PyData Lancaster

Wednesday the 27th of November, 2019 🍂

Automatic Detection of Epilepsy Seizures in NHS Electroencephalography Records

(using classical machine learning models)

Presented by David Luke Elliott

Department of Psychology, D37 Fylde College, Lancaster University, Lancaster, LA1 4YF

Introduction

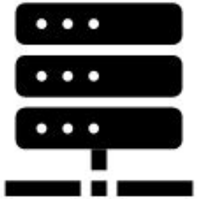
My work focuses on developing hardware, software, and algorithms for Electroencephalography (EEG) monitoring of patients with epilepsy.

I am a psychology methods researcher, with the emphasis on the methods (Data Scientist?).

Focus of Today's Talk

Epilepsy

Clinical Decision Support



Classification

Imbalanced Labels

Classical Methods

Bayesian Optimization

What is Epilepsy

Epilepsy is the tendency to have unprovoked and recurrent seizures.

Seizures are caused by neuronal hyperexcitability and excessive electrical discharges.

There are over 40 types of epilepsy and seizures, of which individuals may experience several.

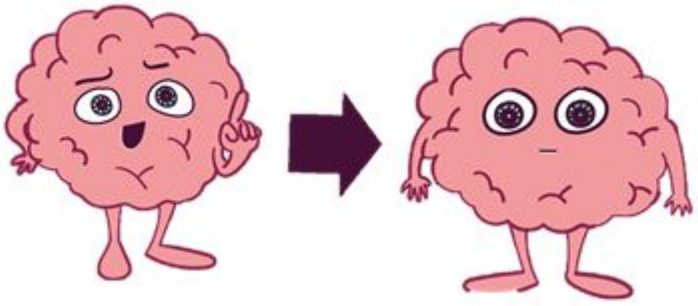
Absence Epilepsy

Absence seizures can develop during childhood (6- to 7-years) or early adolescence (~12-years).

Constitutes around 10% of paediatric epilepsy patients.

Clinical symptoms include…

- Blank stare
- Interrupted activities
- Slowed speech
- Upward rotation of the eyes



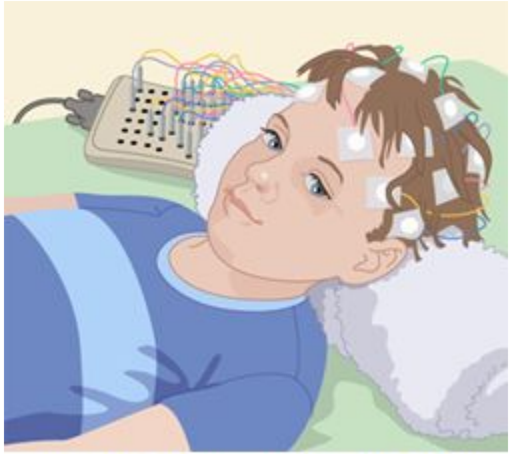
NHS Epilepsy Diagnosis

Diagnosing an epilepsy syndrome is primarily reliant on:

- Patient report
- Identification of clinical features in diagnostic imaging
  - **Electroencephalography (EEG)**
  - Magnetic Resonance Imaging (MRI)
  - Computed Tomography (CT)
  - Video recordings

NHS Epilepsy Diagnosis

A patient's medical history, along with ~30-minute scalp EEG assessment (sometimes also measuring heart rate and blood oxygen saturation), is commonly first assessed.



https://kidshealth.org/en/parents/eeg.html

During the assessment, the patient may be asked to hyperventilate or exposed to flashing lights (photic stimulation) to provoke a seizure.

The patient is monitored by staff, who note events on the records to aid retrospective analysis. If a diagnosis is suspected, but not gained, a patient may then have a longer EEG assessment.

Data Analysis

Manual review of EEG is

- Time consuming
- Expensive
- Prone to error

95-99% of the recorded data is useless for diagnosis

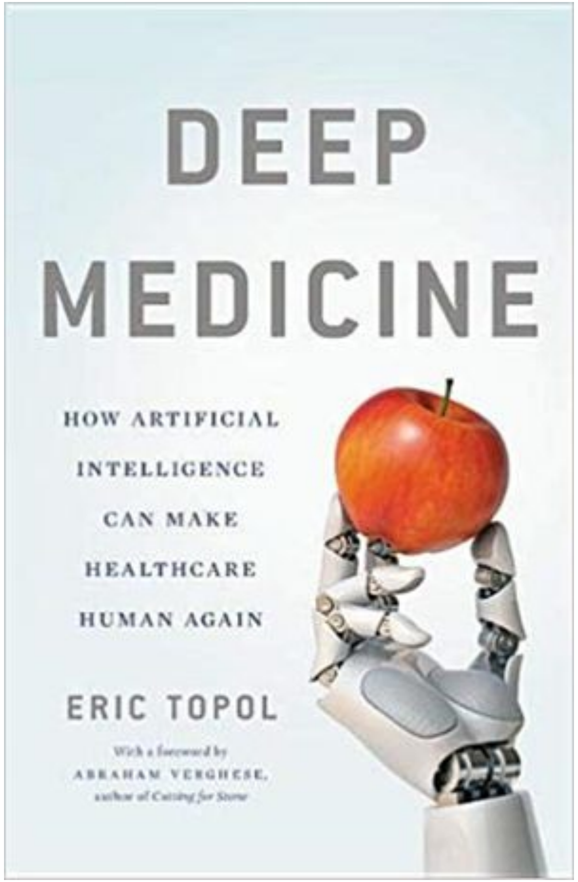
Data Analysis

Algorithms to assist medical practice have been around for decades.

- Computer aided ECG's have been around since the 1970's.
- Use static rule-based models (heuristics) with limited accuracy.

Machine learning models are increasingly being applied to diagnostic imaging:

- Radiology
- Dermatology
- Clinical pathology



Basic Signal Processing System

Pre-processing

Prepare the raw signal

Feature extraction

Quantify values or features of the signal (e.g. biomarkers or artefacts)

Classification

Applying a threshold or model-based criteria

- Model-based classification requires additional feature reduction or extraction, and a training or supervised learning step

Expert System

The global strategy that is developed

- Which features to select
- How to combine features
- Account for contextual information

Basic Signal Processing System

Algorithms generally can be designed for efficiency (online) or accuracy (offline)

Seizure-event detector

Aim

- Identify seizures with the greatest possible sensitivity/specificity/precision

Use

- Provide a summary of frequency, duration, and time of a patient's seizures to enable physicians diagnose and better titrate therapy

Seizure predictors

Aim

- Predict seizures with the greatest accuracy and time in advance

Use

- Trigger neurostimulators to prevent a seizure
- Provide warning that a patient may have a seizure

Seizure-onset detector

Aim

- Detect the onset of a seizure with the shortest possible delay

Use

- Initiate functional neuroimaging to localise the cerebral origin of a seizure
- Trigger neurostimulators to affect seizure progression
- Alert a carer to the patient’s condition or call emergency response

Basic Signal Processing System

Algorithms generally can be designed for efficiency (online) or accuracy (offline)

Seizure-event detector

Aim

- Identify seizures with the greatest possible sensitivity/specificity/precision

Use

- Provide a summary of frequency, duration, and time of a patient's seizures to enable physicians diagnose and better titrate therapy

Seizure predictors

Aim

- Predict seizures with the greatest accuracy and time in advance

Use

- Trigger neurostimulators to prevent a seizure
- Provide warning that a patient may have a seizure

Seizure-onset detector

Aim

- Detect the onset of a seizure with the shortest possible delay

Use

- Initiate functional neuroimaging to localise the cerebral origin of a seizure
- Trigger neurostimulators to affect seizure progression
- Alert a carer to the patient’s condition or call emergency response

Basic Signal Processing System

Models can be trained and tested in various ways for different use cases

Patient-General

Training

- Models are trained on records from a number of patients and tested on a separate test group

Use

- Clinical decision making (diagnosis, treatment)

Patient-Specific

Training

- Trained only on data from an individual patient to detect/predict future seizures
- A patient general algorithm is adapted to fit an individual patient (e.g. Semi-supervised Reinforcement Learning/Transfer Learning)

Use

- Ambulatory (home) patient monitoring

Basic Signal Processing System

Models can be trained and tested in various ways for different use cases

Patient-General

Training

- Models are trained on records from a number of patients and tested on a separate test group

Use

- Clinical decision making (diagnosis, treatment)

Patient-Specific

Training

- Trained only on data from an individual patient to detect/predict future seizures
- A patient general algorithm is adapted to fit an individual patient (e.g. Semi-supervised Reinforcement Learning/Transfer Learning)

Use

- Ambulatory (home) patient monitoring

Study 1

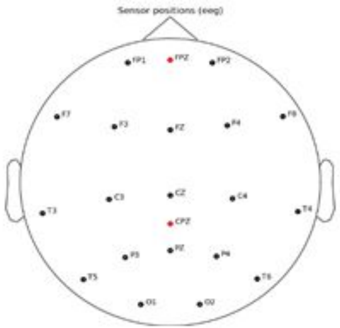
- Dataset with Ecological Validity
- Large Feature Space
- Classical Models
- Exploration Over Pipeline Components & Hyperparameters (Bayesian Optimisation)

Data Collection

EEG records from 21 pediatric patients (ages 4-13) diagnosed with absence epilepsy (~11hrs).

