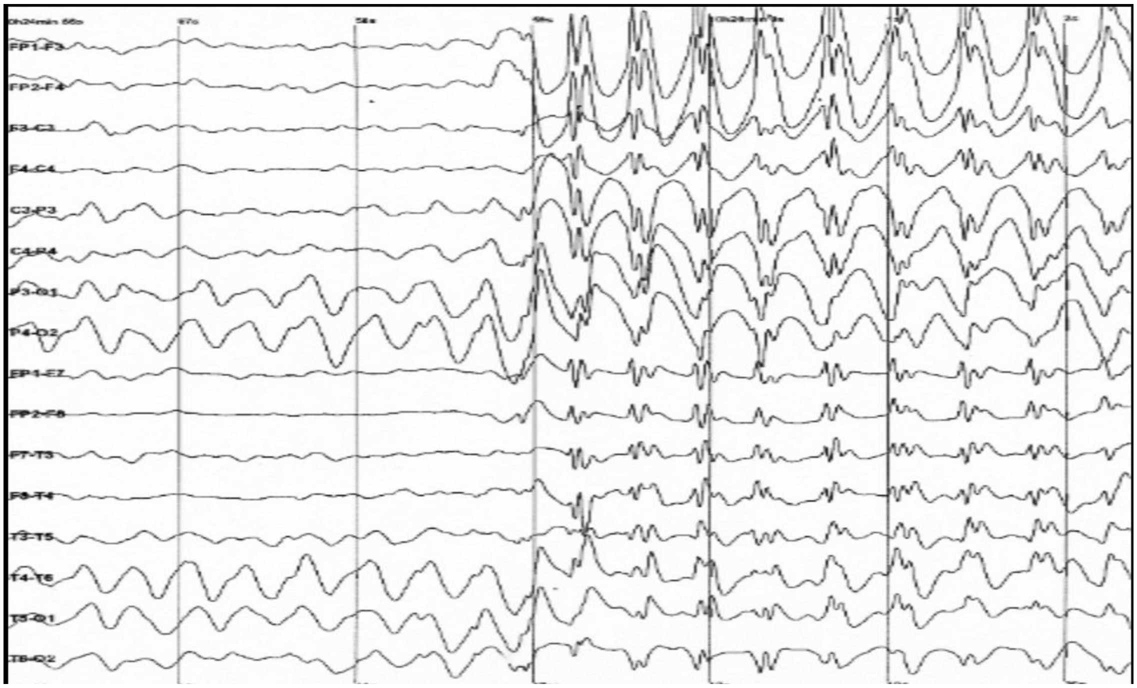
[2]

Epileptic Seizures

An epileptic seizure is a period of conditions that occur due to abnormally high or synchronous neuronal activity in the brain. The effects vary from uncontrolled shaking movements involving much of the body with loss of consciousness to shaking movements involving only part of the body with variable levels of consciousness to a

[2] https://www.epilepsy.org.au

subtle momentary loss of awareness. Most of the time these episodes last less than 2 minutes and it takes some time to return to normal. Loss of bladder control may occur.


[3]

Epilepsy afflicts nearly 1% of the world's population and is characterized by the occurrence of seizures. For many patients, medications can be given at sufficiently high doses to prevent seizures, but patients frequently suffer side effects. For 20- 40% of patients with epilepsy, medications are not effective. Even after surgical removal of epilepsy, many patients continue to experience spontaneous seizures. Despite the fact that seizures occur infrequently, patients with epilepsy experience persistent anxiety due to the possibility of a seizure occurring.

Seizures may be provoked and unprovoked. Provoked seizures are due to a temporary event such as low blood sugar, alcohol withdrawal, low blood sodium, fever, brain infection, or concussion. Unprovoked seizures occur without a known or fixable cause such that ongoing seizures are likely. Unprovoked seizures may be triggered by stress or sleep deprivation. Diseases of the brain, where there has been at least one seizure and a long term risk of further seizures, are collectively known as epilepsy. Conditions that look like epileptic seizures but are not include fainting, nonepileptic psychogenic event, and tremor.

---

[3] Epileptic Seizure cause the neuronal activity to increase hence the change captured by EEG

Literature Review in Depth

This section details all the research papers that are used as references to make features, build models. These research papers serve as a strong foundation for this study, only the most helpful research papers are summarized here, all others papers are mentioned in the reference section

Epilepsyecosystem.org: crowd-sourcing reproducible seizure prediction with long-term human intracranial EEG

Authors - Dean R Freestone, Mark J Cook, David B Grayden, Philippa Karoly, Levin Kuhlmann

Summary

In this authors used IEEG data were recorded chronically from humans with refractory focal epilepsy using the NeuroVista Seizure Advisory System implanted device. Sixteen subdural electrodes (4 4-contact strips) were implanted in each patient, targeted to the presumed seizure focus. The electrode leads were tunnelled to a subclavicularly-placed implanted telemetry unit. A rechargeable battery-powered the implanted device. Data were sampled at 400 Hz with signed 16-bit resolution and wirelessly transmitted to an external, hand-held personal advisory device. Recorded iEEG from the 16 electrode contacts were referenced to the group average across all electrode channels and continuously saved to removable flash media. The data were divided into labelled training and unlabelled testing sets for the contest. Held-out data were used for follow-up evaluation of the top algorithms after the contest had finished. To mimic practical application and avoid. To avoid signal non-stationarities in the immediate period following implantation of electrodes, the contest data came from the period between 1 and 7 months after implantation, with the remaining data used as held-out data. As this was a crowd-sourced competition on Kaggle top results and approaches are summarized in this table

Team	Features	Algorithm	Ensemble	Public LB	Private LB
Team A	Spectral Power, Fractal Dimensions, Riemannian autocorrelation	K-NN, Generalized Linear Models, Linear SVM	Ranked Average	0.85276	0.80701

Team B	Correlation, Distribution Statistics, energy Entropy	RandomForest	n/a	0.78328	0.79898
Team C	Spectral Power, Distribution Statistics, RMS of signal, spectral edge	Polynomial SVM, random under-sampling	Weighted Average	0.811624	0.79652

These approaches are openly available on GitHub and served as reference ideas for getting started. It should be noted that data segments were 10-minute segments hence to increase data points the data were further divided into smaller segments of 100s, 1-minute segments in various approaches.

Seizure Prediction Using Spike Rate of Intracranial EEG

Authors - Shufang Li, Weidong Zhou, Qi Yuan, and Yinxia Liu

Summary

An important characteristic of epileptic EEG is the manifestation of spikes, sharps, spike and slow wave complex. Spikes are transient signals with a pointed peak, clearly distinguishable from the background activity. A variety of automatic spike detection methods have been presented, such as neural network, adaptive time-frequency parameterization, multi-level wavelet, etc. In addition, some previous studies, have shown that the spike rates change significantly minutes prior to a seizure. Lange et al. found that the rate of bilateral spikes increased obviously across 20 min preictal segments. Neuronal spiking activity reflects a distinct and widely occurring physiological state in focal epilepsies, and changes in neuronal spiking activity may be detected minutes before seizure onset. The authors in this paper use Morphological Opening and Closing filtering for extracting positive spikes and morphological Closing and Opening for extracting negative spikes. A brief calculation procedure is outlined below

