Udayan Gaikwad, 2023, 11:6
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

Seizure Detection and Probability Prediction Using Random Forests

Udayan Gaikwad, Mukta Patil, Akash Bhagwat,
Saniya Inamdar, Saraswati Patil
BRACT's Vishwakarma Institute of Technology
Bibwewadi, Pune, Maharashtra, INDIA

Abstract- This paper implements methodologies for seizure detection and prediction using mathematical techniques like FFT and Machine Learning classifiers such as Random Forest. The dataset for this project includes both training data, called Ictal and testing data called Interictal. For data pre-processing the Fast Fourier Transform is applied to each 1 second clip, taking frequency magnitudes in the range 1-47Hz and discarding phase information. Correlation coefficients and their eigenvalues are then calculated in both the time and frequency domains and are appended to the FFT data to form the feature set. This feature set is then trained on a Random Forest classifier using 3000 trees. The approach is used to train per-patient classifiers. A prediction module concludes this project by presenting the probability of seizure within a patient. The results are visualized for easy and clear representation.

Keywords- continuous electroencephalography, grid search optimization, random forest, epileptic seizure detection, simulation model

I. FEATURE SELECTION

The features used in the model were determined through heavy experimentation. Study of the literature of prior work in the field as well as our own study of signal processing and machine learning gave inspiration for ideas to experiment with. Features were kept or discarded based on their cross-validation performance. The first thing we experimented on with was Fast Fourier Transform on the belief that the various magnitudes of different frequencies would provide a good source of features. This turned out to be correct with FFT alone providing the best single-feature model compared to all other singular pre-processing steps we tried. Combinations of multiple features eventually proved to provide a better classification score once the right features were combined.

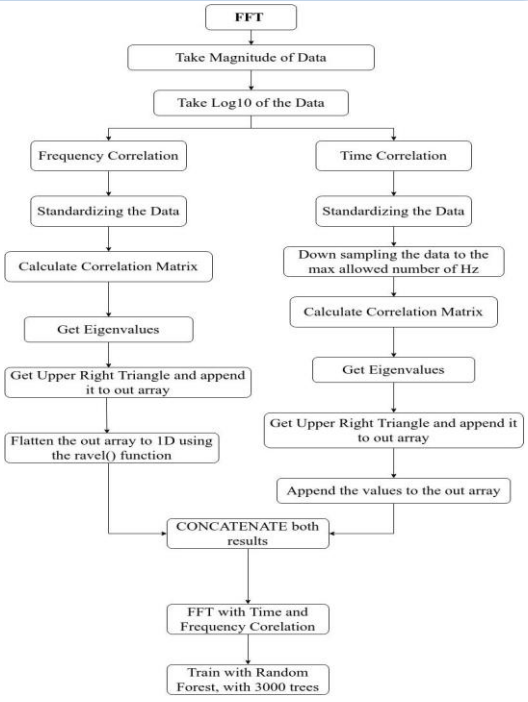

Fig 1. Flowchart explaining the process of Feature Selection.

It was shown by Schindler, Leung, Elger & Lehnertz (2007) [6] that correlation coefficients in the time domain and their corresponding eigenvalues are effective features for seizure detection. So we decided to apply this technique over both the time domain of the original data, and in the frequency domain. Three sources of features are used to form the whole feature-set:

1. FFT Magnitudes in the Low Frequency Range 1 To 47Hz

Time-series -> FFT -> Slice(1, 47) -> Magnitude -> Log10

Fast Fourier Transform is applied to each 1 second clip across all EEG channels, taking log10 of the magnitudes of frequencies in the range 1-47Hz. Phase information is discarded. The output of each training example in this stage is in the shape (N, 47) where N is the number of EEG channels used for a given patient.

The Fourier transform (FT) of the function f(x) is the function F(ω), where:

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i\omega x} dx$$

and the inverse Fourier transform is:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega x} d\omega$$

The range 1-47Hz was chosen through trial and error while attempting dimensionality reduction. It turned out to give a better result than using other frequency ranges such as 1-63Hz, 1-127 Hz, 1-191Hz, or the entire frequency range. Note that we have also omitted the 0Hz frequency, assuming this bias to be attributed to instrument error and would contribute only noise.

Finally, the frequency features from all of the different channels are concatenated together when used for training.