Patients underwent a routine clinical EEG assessment, lasting approximately 30 minutes, and were asked to hyperventilate or exposed to photic stimulation to provoke a seizure.

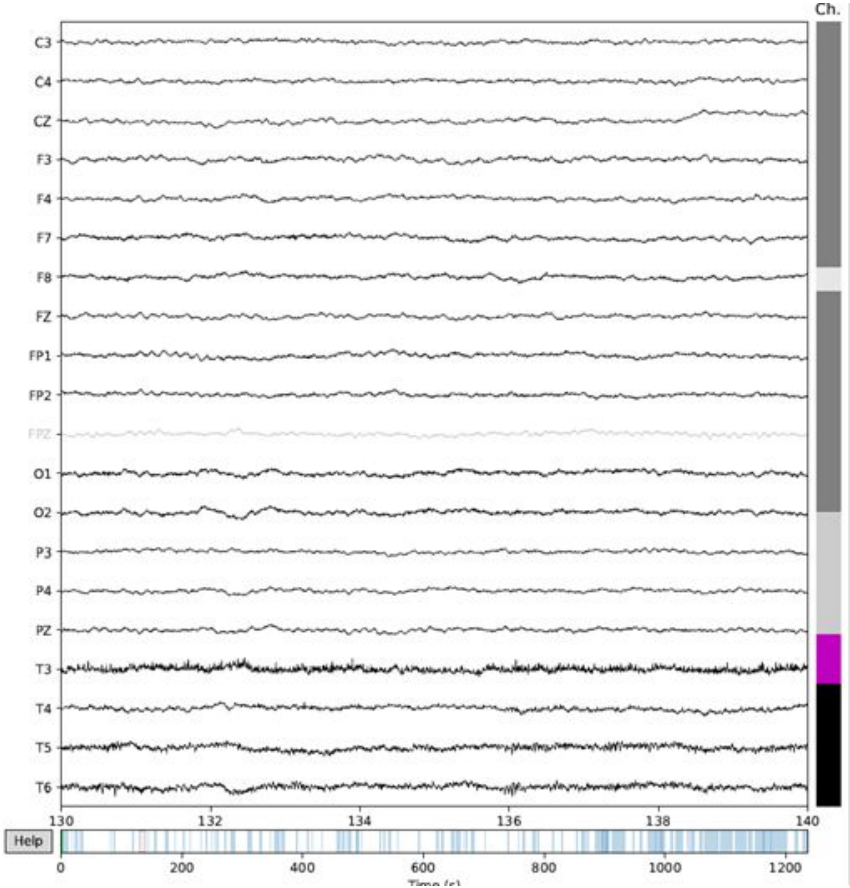
Data from these sessions were anonymized and burned to a CD by a clinical physiologist after being used for diagnostic purposes.



Pre-processing

Baseline (47.60%)

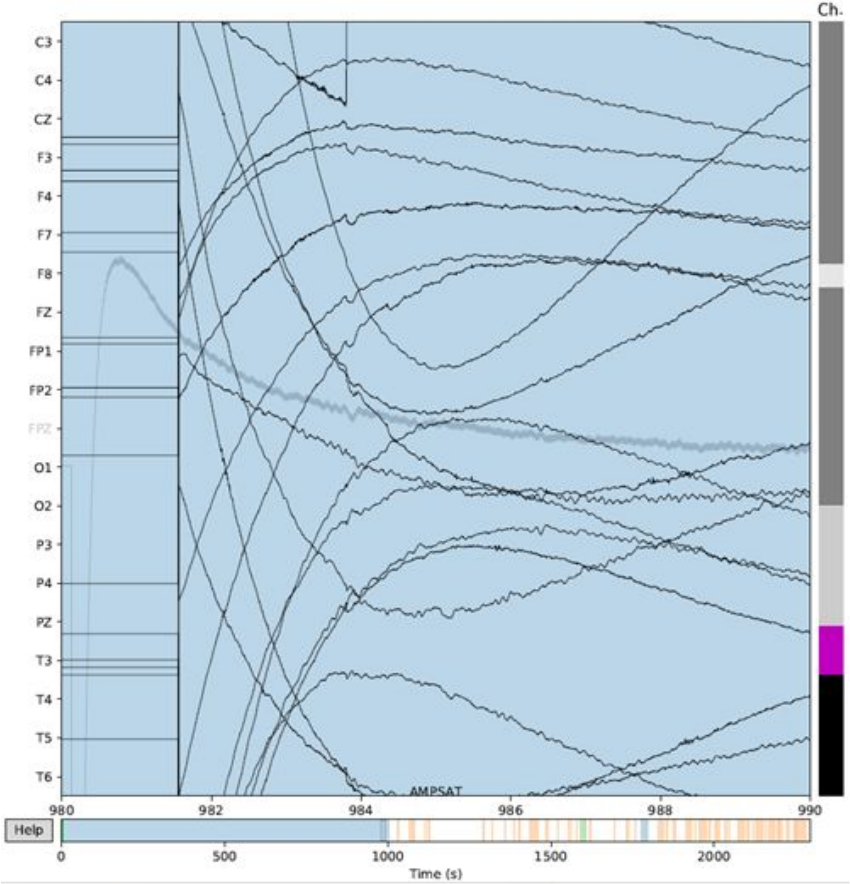
All data that was not marked represents interictal EEG with no content of interest



Pre-processing

AMPSAT (27.41%)

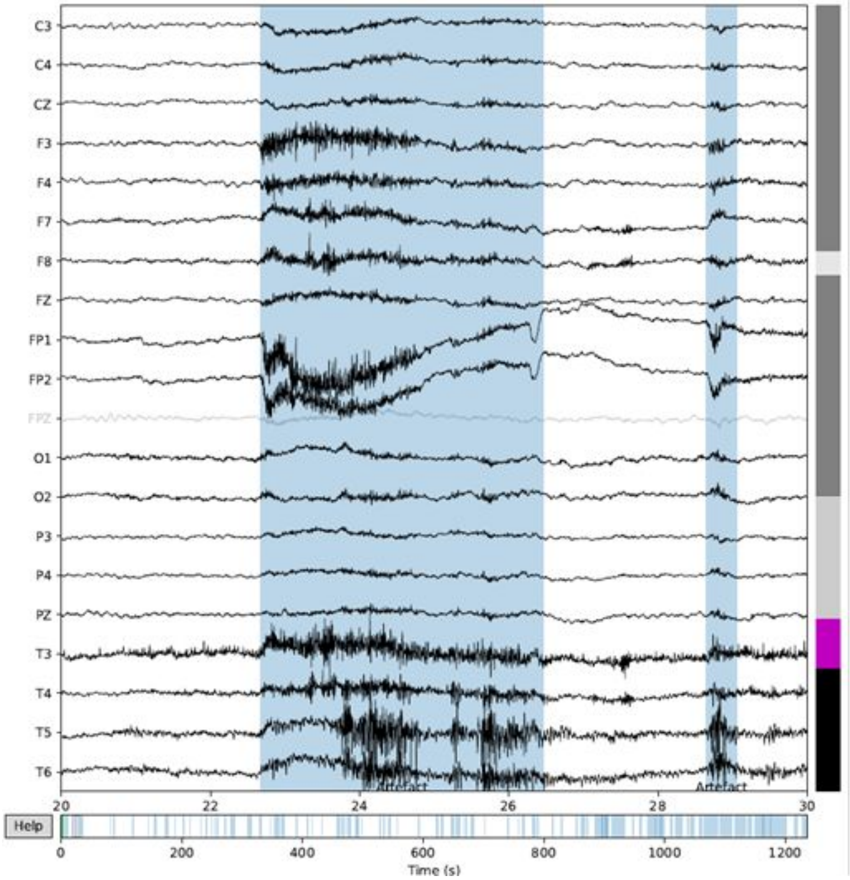
Segments with amplifier saturation, mostly at the start of the recordings where the signals data quality is being improved



Pre-processing

Artefact (22.96%)