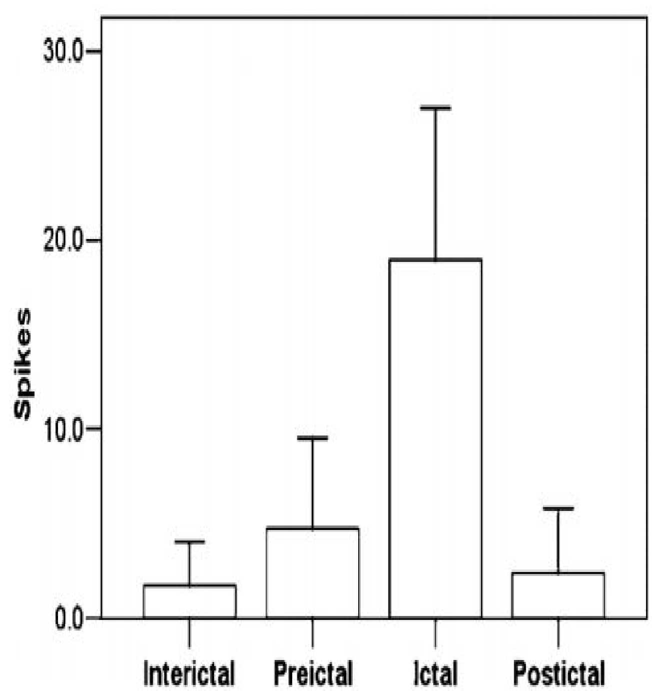
Some notations,

Morphological Closing-Opening - CO
Morphological Opening-Closing - OC

Original Waveform - x_t
Waveform after morphological Transformations - y_t
Final Spikes obtained by operation - $z_t = x_t - y_t$

The formula for calculating y_t

$$y_t = \frac{1}{2}(OC(x_t)) + CO(x_t))$$



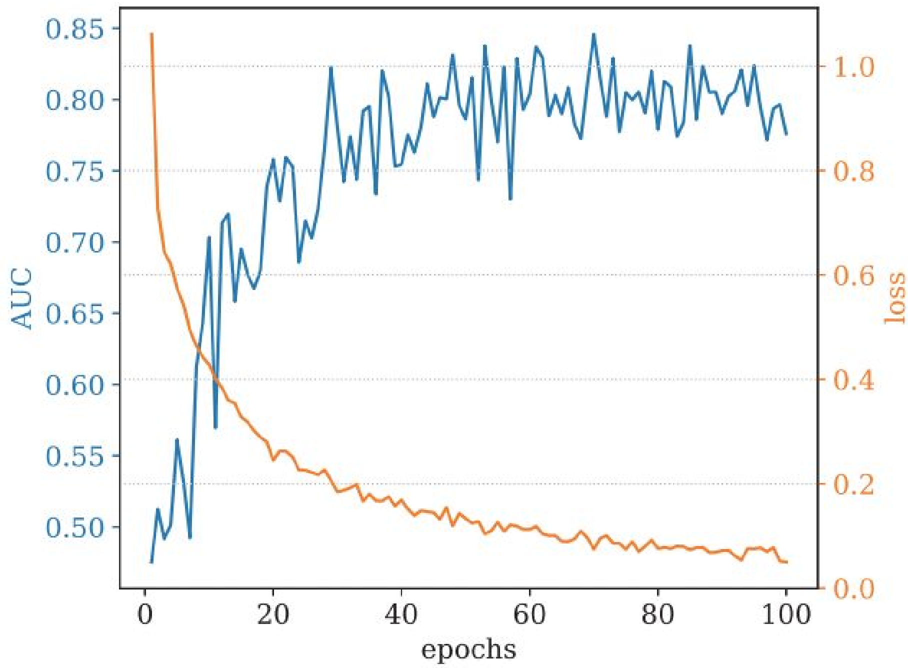
After transformation, the final spikes are summed up and divided by the time interval to get the spike rate. The paper reveals that there are significant differences in the spike rates of different data segments and can serve as a discriminating factor between the signals.

Convolutional Neural Networks for Epileptic Seizure Prediction

Authors - Mathias EberLien, Raphael Hildebrand, Nico Hoffmann, Levin Kuhlmann

Summary

In this paper, authors acquire a very novel approach of using a convolutional neural network algorithm as a feature extractor. It is very clear that a naked eye can distinguish an inter-ictal segment from a pre-ictal segment, here is where CNN's play a near accurate role in automating this procedure. Authors propose to overcome the concept of separated extraction of manually selected features and subsequent classification. By applying a CNN topology directly to the multichannel EEG time series, an appropriate representation of the data, as well as suitable models for classification, are directly derived from the data. In the proposed work, different topologies are evaluated and possible benefits from including local information about electrodes are investigated. All evaluations have been made prospectively and with the use of long-term data sets, containing data of different subjects that range over periods of multiple months. Therefore, the proposed results are actually reflecting performances in real-world applications. One of the main reasons for the popularity of CNN in other domains is their ability to learn suitable features, that are spatial invariant [24]. This corresponds to the capability of an algorithm to detect specific patterns regardless of the point of their occurrence. In order to exploit these advantages for the classification of EEG time series, no features have been extracted, but the clips of multi-channel time series were directly assigned to the input of a deep neural network. Therefore, the proposed networks learn to detect local patterns of samples, independent of the time and channel of their appearance.



[4]

These results are quite promising and will be explored as a part of this project. Different architectures are yet to be tested.

Predicting Epileptic Seizures in Advance

Authors - Negin Moghim, David W.Corne

Summary

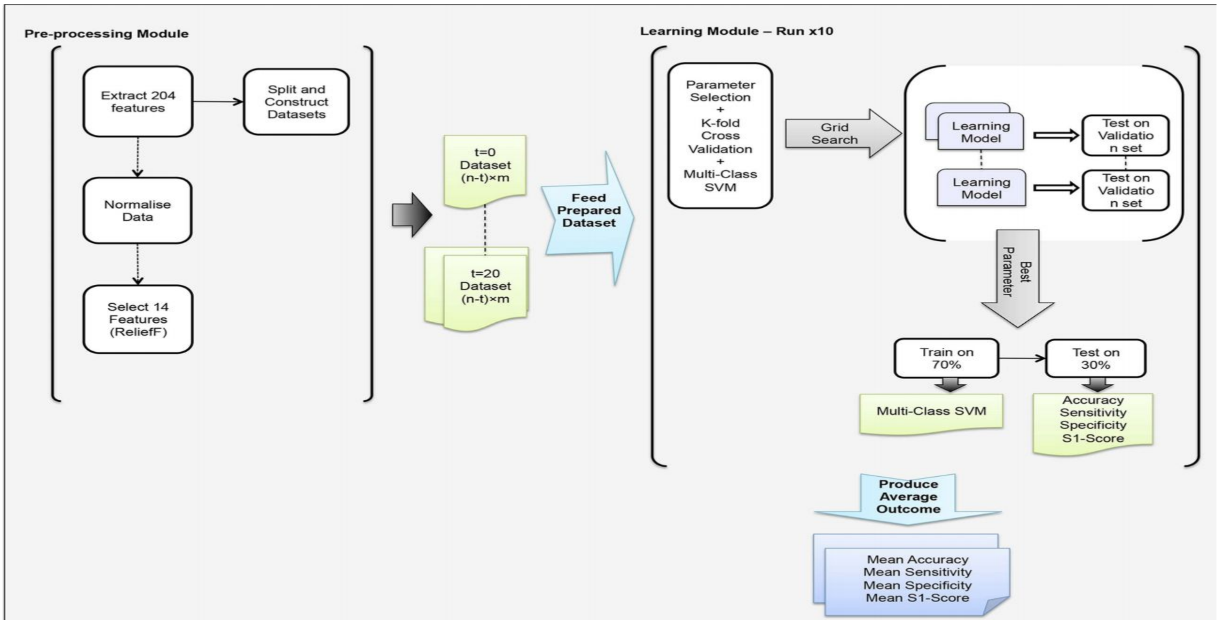
In this paper, the authors use the Freiburg Dataset which has both ‘ictal` and ‘inter-ictal’ data. In this study, only ‘ictal’ segments were used. Excessive Feature Engineering was done to calculate a total of 204 distinct features, out of which 34 are distinct features for each eeg segment. The set of features used were