2. Correlation Coefficients (Between EEG Channels) and Their Eigenvalues in the Frequency Domain

FFT 1-47Hz -> normalization -> correlation coefficients -> eigenvalues

The output of the FFT stage with shape (N, 47) is then normalized across frequency buckets. For example, all the 1Hz buckets for each EEG channel are treated as a single vector, subtracting the mean and dividing by standard deviation.

This is done for every frequency bucket from 1Hz up to 47Hz. The correlation coefficients (N, N) matrix is calculated from this normalized matrix of (N, 47). Real eigenvalues are calculated on this correlation coefficient matrix with complex eigenvalues made real by taking the complex magnitude. The output of this stage is the upper right triangle of the correlation coefficient matrix (as it is symmetric there is redundant data), and the eigenvalues are sorted by magnitude.

3. Correlation Coefficients (Between EEG Channels) and Their Eigenvalues in the Time Domain.

Time series -> normalization -> correlation coefficients -> eigenvalues

Schindler, Leung, Elger & Lehnertz (2007) [6] showed this to be an effective technique. However, we believe they normalized each channel independently, using mean and standard deviation across all the time-samples within a single channel.

The cross validation showed that normalizing across samples performed the same as not applying normalization at all, and that normalizing each feature performed worse. However, leader board submissions showed better results normalizing each feature. The features from this stage are calculated the same as stage 2, but the input source of data is the original time-series source data instead of the FFT output. Like the previous stage, both the upper triangle of the correlation coefficients and the sorted eigenvalues are used as features. The output of all 3 stages is combined together to form the feature set.

The peculiarity in the solution is that normalizing across features (axis=0) instead of on samples performed better on the leaderboard, but worse in cross-validation. In cross-validation better performance was obtained by normalization across samples (axis=1) or by not normalizing at all.

II. MODELLING TECHNIQUES AND TRAINING

Our background is Computer Engineering with self-study of signal processing and machine learning. Our focus was therefore on writing performant code that would allow us to systematically experiment with different classifiers and different mathematical transforms to get quick feedback about what works and what doesn't. It was a greedy brute force approach, trying the most promising techniques first but otherwise trying every technique we could find. To facilitate this heavy experimentation, we wrote a rudimentary framework to allow quick cross-validation of many different combinations of data processing techniques and classifiers.

1. Experimentation Framework

Like the feature selection, the chosen model for classification was determined through experimentation. Each run gives a cross-validation score matching the scoring criteria used by the competition leaderboard (ROC AUC). Combinations of feature sets and classifiers that gave higher scores were explored further. This approach was made easy using the scikit-learn python machine learning library which offers many different classifiers that can be easily substituted in the code. Many different classifiers were tried in succession from logistic regression to decision trees or support vector machines. There are too many in scikit-learn to list here. However Random Forest offered not only the best performance but also consistent performance. The performance of SVM for example changed wildly with only small changes to the tuning parameters. Some manual experimentation was then done to determine good parameters to maximize the Random Forest performance.

The following python scikit-learn classifier was used to train the winning submission:
Random Forest Classifier (n_estimators=3000, min_ samples_ split=1, bootstrap=False, random_state=0)

2. Cross-Validation

Initially we split the training set and cross-validation randomly. However, scores achieved in cross-validation varied a significant amount from leaderboard scores suggesting this was not a good approach. Following advice on the forums from Kaggle user Alexandre, we split the ictal training data based on whole seizures. For example for a ratio of 0.25, if there were 4 seizures then 1 entire seizure was split out leaving the other 3 to train on. Some seizures for each patient were of different length. We sorted the seizures by length, then took the cross validation set from the middle, and trained on the shorter/longer seizures. This gave cross-validation scores that matched them very closely. However once cross-validation scores reached the 0.96-0.97 mark it was no longer sufficient to reliably rank models against each other and we fell back to submitting models based on intuition and checking performance on the leaderboard. For example, our first submission of FFT combined with correlation data increased my score significantly on the public leader board from 0.95748 to 0.96961, an increase of 0.01213. However in cross-validation we saw an order of magnitude smaller increase of approximately 0.001. This indicates that there is perhaps room to improve in the cross-validation setup. These last numbers were taken with a cross-validation split of 50%. Other splits to try might be to train on the medium-length seizures, and cross-validate on the longer/shorter ones. However, after optimization, and running cross validation thrice we achieved accuracy of 98%, 96% and 94% respectively.