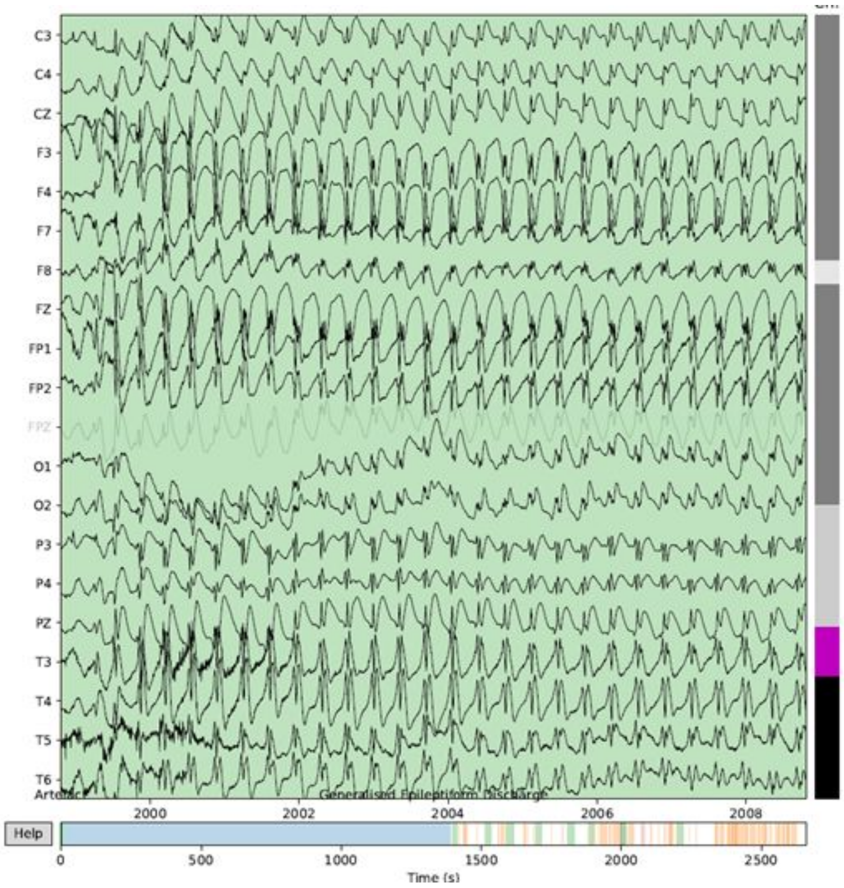
Electrical phenomena which distorts the neural signal such as respiratory, eye movement, muscle, or environmental sources



Pre-processing

Generalized Epileptiform Discharge (1.45%)

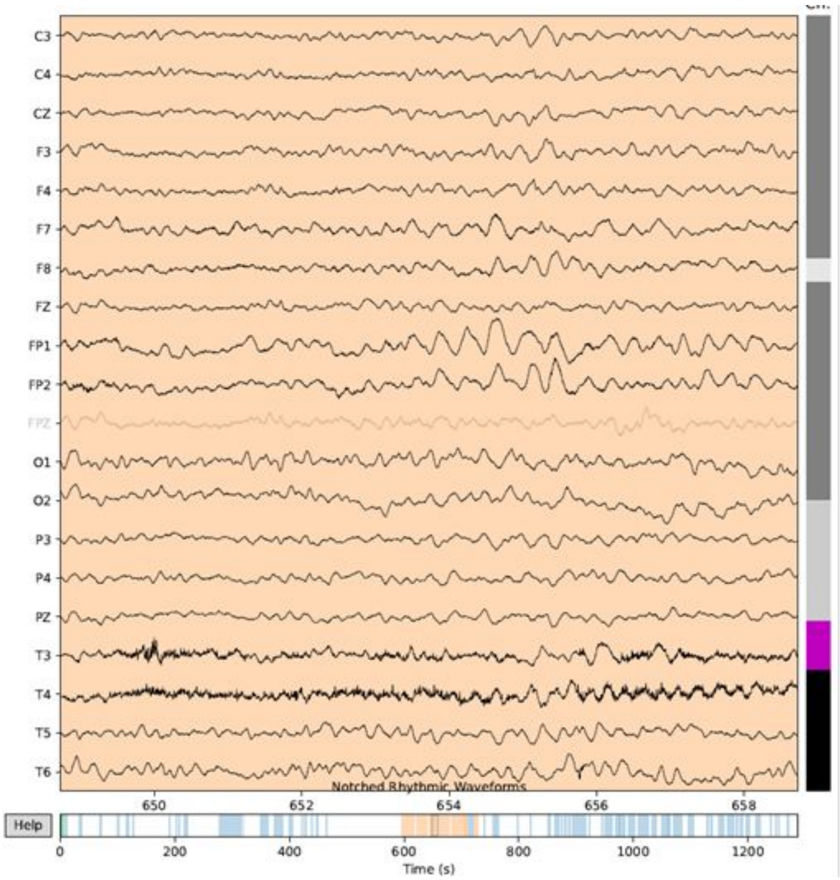
Spike-and-wave discharges which are sometimes proceeded by polyspikes



Pre-processing

Notched Rhythmic Waveforms (0.54%)

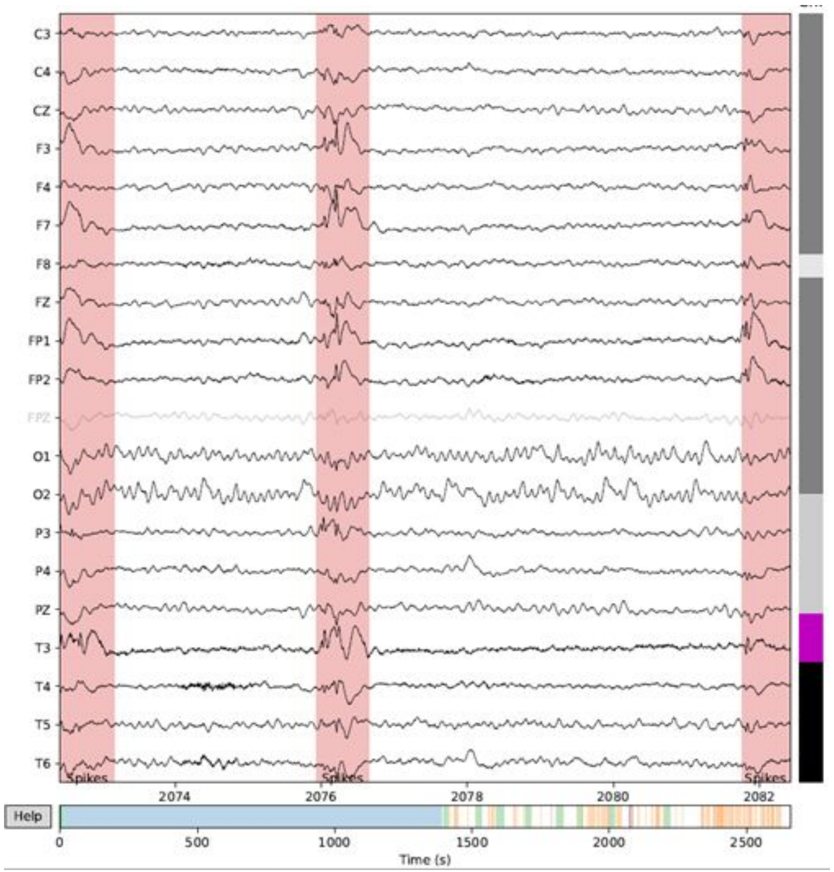
Benign activity likely a result of the patient being in a state of drowsiness



Pre-processing

Spikes (0.04%)

Events that in isolation would be unlikely to be used as a diagnostic marker



Pre-processing

Binary

Ictal

- Generalised Epileptiform Discharge

Inter-ictal

- Baseline
- Artefact
- Notched Rhythmic Waveforms
- Spikes

Multiclass

Ictal

- Generalised Epileptiform Discharge

Inter-ictal

- Baseline
- Notched Rhythmic Waveforms
- Spikes

Artefact

Feature Extraction

Quantify values or features of the signal (e.g. biomarkers or artefacts)

The data was epoched into window sizes of 2 seconds with a 1 second overlap (most records were sampled at 256Hz). In each epoch, for each channel, the following features were extracted…

Time	**Frequency**	**Time-Frequency**
Correlation Coefficients	Correlation Coefficients	Mean
Eigenvalues	Eigenvalues	Standard Deviation
	Median Power	Log Sum
	Relative Power	Mean Absolute
		Ratio

To get all these features for the full dataset (11hrs) takes around 6:03 mins on my laptop

Dimension Reduction

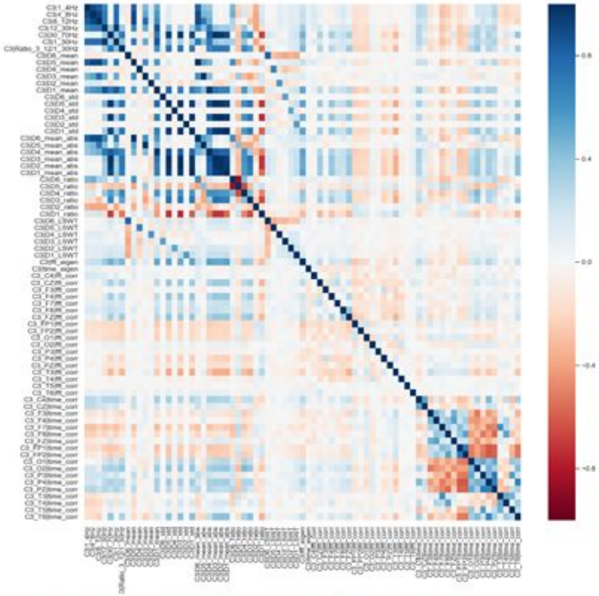
Principal Component Analysis (PCA)

Dimensionality reduction algorithms remove multicollinearity and retain important information by creating new synthetic features through combining features

PCA aims to separate a set of mixed signals into their component sources.