1. Features based on energy
2. Features based on wavelet transform
3. Features based on non-linear dynamics

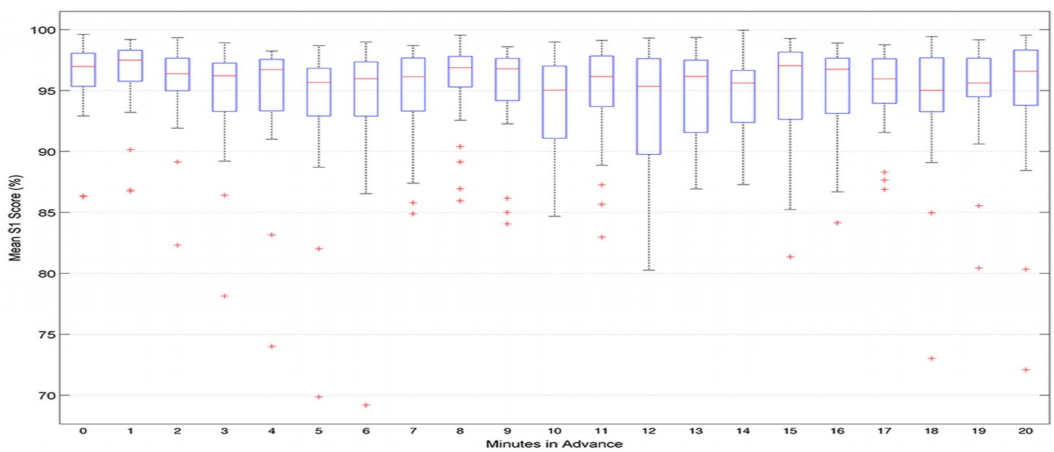
This paper has served as a base for research in the domain of forecasting seizures. The below image summarizes the approach used in this paper.

---

[4] ROC-AUC curve as blue and loss as organe



The works prior to this paper do not seem to include machine learning models into forecasting, this paper is novel in its approach and opens up a window of possibilities. This paper also does a comparative study of the accuracy of prediction of various lag periods as summarized in the below picture, we conclude that time (t) .



We see that the median accuracy of prediction remains almost the same for various time periods. This is probably because of the fact that the nearer the estimate the more variation, while farther the prediction period the lesser the data. Keeping this in mind and also the data constraints time horizon is selected for this project.

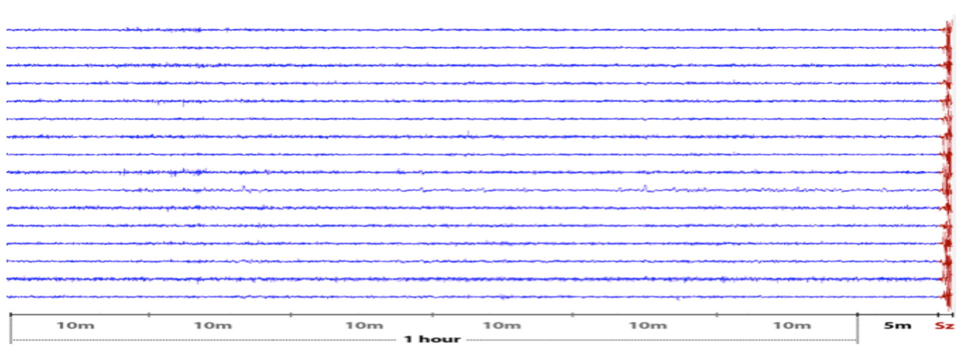
Data Specifications

The data was kindly provided by Dr Levin Kuhlmann of Melbourne University. There is emerging evidence that the temporal dynamics of brain activity can be classified into 4 states: Interictal (between seizures, or baseline), Preictal (prior to seizure), Ictal (seizure), and Post-ictal (after seizures). Seizure forecasting requires the ability to reliably identify a preictal state that can be differentiated from the interictal, ictal, and postictal state. The primary challenge in seizure forecasting is differentiating between the preictal and interictal states. The goal of the competition is to demonstrate the existence and accurate classification of the preictal brain state in humans with epilepsy.

Human brain activity was recorded in the form of intracranial EEG (iEEG) which involves electrodes positioned on the surface of the cerebral cortex and the recording of electrical signals with an ambulatory monitoring system. iEEG was sampled from 16 electrodes at 400 Hz, and recorded voltages were referenced to the electrode group average. These are long duration recordings, spanning multiple months up to multiple years and recording large numbers of seizures in some humans.

The challenge is to distinguish between ten-minute long data clips covering an hour prior to a seizure, and ten-minute iEEG clips of interictal activity. Seizures are known to cluster, or occur in groups. Patients who typically have seizure clusters receive little benefit from forecasting follow-on seizures. For this contest only lead seizures, defined here as seizures occurring four hours or more after another seizure, are included in the training and testing data sets. In order to avoid any potential contamination between interictal, preictal, and post-ictal EEG signals, interictal data segments were restricted to be at least four hours before or after any seizure. Interictal data segments were chosen at random within these restrictions.

Preictal training and testing data segments are provided covering one hour prior to seizure with a five-minute seizure horizon. (i.e. from 1:05 to 0:05 before seizure onset.) This pre-seizure horizon ensures that seizures could be predicted with enough warning to allow administration of fast-acting medications



Similarly, one-hour sequences of interictal ten-minute data segments are provided. The interictal data were chosen randomly from the full data record, with the restriction that interictal segments be at least 4 hours away from any seizure, to avoid contamination with preictal or postictal signals.

Any part of any 10-minute data segment can potentially contain “data drop-out” where the intracranial brain implant has temporarily failed to record data. This data drop-out corresponds to iEEG signal values of zeros across all channels at a given time sample. Data drop-out provides no predictive information as to whether a given 10-minute segment is preictal or interictal. A handful of 10-minute segments contain 100% data drop-out and cannot be classified. The data may also contain artefacts such as large amplitude rapid signal transitions that can be removed from the analysis.

Objective and Goal

After a bulk of literature review, there is a clear pathway to define the goal for this project. The literature review gave a glimpse of all the methods employed for seizure forecasting, while the spectral and temporal features capture the variation and signal strength which act as a discriminating factor in prediction, the CNN's are particularly useful for incorporating the spatial knowledge into the model. This project uses the best of both worlds and ensembles the methods to reach a state of the art results with minimal training and fine-tuning. Note that as heavy architectures are used, this project is not suitable for small devices and there needs to be some variation in the model size for the purpose of deploying the machine.

From the research, it was clear that not much of an emphasis was done on boosting tree-based models with success of algorithms like xgboost[5], it is very exciting to run the algorithm on this particular dataset.