An alternative option for cross validation would be to use k-fold cross-validation, which takes a longer time for evaluation making it less desirable.

III. CROSS-VALIDATION RESULTS

The following is an example of the kind of output (although some data removed to make it easier to read in this document) that could be produced after 1 hour of trying different data processing steps. This includes trying various transforms such as FFT, frequency correlation, time correlation, MFCC, Daubechies wavelet and just basic stats of the time-series data. Note this is not an exhaustive list of the methods that we have tried to implement.

Table 1: Cross validation Results.

SN	Method	Norm.	Acc.ury
1	FFT with Time and Frequency Correlation taking frequency slice 1-48	none	96.750
2	FFT with Time and Frequency Correlation taking frequency slice 1-48	uss	96.750
3	FFT with frequency slice 1-48 and taking the magnitude and log 10 of the result	none	96.623
4	FFT with frequency slice 1-128 and taking the magnitude and log 10 of the result	none	96.496
5	Daubechies Wavelet coefficients	none	96.315
6	FFT with frequency slice 1-96 and taking the magnitude and log 10 of the result	none	96.257
7	Gen FFT with Time and Frequency Correlation taking frequency slice 1-48	none	96.237
8	Gen FFT with Time and Frequency Correlation taking frequency slice 1-48	us	96.237
9	Resampling time series data using a Hanning window and then taking the Daubechies Wavelet coefficients	none	96.188
10	FFT with frequency slice 1-160 and taking the magnitude and log 10 of the result	non	96.151
11	FFT with Time and Frequency Correlation taking frequency slice 1-48	usf	96.147
12	FFT and taking the magnitude and log 10 of the result	none	96.069
13	Gen FFT with Time and Frequency Correlation taking frequency slice 1-48	usf	95.886
14	FFT with frequency slice 1-64 and taking magnitude and log 10 of the result	none	95.818
15	Resampling time series data using a Hanning window and taking the Mel-frequency coefficients	none	95.462
16	Time Frequency correlation taking frequency slice 1-48	none	95.253
17	Time Frequency correlation taking frequency slice 1-48	us	95.253
18	Time Frequency correlation taking frequency slice 1-48	usf	93.334

All validators are implemented in random forests with 150 estimators, minimum sample split as 1, no bootstrap, and random state as 0

The tags us, usf and none refer to the normalization option used with us referring to normalizing across all samples within a channel, and usf referring to normalizing each feature across all channels one by one.

IV. CONCLUSION

Thus we have achieved our aim of seizure detection and prediction using Mathematical and ML techniques by the implementation of Fast Fourier Transform and Random forest. We discussed in detail how the methods chosen were the optimal choices for this project. We have personally tried various alternatives and found these to be the best, providing fast and accurate results with great consistency. The prediction module of the project also helps identify the probability of seizure onset, thereby warning both the patient and the doctor. This could help prevent the seizure and deal with potential risks it proposes. It could really mean the difference of life and death for some. We hope to have achieved this target through this software.

REFERENCES

1. Koch, M., Uyttenboogaart, M., Polman, S. and De Keyser, J., 2008. Seizures in multiple sclerosis. Epilepsia, 49(6), pp.948-953.
2. Basri, A. and Arif, M., 2021. Classification of Seizure Types Using Random Forest Classifier. Advances in Science and Technology. Research Journal, 15(3).
3. Messaoud, R.B. and Chavez, M., 2021. Random Forest classifier for EEG-based seizure prediction. arXiv preprint arXiv:2106.04510.
4. Kaspar Schindler, Howan Leung, Christian E. Elger, Klaus Lehnertz, Assessing seizure dynamics by analyzing the correlation structure of multichannel intracranial EEG, Brain, Volume 130, Issue 1, January 2007, Pages 65–77, https://doi.org/10.1093/brain/awl304
5. Shoeb, Ali H., and John V. Guttag. "Application of machine learning to epileptic seizure

detection." Proceedings of the 27th international conference on machine learning (ICML-10). 2010.

6. Tzallas, Alexandros T., Markos G. Tsipouras, and Dimitrios I. Fotiadis. "Epileptic seizure detection in EEGs using time–frequency analysis." IEEE transactions on information technology in biomedicine 13.5 (2009): 703-710.
7. Shoeb, Ali H., and John V. Guttag. "Application of machine learning to epileptic seizure detection." Proceedings of the 27th international conference on machine learning (ICML-10). 2010.