PCA aims to find vectors that best explain a data's variability by transforming data onto an equal or lower dimensional subspace, combining features that are highly correlated.

PCA can reduce a model's complexity, run time, and potential for overfitting to the training data.

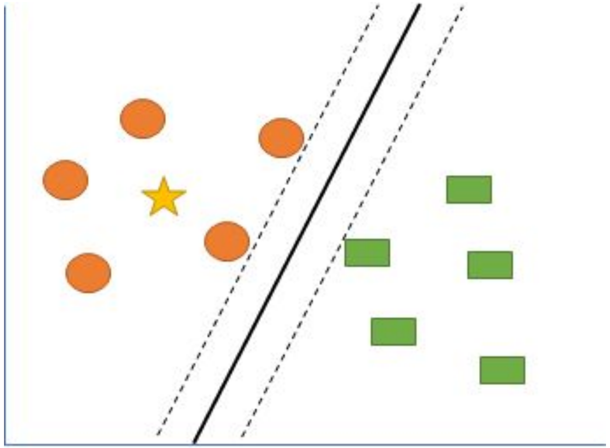


Correlation heatmap of features involving channel C3 across the whole of P19's EEG session

Classification

SVMs are discriminative algorithms that distinguish classes of objects by finding a hyperplane that provides the maximum margin of separation from classes.

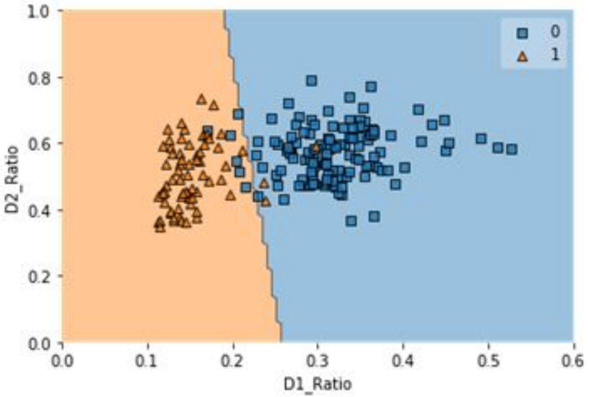
If data can be linearly separated, then a 'hard' margin of separation can be used; whereby a point on the edge of a class is used as the support vector for the decision boundary.



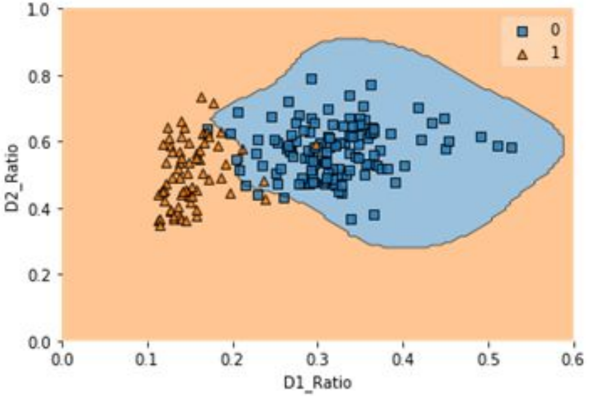
Classification

However 'hard' margins are sensitive to outliers, so often a 'soft' margin is used to allow for some errors (*C*).

If classes cannot be linearly separated, the input feature space can be projected to higher dimensions to create a nonlinear separation boundary



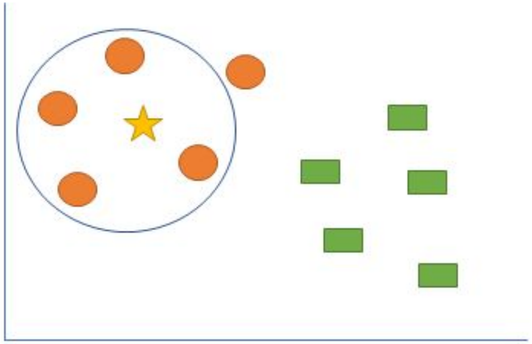

```
pipe_svc_rbf = Pipeline([('scl', StandardScaler()),
                         ('clf', SVC(C=100,
                                     kernel='rbf',
                                     class_weight = 'balanced',
                                     random_state=RANDOM_STATE))])

pipe_svc_rbf.fit(vis_data, y_train)

plot_decision_regions(vis_data,
                      y_train,
                      clf = pipe_svc_rbf)

plt.xlabel(x_axis_label)
plt.ylabel(y_axis_label)
plt.xlim(0,.6)
plt.ylim(0,1.)
plt.show()
```



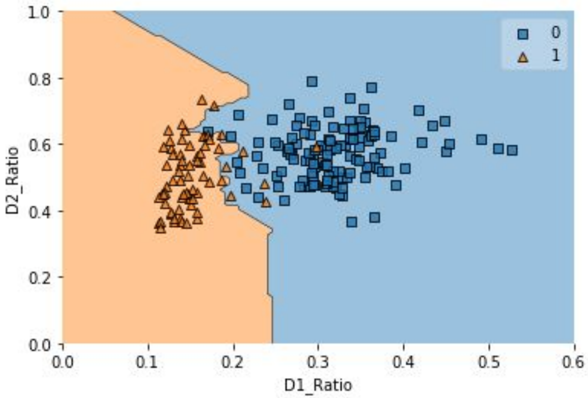
Classification

KNN is a lazy learner that memorizes training data, rather than learning a discriminative function, assigning data points to the class with the greatest number of "nearest neighbors.

The number of nearest neighbors (*k*) and a distance metric, to measure the distance between samples, need to be specified.



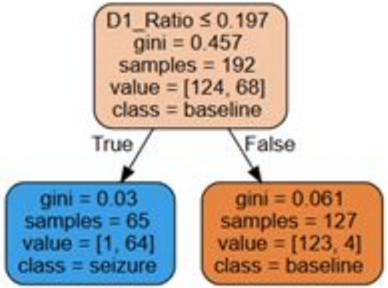
```
from sklearn.neighbors import KNeighborsClassifier
pipe_knn = Pipeline([('scl', StandardScaler()),
                     ('clf', KNeighborsClassifier(n_neighbors=2))])

pipe_knn.fit(vis_data, y_train)

plot_decision_regions(vis_data,
                      y_train,
                      clf = pipe_knn)

plt.xlabel(x_axis_label)
plt.ylabel(y_axis_label)
plt.xlim(0,.6)
plt.ylim(0,1.)
plt.show()
```

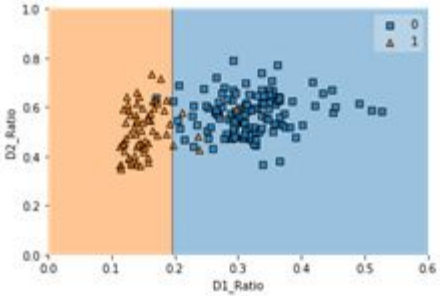


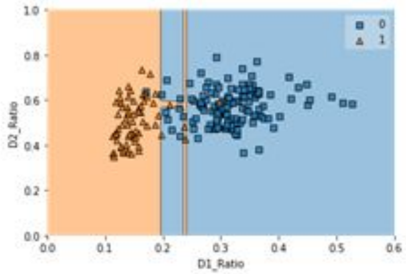
Classification

Decision trees split the data based on the features that best separates data into the class labels.

Data is split until all the samples within each node all belong to the same class or a maximum depth is reached.