The whole project is divided into six parts

1. Data Preprocessing
2. Feature Generation by Signal Processing Techniques
3. Building model on generated Features
4. Applying and training CNN on EEG as a spectrogram
5. Ensembling the two techniques
6. Benchmarking the results

Note - The aim of this project is to come up with Machine Learning based algorithms to warn patients of seizures some time t before the occurrence. Patients could avoid potentially dangerous activities like driving or swimming, and medications could be administered only when needed to prevent impending seizures, reducing overall side effects. A word of caution to use this study is that as it is known seizures are known to cluster, or occur in groups. Patients who typically have seizure clusters receive little benefit from forecasting follow-on seizures. Seizures that are instantly triggered also cannot be accurately forecasted beforehand, hence this study does not carry any benefits for those patients. For this project only lead seizures, defined here as seizures occurring four hours or more after another seizure, are included in the training and testing data sets.

---

5 https://homes.cs.washington.edu/~tqchen/2016/03/10/story-and-lessons-behind-the-evolution-of-xgboost.html

Data Preprocessing

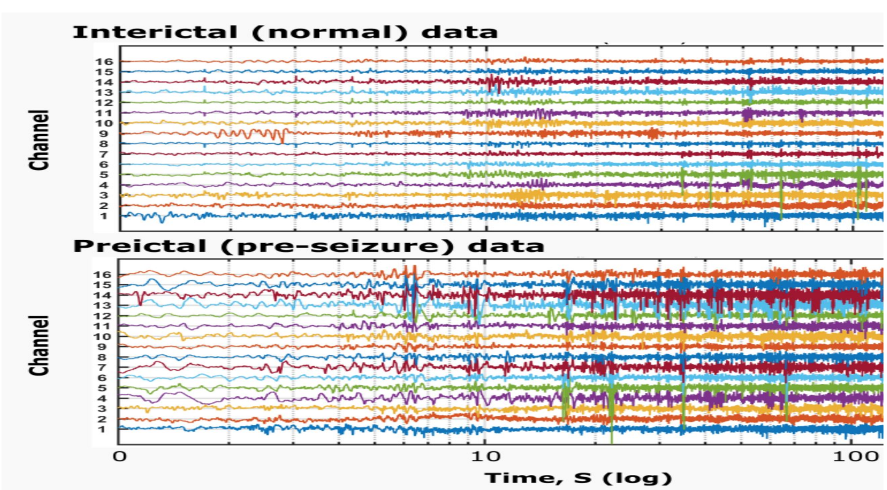
The first major part was preprocessing the following steps were followed

1. Due to less number of samples, the 10-minute data segments were divided into 100s segments. Padding was done while splitting data to maintain regularity in the data. As the dataset was labelled the labels had to be replicated for all the subsamples and to be kept the same as a superset of the samples. The sampling frequency was 400Hz, hence the changes had to be made accordingly. The below code segment describes the splitting procedure

```
#Dividing 10 minute segments into x seconds segments ie.
def divide_segment(x, segment, sampling_freq):
  '''Returns a list of  24000/(num_points) segments, each sub_segment has shape num_points*16'''
  div_list = []
  i = 0
  num_points = x*sampling_freq
  while(i*num_points<240000):
    div_list.append(segment[i*num_points:(i+1)*num_points,:])
    i+=1
  return div_list
```

2. Normalizing the data would also be important in order to compare data of various patients, scikit-learn's StandardScaler() was used for data normalization.

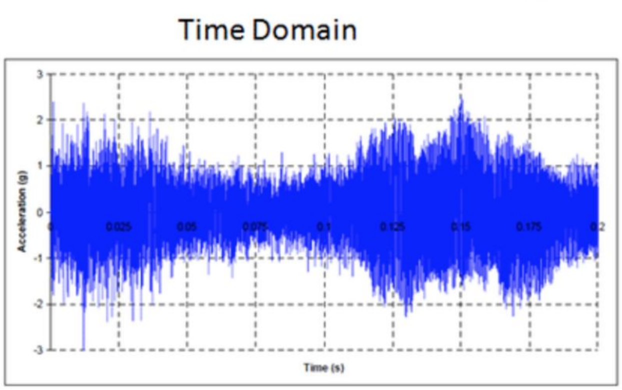
The below image aptly describes the two data segments I am trying to discriminate between

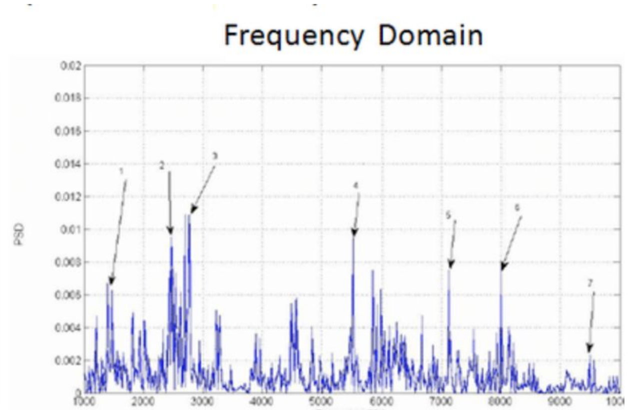


Feature Generation by signal processing techniques

1. **Fast-Fourier-Transformation**

   The Fourier transform is a mathematical analysis tool. The Fourier transform is used to analyze problems involving continuous-time signals or mixtures of continuous- and discrete-time signals. The discrete-time Fourier transform is used to analyze problems involving discrete-time signals or systems. In contrast, the discrete Fourier transform is the computational workhorse of signal processing. It is used solely for the numerical analysis of data. Lastly, the short-time Fourier transform is a variation of the discrete Fourier transform that is used for numerical analysis of data whose frequency content changes with time. Spectral analysis is an important aspect of signal processing. It is analogous to producing a score from a piece of music. The goal is to start with a signal and identify the strength of the sinusoidal components that make up the signal. The strength or amplitude of the sinusoids is displayed as a function of frequency. This is what is exactly done here, a signal in the time domain is converted to the frequency domain as shown below.





In EEG analysis bands of a particular range of frequency is of interest the bands are described below.

alpha waves = 8-13 Hz = Awake with eyes closed
beta waves = 14-30 Hz = Awake and thinking, interacting, doing calculations, etc.

gamma waves = 30-45 Hz = Might be related to consciousness and/or perception (particular 40 Hz)
theta waves = 4-7 Hz = Light sleep
delta waves < 3.5 Hz = Deep sleep

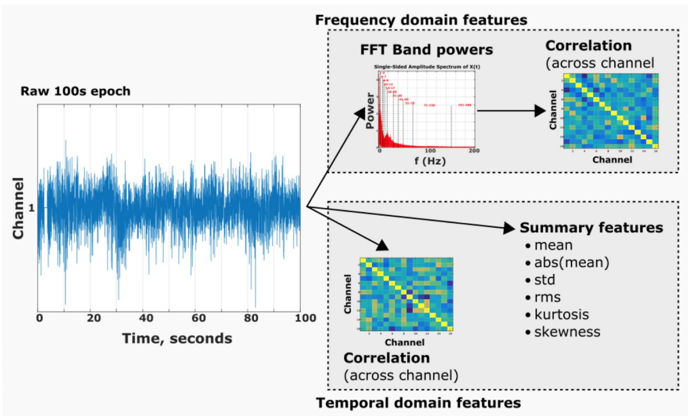
Fortunately, due to open source implementations available, fast Fourier transformations were easy to calculate. The famous numpy library was used for this as in the code section below

```
for i in tqdm(train_list):
    reg1 = r'.*?\_(.*)\..*'
    match = re.search(reg1, i)
    text = match.group(1)
    segment = text.split('_')[0]
    label = text.split('_')[1]
    temp = loadmat(path + '/' + i)['data']
    divided = divide_segment(100, temp, 400)
    for i,elem in enumerate(divided):
        #Extracting FFT
        fft = abs(np.fft.fft(elem, axis = 0))
        fft = fft.T
        gamma, theta, alpha, beta = get_amps(fft)
        gamma.extend(theta)
        gamma.extend(alpha)
        gamma.extend(beta)
        elem = ['segment_no' + segment + '_sub_{}'.format(i)] + gamma + [label]
        train_df = train_df.append(pd.DataFrame(elem).T)

def get_amps(fft):
    gamma = np.mean(fft[:, 0:3], axis = 1)
    gamma = (gamma - np.mean(gamma))/np.std(gamma)
    theta = np.mean(fft[:, 4:7], axis = 1)
    theta = (theta - np.mean(theta))/np.std(theta)
    alpha = np.mean(fft[:,8:13], axis = 1)
    alpha = (alpha - np.mean(alpha))/np.std(alpha)
    beta = np.mean(fft[:,14:30] , axis = 1)
    beta = (beta - np.mean(beta))/np.std(beta)
    return list(gamma), list(theta), list(alpha), list(beta)
```

This was done for each channel in the data hence the features space was multiplied by 16 for every new feature.