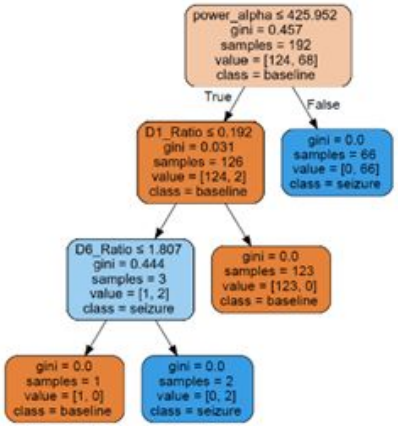


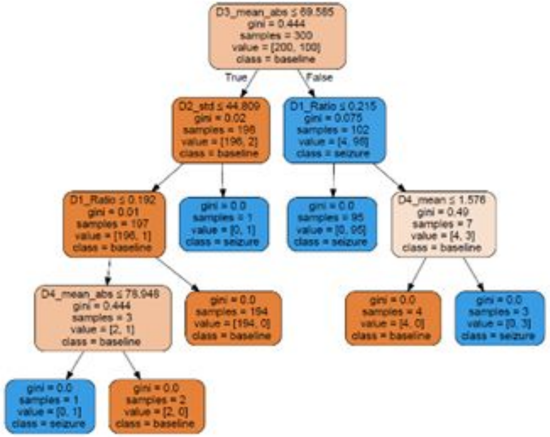

```
DT = DecisionTreeClassifier(criterion='gini',
                            max_depth = None,
                            random_state=RANDOM_STATE)
DT.fit(vis_data, y_train)

dot_data = export_graphviz(DT, out_file=None,
                           feature_names=[x_axis_label, y_axis_label],
                           class_names=feature_reduced['class'].unique(),
                           filled=True, rounded=True,
                           special_characters=True)

graphviz.Source(dot_data)
```

Classification

A RF is an ensemble of multiple decision trees which are averaged together.

A random forest draws a random bootstrap sample of data and features to grow individual decision trees on. This process is repeated *n* times and the prediction of each tree is aggregated to assign class labels.





Optimisation and Cross-Validation

A Bayesian optimization method was used to search over classification pipeline components and model hyperparameters for each classifier. The search space begins with a random combination of components and hyperparameters, which are optimized over 1000 iterations.

Optimisation and Cross-Validation

The objective function is used at each iteration to update a prior from a history of model configuration and score pairs. The probability model P(score|configuration) is used to search for the most promising candidates and is therefore quicker than evaluating all possible combinations (e.g. GridSearch).

There are a few different algorithms for bayesian optimisation, such as gaussian processes and tree-structured-parzen-estimators.

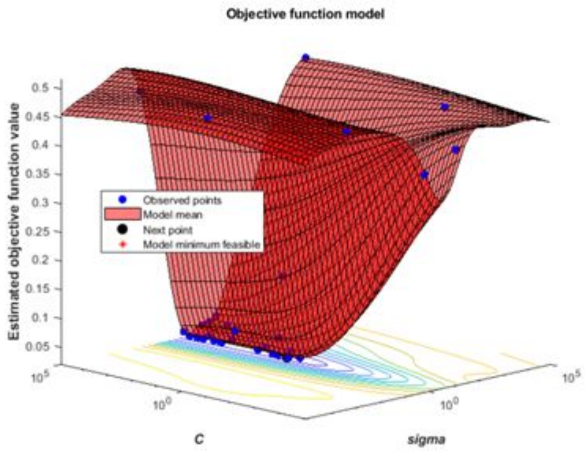


Fig. 3: SVM parameter optimization using Bayesian optimization algorithm.

Nandy, A., Alahe, M. A., Uddin, S. N., Alam, S., Nahid, A. A., & Awal, M. A. (2019, January). Feature Extraction and Classification of EEG Signals for Seizure Detection. In *2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)* (pp. 480-485). IEEE.

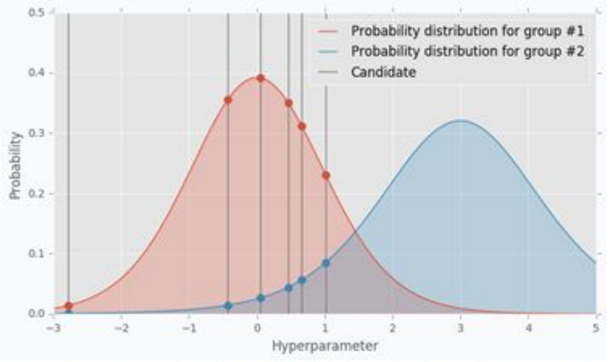
Optimisation and Cross-Validation

Tree of Parzen Estimators (TPE)

Gaussian Mixture Model's (GMM) or regression models can be used for modelling the probability.

For TPE's a prior distribution needs to be defined for the hyperparameters, although these can just be uniformly distributed if there is little previous guidance.

The first few iterations just perform a random search to build a distribution of the best observations for each hyperparameter. A GMM $l(x)$ is fitted to parameters associated with the smallest loss function values, and another GMM $g(x)$ to the remaining values to choose a parameter value x that maximizes the ratio $l(x)/g(x)$.



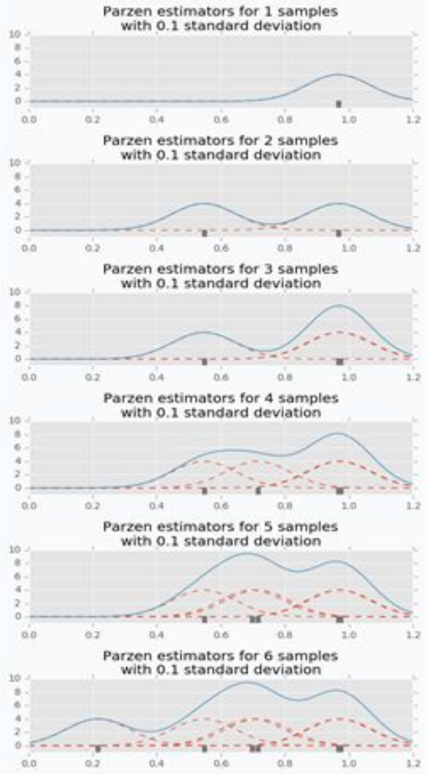
http://neupy.com/2016/12/17/hyperparameter_optimization_for_neural_networks.html

Optimisation and Cross-Validation

Tree of Parzen Estimators (TPE)

The distributions are modelled using parzen-window density estimators so that each sample defines a gaussian distribution which can be stacked together and normalised to give a probability density function.

The tree structure refers to the fact parameters can have tree-structured dependencies; for example, the Gamma parameter of a SVM can only be selected if the kernel is chosen to be a RBF rather than linear.



http://neupy.com/2016/12/17/hyperparameter_optimization_for_neural_networks.html

Optimisation and Cross-Validation

Tree of Parzen Estimators (TPE)

```python
from hyperopt import fmin, tpe, hp, STATUS_OK, Trials
from sklearn.preprocessing import StandardScaler

PARAM_DIST = {
  'C': hp.uniform('C', 0, 8),
  'kernel': hp.choice('kernel', [
      {'ktype': 'linear', 'gamma':'auto'}, # gamma ignored
      {'ktype': 'sigmoid', 'gamma': hp.uniform('sig_gamma', 0, 1)},
      {'ktype':'poly', 'gamma': hp.uniform('poly_gamma', 0, 1)},
      {'ktype': 'rbf','gamma': hp.uniform('rbf_gamma', 0, 1)}]),
  'scale': hp.choice('scale', [0, 1])
}

def hyperopt_train_test(params):
  X_ = X_train[:]

  if 'scale' in params:
    if params['scale'] == 1:
      sc = StandardScaler()
      X_ = sc.fit_transform(X_)

  clf = SVC(C = params['C'],
            kernel = params['kernel']['ktype'],
            gamma = params['kernel']['gamma'],
           )

  return cross_val_score(clf, X_, y_train, cv = 5).mean()

def objective(params):
  acc = hyperopt_train_test(params)
  return {'loss': -acc,           # minus because we need to reduce
          'status': STATUS_OK}

trials = Trials()
best = fmin(objective, PARAM_DIST,
            algo=tpe.suggest, max_evals=500,
            trials=trials)

print('best:')
print(best)
```

```
100%|██████████| 500/500 [01:02<00:00,  8.06it/s, best loss: -0.9875]
best:
{'C': 0.7576142466789729, 'kernel': 0, 'scale': 1}
```

Optimisation and Cross-Validation

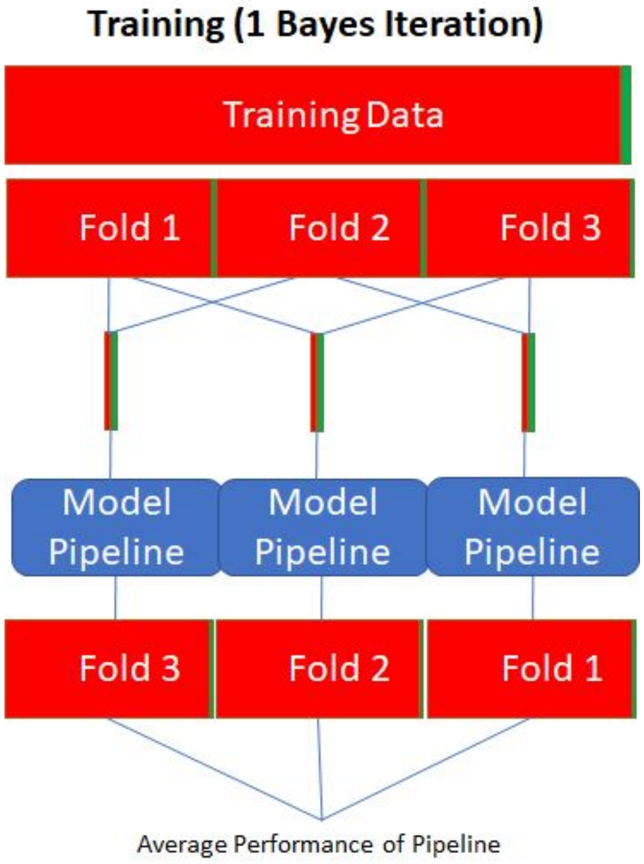
Pipeline Step	Algorithm	Hyperparameter	Parameter Space
Feature Selection	None	-	-
	SelectFromModel(RF)	Max Features	randint(1, 541)
Dimension Reduction	None	-	-
	PCA	Number of Components	uniform(0.05, 1.0)
Classification	Dummy Classifier	-	-
	KNN	Nearest Neighbors	randint(1,10)
		Algorithm	choice(ball tree, kd tree, brute)
		p	randint(1,10)
		Leaf Size	normal(m=30, sd=8)
	SVM	C	uniform(0.05, 8)
		Kernel	choice(linear, rbf)
		Gamma	uniform(0.005, 2)
	RF	Number of Estimators	normal(m=2000, sd=500)
		Criterion	choice(Gini, Entropy)
		Max Depth	choice(None, randint(1, 50))
		Min Samples Split	uniform(0.01, 1.)
		Max Features	uniform(0.01, 1.)

Optimisation and Cross-Validation

Pipelines were cross-validated using a 5-fold StratifiedKFold so each fold had a similar proportion of seizure and non-seizure data to the full data.

Each fold was undersampled to balance the number of ictal and interictal data for training.

Separately for each classifier, at each trial of the Bayesian optimization training, features could be selected using a random forest, extracted using PCA, or both.

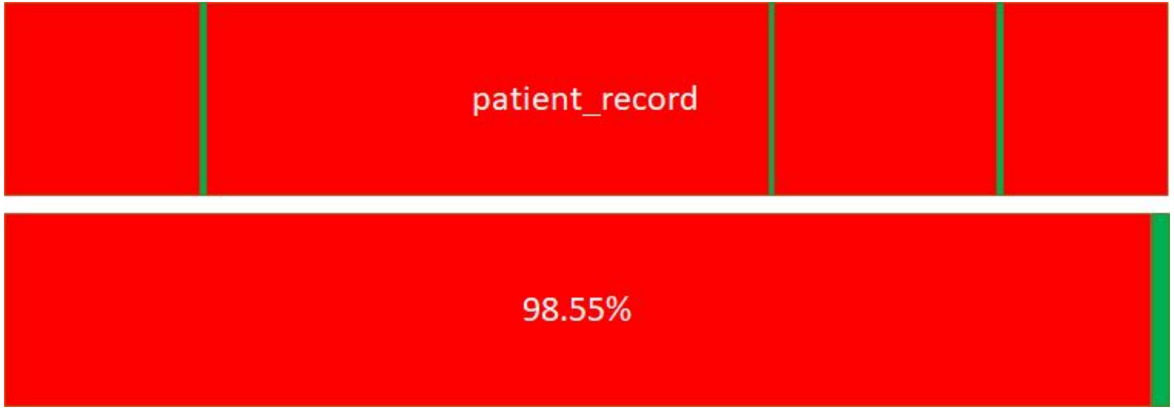


Optimisation and Cross-Validation

Question: How can I get 98.55% accuracy using 1 rule/line of code on a 20 minute EEG record where a patient has 3 seizures each lasting 5.8 seconds?

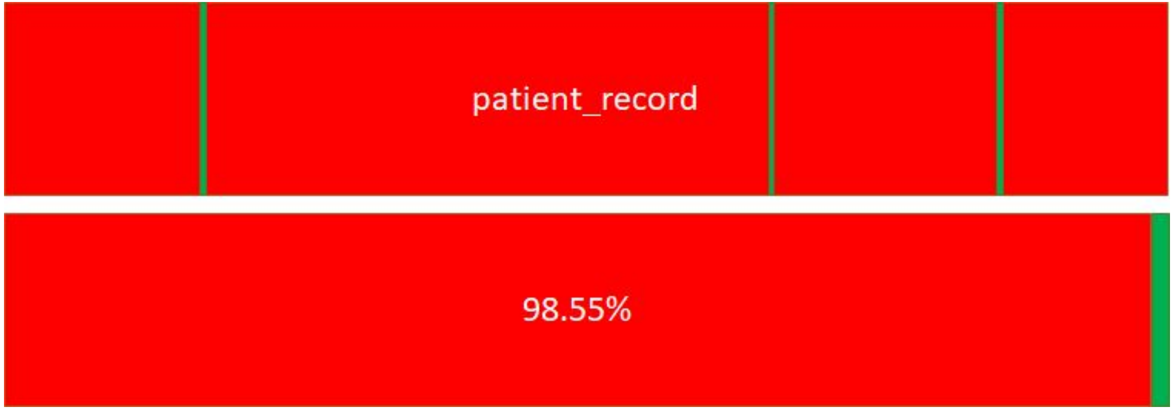
Optimisation and Cross-Validation

Question: How can I get 98.55% accuracy using 1 rule/line of code on a 20 minute EEG record where a patient has 3 seizures each lasting 5.8 seconds?



Optimisation and Cross-Validation

Question: How can I get 98.55% accuracy using 1 rule/line of code on a 20 minute EEG record where a patient has 3 seizures each lasting 5.8 seconds?



Answer: Just always predict a patient is never having a seizure

> *predictions = [0]*len(patient_record)*

(Also I find that weighting classes or up-sampling tends to perform worse as the imbalance is huge in this case)

Performance Evaluation

Patient-specific leave-one-out cross-validation

- Performance is assessed in a manner that is like how the models would be used in practice

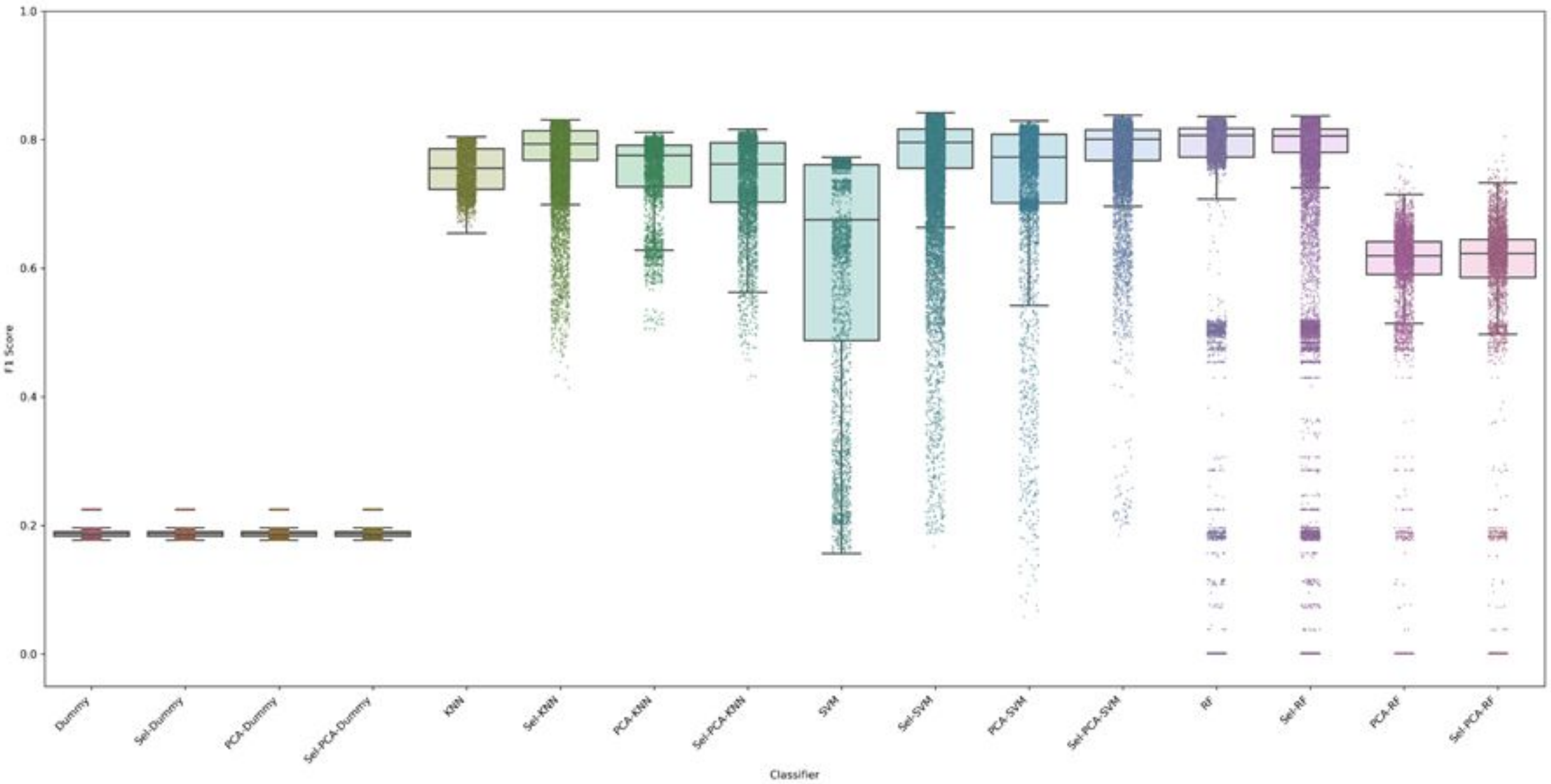
P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21
P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21
P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21
										…										
P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21

Performance Evaluation

The best pipelines for each classifier (SVM, RF, KNN) on each held-out dataset were selected and re-trained on an undersample of the full held-out dataset it was previously cross-validated on. This is because during cross-validation it was only trained on 4/5th of data.