2. **Temporal Features**

The below image describes the feature selection idea.

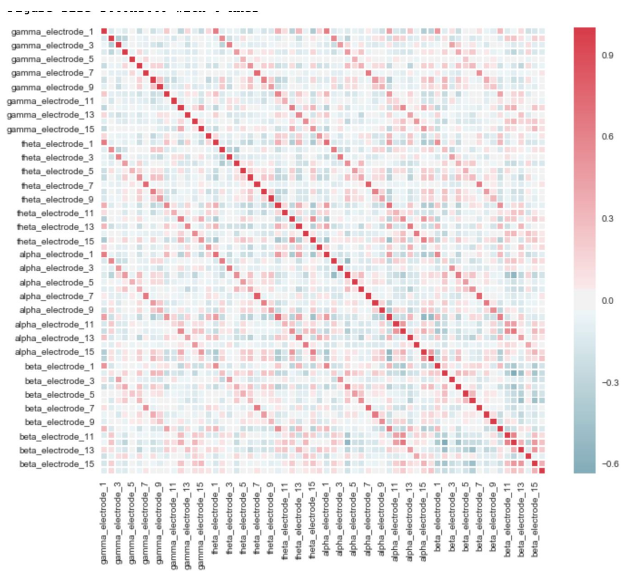


The Temporal feature consisted of statistics and distribution based features like, mean, std, skewness, kurtosis. Temporal domain features are necessary as they provide the time-based statistics of the signal, as no time series based models are used in this study these is the closest this project gets into the time domain. The code for the features is shown below

```
#Simple Temporal Domain Features Mean, Std, Kurtosis, Skewness
for i in tqdm(train_list):
    reg1 = r'.*?\_(.*)\..*'
    match = re.search(reg1, i)
    text = match.group(1)
    segment = text.split('_')[0]
    label = text.split('_')[1]
    temp = loadmat(path + '/' + i)['data']
    divided = divide_segment(100, temp, 400)
    for i,elem in enumerate(divided):
        mean = list(np.mean(elem,axis = 0))
        std = list(np.std(elem, axis = 0))
        kur = list(scipy.stats.kurtosis(elem,axis = 0))
        skew = list(scipy.stats.skew(elem,axis = 0))
        mean.extend(std)
        mean.extend(kur)
        mean.extend(skew)
        temporal_df = temporal_df.append(pd.DataFrame(mean).T)
```

3. **Correlation Between Channels as a Feature**

This feature selection was motivated by this paper[6]. Where the authors show that correlation features have high discriminative power, so those results are tested in this dataset. the correlation between channels was done in the time domain and also in the frequency domain. Some interesting trends were observed as in the correlation matrix below. It was observed that all the frequency bands of a particular channel were correlated. This was true for all three patients, hence this does make for an interesting feature.



Code for generating the results is shown below

```
#Simple correlation feature along temporal domain.
cross_df = pd.DataFrame()
for i in tqdm(train_list):
    reg1 = r'.*?\_(.*)\..*'
    match = re.search(reg1, i)
    text = match.group(1)
    segment = text.split('_')[0]
    label = text.split('_')[1]
    temp = loadmat(path + '/' + i)['data']
    divided = divide_segment(100, temp, 400)
    for i,elem in enumerate(divided):
        temp = pd.DataFrame(elem)
        corr_matrix = np.array(temp.corr())
        mask = np.tril_indices(16, k = -1)
        final_temp = corr_matrix[mask]
```

[6] EEG-based neonatal seizure detection with Support Vector Machines - A.Temko, E.Thomas, G.LightBody, G.Boylan

```
    final_temp = final_temp.flatten()
    cross_df = cross_df.append(pd.DataFrame(final_temp).T)
```

4. **Feature based on Spike Rate**

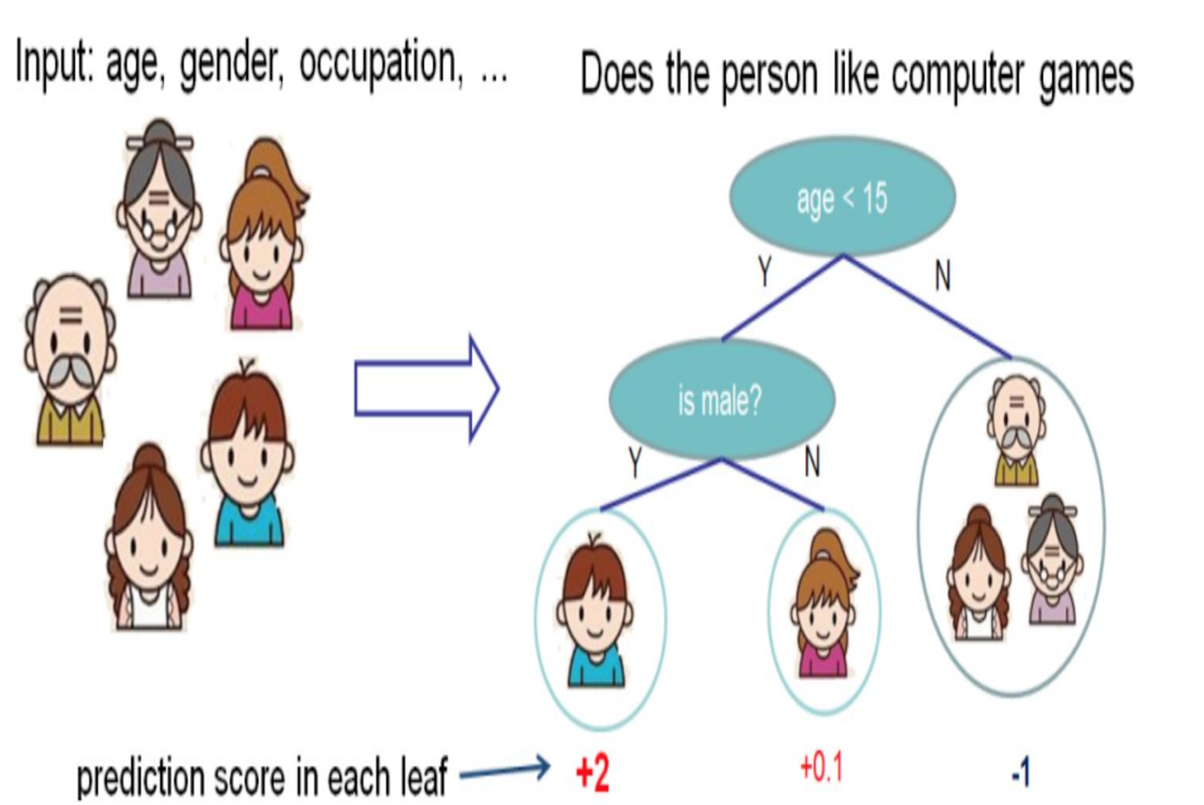
This was one of the most intuitive but the most interesting feature, as detailed in the literature review this feature had a huge potential in discriminating between the segments hence it was expected to be apt for this kind of data set, hence the feature was calculated. Due to the massive amount of calculation and no open source efficient implementation available, this had to be written from scratch with the use of parallel processing tricks, the multiprocessing trick gave a speedup of 6x which effectively making the task doable.