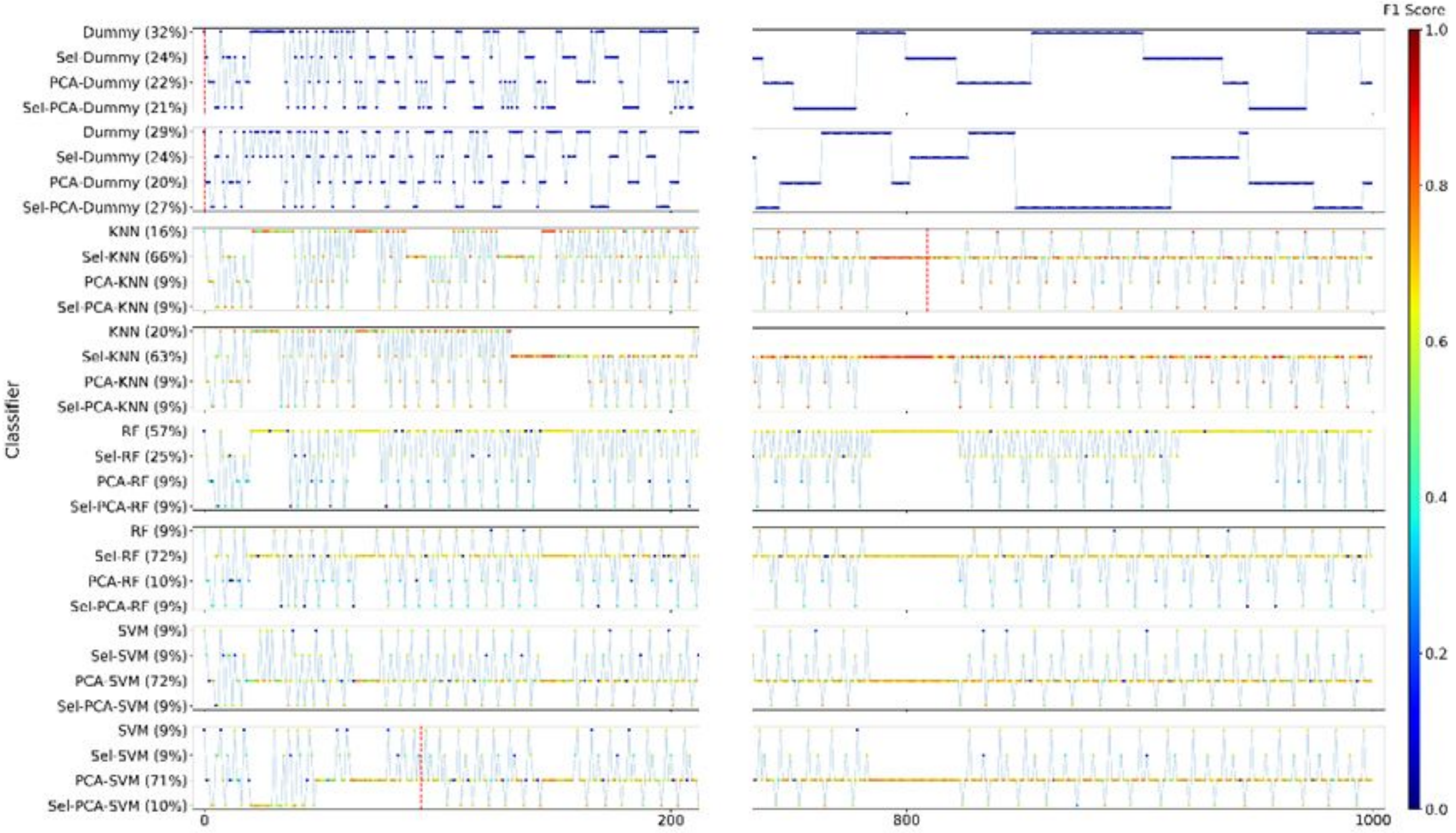
These models were then also grouped into a soft voting ensembles (SVE's) for each patient.

Training



1.0
0.8
0.6
0.4
0.2
0.0
F1 Score
Dummy
Sel-Dummy
PCA-Dummy
Sel-PCA-Dummy
KNN
Sel-KNN
PCA-KNN
Sel-PCA-KNN
SVM
Sel-SVM
PCA-SVM
Sel-PCA-SVM
RF
Sel-RF
PCA-RF
Sel-PCA-RF
Classifier

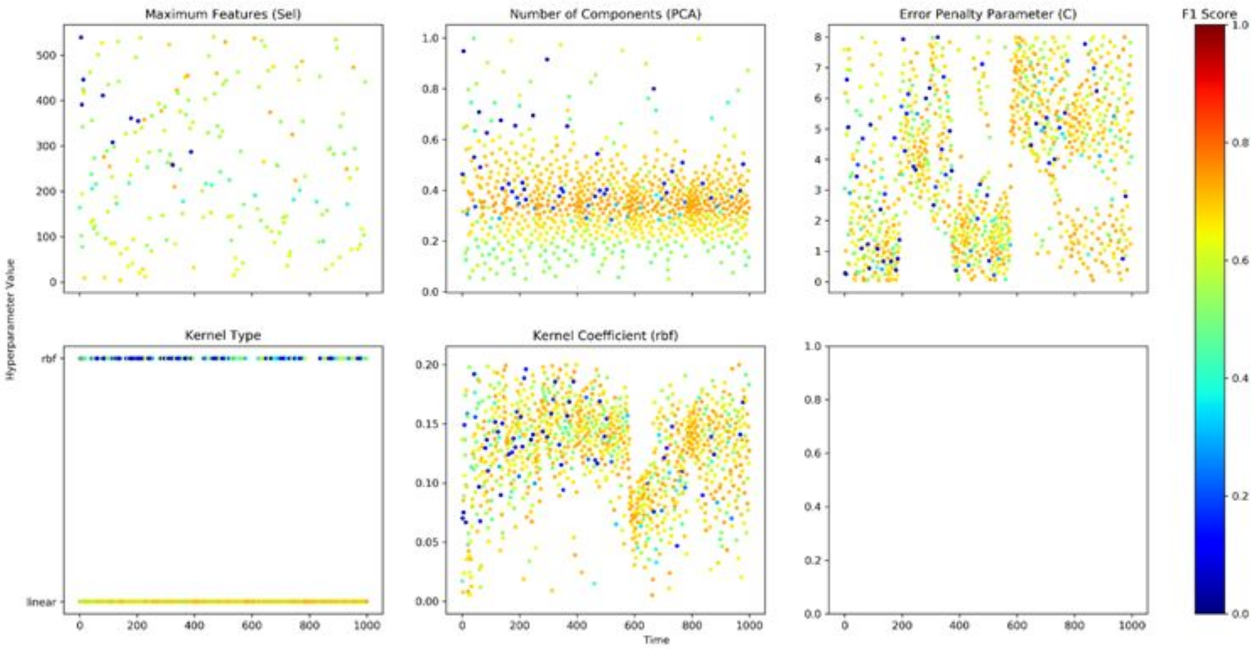
Training



F1 Score
1.0
0.8
0.6
0.4
0.2
0.0
Classifier
Dummy (32%)
Sel-Dummy (24%)
PCA-Dummy (22%)
Sel-PCA-Dummy (21%)
Dummy (29%)
Sel-Dummy (24%)
PCA-Dummy (20%)
Sel-PCA-Dummy (27%)
KNN (16%)
Sel-KNN (66%)
PCA-KNN (9%)
Sel-PCA-KNN (9%)
KNN (20%)
Sel-KNN (63%)
PCA-KNN (9%)
Sel-PCA-KNN (9%)
RF (57%)
Sel-RF (25%)
PCA-RF (9%)
Sel-PCA-RF (9%)
RF (9%)
Sel-RF (72%)
PCA-RF (10%)
Sel-PCA-RF (9%)
SVM (9%)
Sel-SVM (9%)
PCA-SVM (72%)
Sel-PCA-SVM (9%)
SVM (9%)
Sel-SVM (9%)
PCA-SVM (71%)
Sel-PCA-SVM (10%)
0
200
800
1000