Below code is for calculating and creating the feature. Pythons OpenCV library was used for morphological operations

```
def calculate_spike_rate(segment_all_channels, threshold):
    segment_all_channels = segment_all_channels.T
    p = Pool()
    result = p.map(multicalc , segment_all_channels)
    p.close()
    p.join()
    return result

#For multiprocesing
def multicalc(channel):
    threshold = 0.5
    std_scale = StandardScaler()
    segment = std_scale.fit_transform(np.expand_dims(channel, axis = -1))
    total_spikes = cv2.morph((segment > threshold) + (segment < -1*threshold))
    spike_rate = sum(total_spikes > 0)/len(segment)
    return spike_rate

for i in tqdm(train_list):
    reg1 = r'.*?\_(.*)\..*'
    match = re.search(reg1, i)
    text = match.group(1)
    segment = text.split('_')[0]
    label = text.split('_')[1]
    temp = loadmat(path + '/' + i)['data']
    divided = divide_segment(100, temp, 400)
    for i,elem in enumerate(divided):
        spikes = calculate_spike_rate(elem, 0.5)
        spike_df = spike_df.append(pd.DataFrame(spikes).T)
```

Model XGBoost

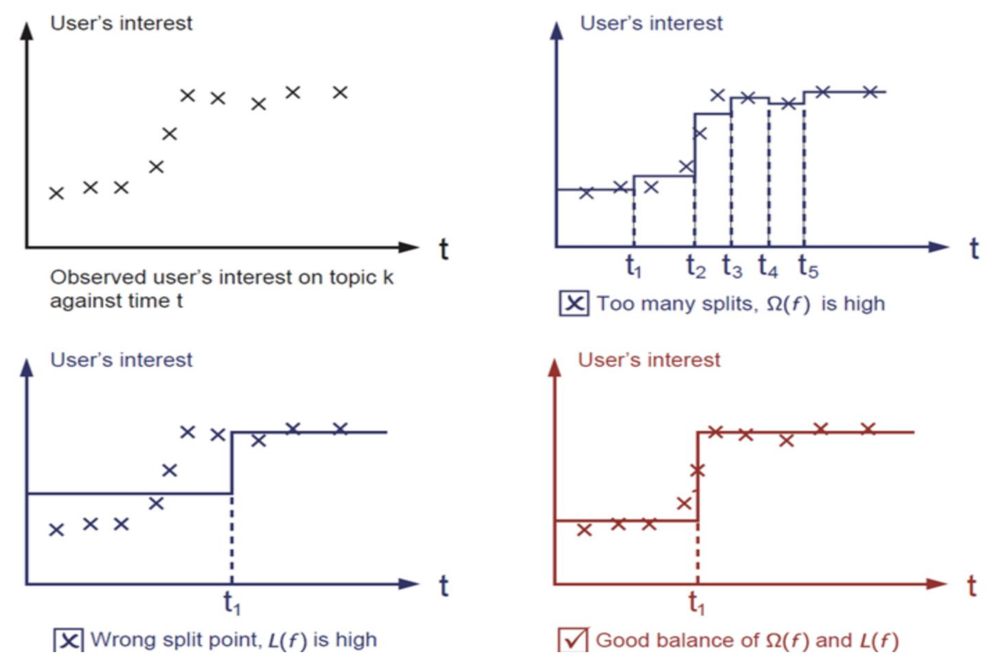
Xgboost also is known as Extreme Gradient Boosting algorithm, is widely known in the data science community. It operated based on a decision tree model as shown below



[7]

Traditional Decision Tree models are known to perform well and capture the interaction between columns due to their hierarchical structure, but they result in too many splits causing the data to overfit and require heavy pruning, hence gradient boosting methods help in building optimum tree without much pruning etc. Here is a explanatory example

[7] www.xgboost.docs.io



[8]

Trees in gradient boosting models are grown sequentially hence it is quite easy to monitor how much the model is learning after each iteration. This is what was done by the code shown below

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import pylab
from math import floor, log
from scipy.stats import skew, kurtosis
from scipy.io import loadmat
import scipy.fftpack
from tqdm import tqdm
import os
import re
import seaborn as sns
from string import ascii_letters
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import scipy
```

[8] www.xgboost.docs.io

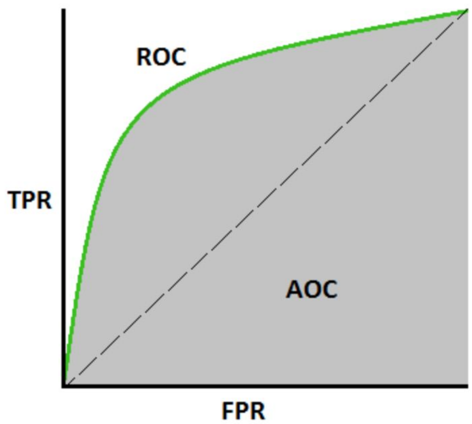
```python
import xgboost

df = pd.read_csv('train_pat1_fft_temporal_cross_corr.csv')
df.head()
from sklearn.model_selection import train_test_split
df.drop('segment_number', axis = 1, inplace = True)
X_full = df.drop('Class', axis = 1)
X_full.shape
y_full = df['Class']
X_train, X_val, y_train, y_val = train_test_split(X_full, y_full,
test_size = 0.2)
param = {'max_depth': 6, 'eta': 0.01, 'silent': 1, 'objective':
'binary:logistic', 'subsample' : 1,'eval_metric': 'logloss'}
xg_train = xgboost.DMatrix(X_train,label = y_train)
xg_val = xgboost.DMatrix(X_val,label = y_val)
xgb_model = xgboost.train(param ,
xg_train,early_stopping_rounds=10,evals = [(xg_train ,
"train_set"),(xg_val,"validation_set")] ,num_boost_round=1000)
```

After training this model along with the above set of features a validation roc-auc of 0.76 is obtained as can be generated from the above code.

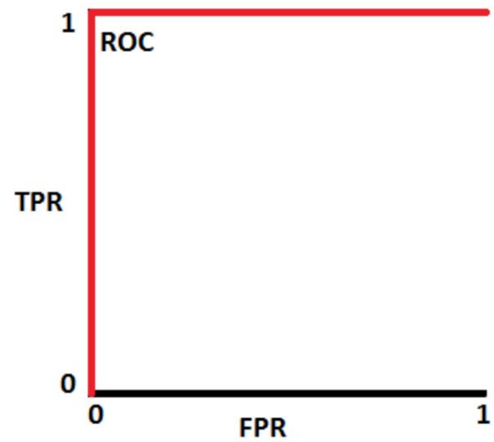
The most important feature recognized by the model was the spike rate followed by theta band, which is in accordance with the literature that theta band is found to be a strong predictor for epilepsy.

Brief Overview of ROC-AUC

ROC-AUC has become popular in the machine learning community due to its added advantage over traditional accuracy metric. ROC-AUC is independent of threshold and tells not the accuracy of classification but the ability of model to discriminate between two classes. This curve is obtained by plotting TPR (True Positive Rate vs False Positive Rate) at various thresholds from 0 to 1. Then the area under the curve gives the AUC metric. For a random classifier the AUC would be 0.5 as its ability to distinguish between classes would be absent hence this is considered as a baseline. The below figures show the current and extreme cases of this curve



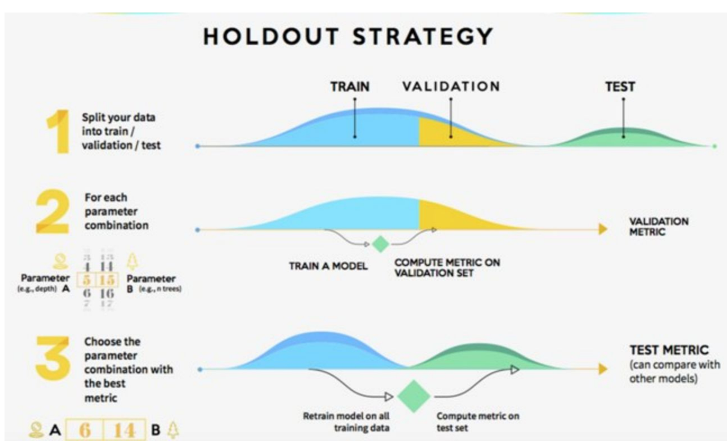
[9]



[10]

Validation Strategy

One of the most important ways to evaluate the model is to test it on the most real possible test set only then robustness of the model can be measured. So in this setting the out of 3 patients the third patients data was held as hold out set. This split was done using sklearn library's train_test_split function as shown in in the above code. This is called hold-out validation method.

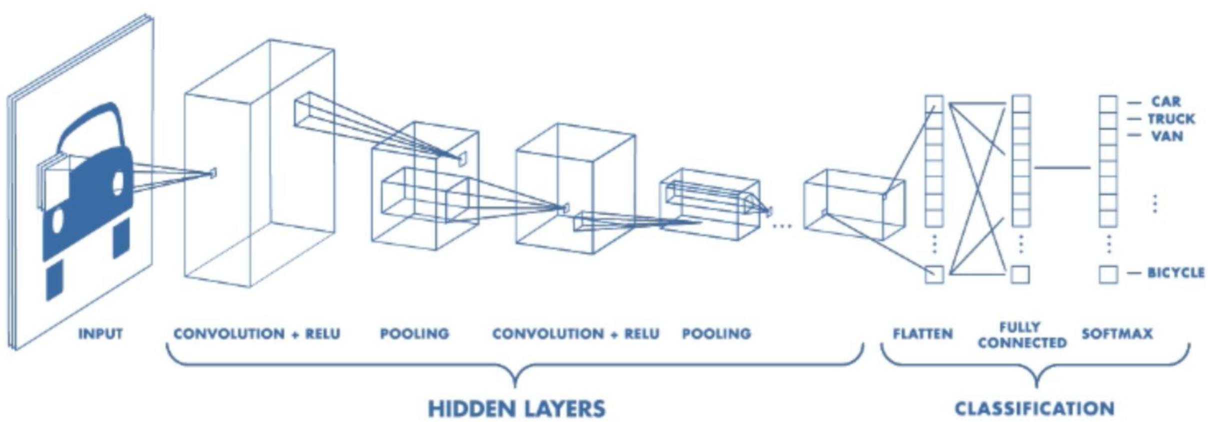


[11]

[9] AUC of the current xgboost model
[10] AUC of a perfect classifier ie. 1
[11] www.google.com

Convolutional Neural Networks

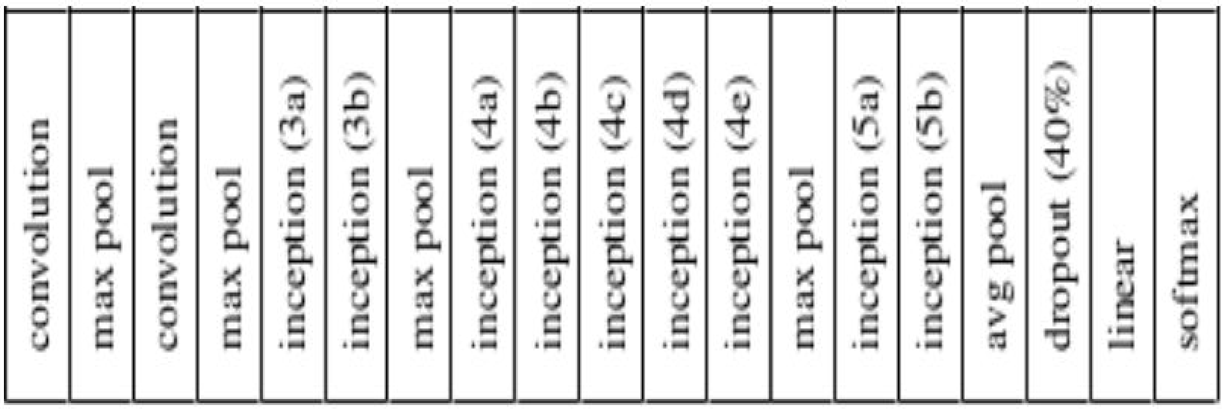
A convolutional neural network consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, RELU layer i.e. activation function, pooling layers, fully connected layers and normalization layers.Description of the process as a convolution in neural networks is by convention. Mathematically it is a cross-correlation rather than a convolution (although cross-correlation is a related operation). This only has significance for the indices in the matrix, and thus which weights are placed at which index.



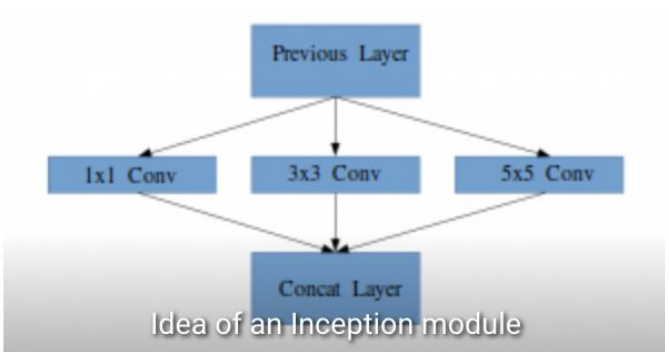
Convolutional neural network have been shown to achieve human-level performance in classification and detection. As the spectrogram of EEG shows an easily distinguishable by the human eye, it is expected that CNN's will perform well in this case as well. The previous research papers have already done this but they have used custom architectures which might not have resulted in good training in this report, we use the most famous Google's Inception architecture[12] which is shown to perform excellently on various tasks. Our Task here is a binary classification task and hence the architecture for inception is shown below.

[12] https://github.com/google/inception

The inception model



The most important feature of this network is the inception block, which makes the layer deeper but avoids it from overfitting at the same time. This is particularly suitable for this case due to less number of patients it is easy to overfit the model and sabotage the model performance. Hence the choice of this model which suits this particular dataet very well.



[13]

The important thing to note is that the image is resized and even though the network requires 3 channels but the data being a single channel. This is not an issue as open source libraries like keras handle this detail. The code is super easy to set up. Due to images occupying a lot of RAM it is not possible to load all images(spectrograms) at once hence the images are loaded in batches into the RAM memory. The batch generators along with

---

[13] The Inception block

```
from keras.applications.inception_v3 import preprocess_input
train_datagen = ImageDataGenerator(
   preprocessing_function=preprocess_input,
   rotation_range=40,
   width_shift_range=0.2,
   height_shift_range=0.2,
   shear_range=0.2,
   zoom_range=0.2,
   horizontal_flip=True,
   fill_mode='nearest')VALIDATION_STEPS = 64
MODEL_FILE = 'epilepsy.model'
history = model.fit_generator(
   train_generator,
   epochs=EPOCHS,
   steps_per_epoch=STEPS_PER_EPOCH,
   validation_data=validation_generator,
   validation_steps=VALIDATION_STEPS)

model.save(MODEL_FILE)
```

The training was done for 6 hours on a RTX2070 GPU, the validation score was found to be 0.77. Which is remarkable as CNN itself extracted features and performed on par with models that required manual feature generation and other procedures. The final part will focus on the ensemble of both the models ie. classical feature model and CNN.

To add further predictive power to the model, the technique of ensembling is used which builds upon the current two models which is discusses in the next section.

Ensembling

Ensemble Methods, what are they?

Ensemble methods is a machine learning technique that combines several base models in order to produce one optimal predictive model. To better understand this definition lets take a step back into the ultimate goal of machine learning and model building. This is going to make more sense as I dive into specific examples and why Ensemble methods are used. An example figure is shown below

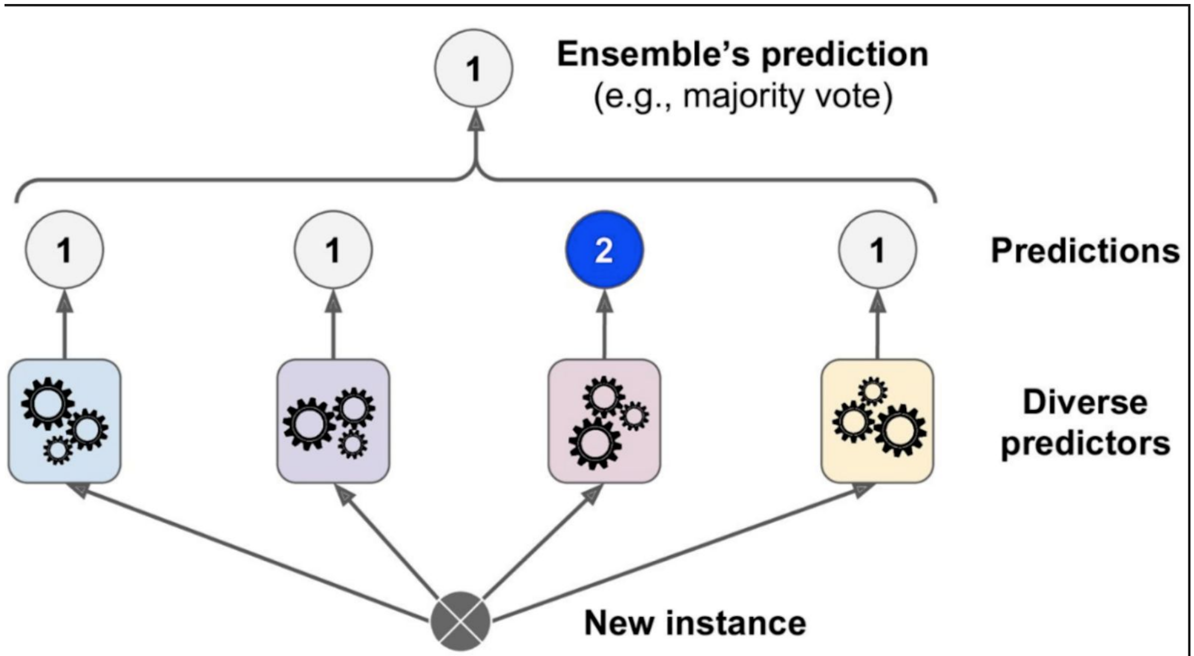


Figure 7-2. Hard voting classifier predictions

There are various reasons that a simple averaging can lead to high-performance gains

1. Variance is reduced by averaging
2. Model Biases are eased out by providing equal weight

The above two points are crucial for any Machine Learning Model's Generalizability.

Simple Averaging method is used in the case of this model which alone leads to a performance gain of 0.02, finally an averaging produces an roc-auc of 0.79, which is very close to the current state of the art.

Results and Way Forward

The below table summarizes the results

Model	Train Score	Val Score
Xgboost with features	0.79	0.76
CNN	0.83	0.77
Average Ensemble	0.82	0.79

This summarizes that Xgboost and tree-based models are indeed effective in the case of epilepsy prediction in advance. While trees do not capture spatial relationship, CNN the the tree based model with a spatial understanding of the data. Making both models diverse and good for ensembling.

The things that remain to try and that can further increase predictive power.

1. Fine tuning - Both the models ie. Xgboost and CNN can lead to further improvement but require heavy computational resources.
2. Stacking - It is a version of ensembling where a meta learning model is created based on the output of the two models.
3. We note that I've not used time-based model anywhere models are based on characteristics and spatial arrangement. With the development of RNN's which are ideally suited for time domain prediction a new, another dimension to the study can be acheieved.

Overall it is to conclude that Machine Learning models can be deployed with pretty high confidence into hospitals/clinics to automate redundant tasks of doctors and let them concentrate on the deeper aspects. The three points above need to be taken into consideration and will be looked into when forming the final research article over the summers.

Literature Review

- Amir Eftekhar, Walid Juffali, Jamil El-Imad, Timothy G. Constandinou, and Christofer Touma- zou. Ngram-derived pattern recognition for the detection and prediction of epileptic seizures. *PLOS ONE*, 9(6):1–15, 06 2014. doi: 10.1371/journal.pone.0096235.
- Dean R Freestone, Mark J Cook, David B Grayden, Philippa Karoly, Levin Kuhlmann, David T J Liley, Gareth Jones, Qingnan Tang, Irina Ivanenko, Oleg Panichev, Timothée Proix, Michal Náhlík, Daniel B Grunberg, Chip Reuben, Brian Litt, Benjamin H Brinkmann, Gregory Worrell, Andriy Temko, Alexandre Barachant, Feng Li, Jr. Titericz, Gilberto, Brian W Lang, Daniel Lavery, Derek Broadhead, Kelly Roman, and Scott Dobson. Epilepsyecosystem.org: crowd- sourcing reproducible seizure prediction with long-term human intracranial EEG. *Brain*, 141 (9):2619–2630, 08 2018. ISSN 0006-8950. doi: 10.1093/brain/awy210.
- “Crowdsourcing reproducible seizure forecasting in human B. H. Brinkmann, J. Wagenaar et al. and vol. 139 no. 6 pp. 1713–1722 2016. canine epilepsy, ” Brain.
- Nhan Duy Truong, Anh Duy Nguyen, Levin Kuhlmann, Mohammad Reza Bonyadi, Jiawei Yang, Samuel Ippolito, and Omid Kavehei. Convolutional neural networks for seizure pre- diction using intracranial and scalp electroencephalogram. *Neural Networks*, 105:104 – 111, 2018. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2018.04.018.
- S. Li, W. Zhou, Q. Yuan, and Y. Liu. Seizure prediction using spike rate of intracranial eeg. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 21(6):880–886, Nov 2013. ISSN 1534-4320. doi: 10.1109/TNSRE.2013.2282153.
- Negin Moghim and David W. Corne. Predicting epileptic seizures in advance. *PLOS ONE*, 9 (6):1–17, 06 2014. doi: 10.1371/journal.pone.0099334.