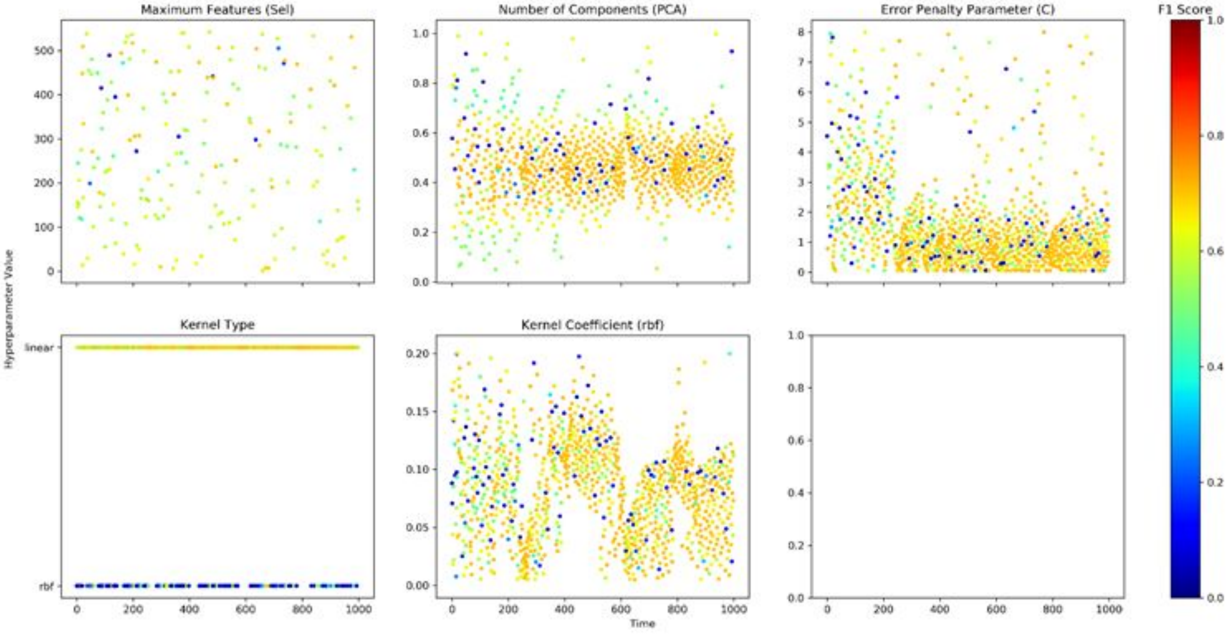
Training

SVM Hyperparameters on data where P2 was left out (random state = 1)

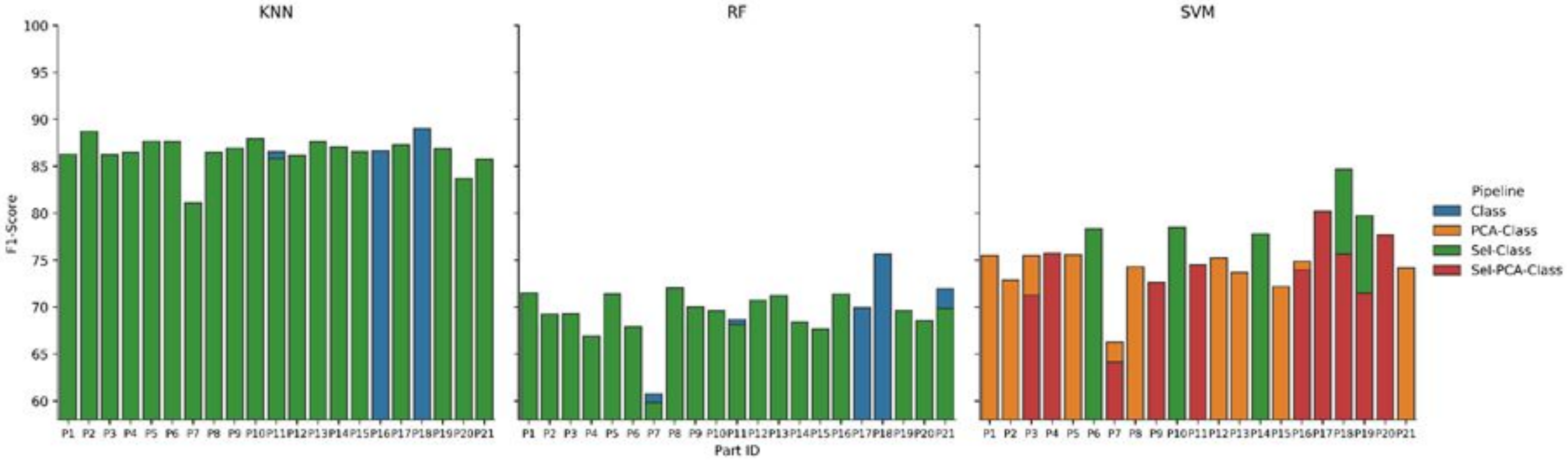


Training

SVM Hyperparameters on data where P2 was left out (random state = 2)

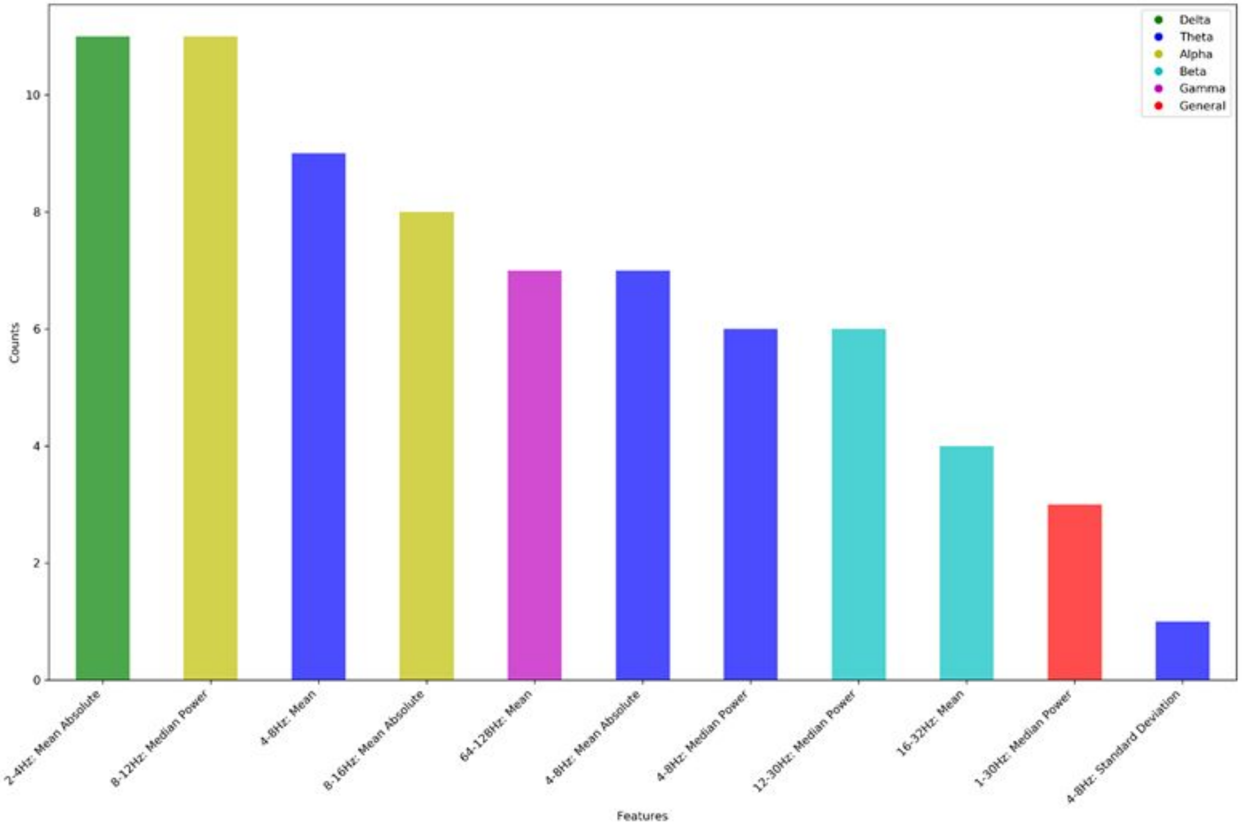


Training


KNN
RF
SVM
F1-Score
100
95
90
85
80
75
70
65
60
P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20 P21
Part ID
Pipeline
Class
PCA-Class
Sel-Class
Sel-PCA-Class

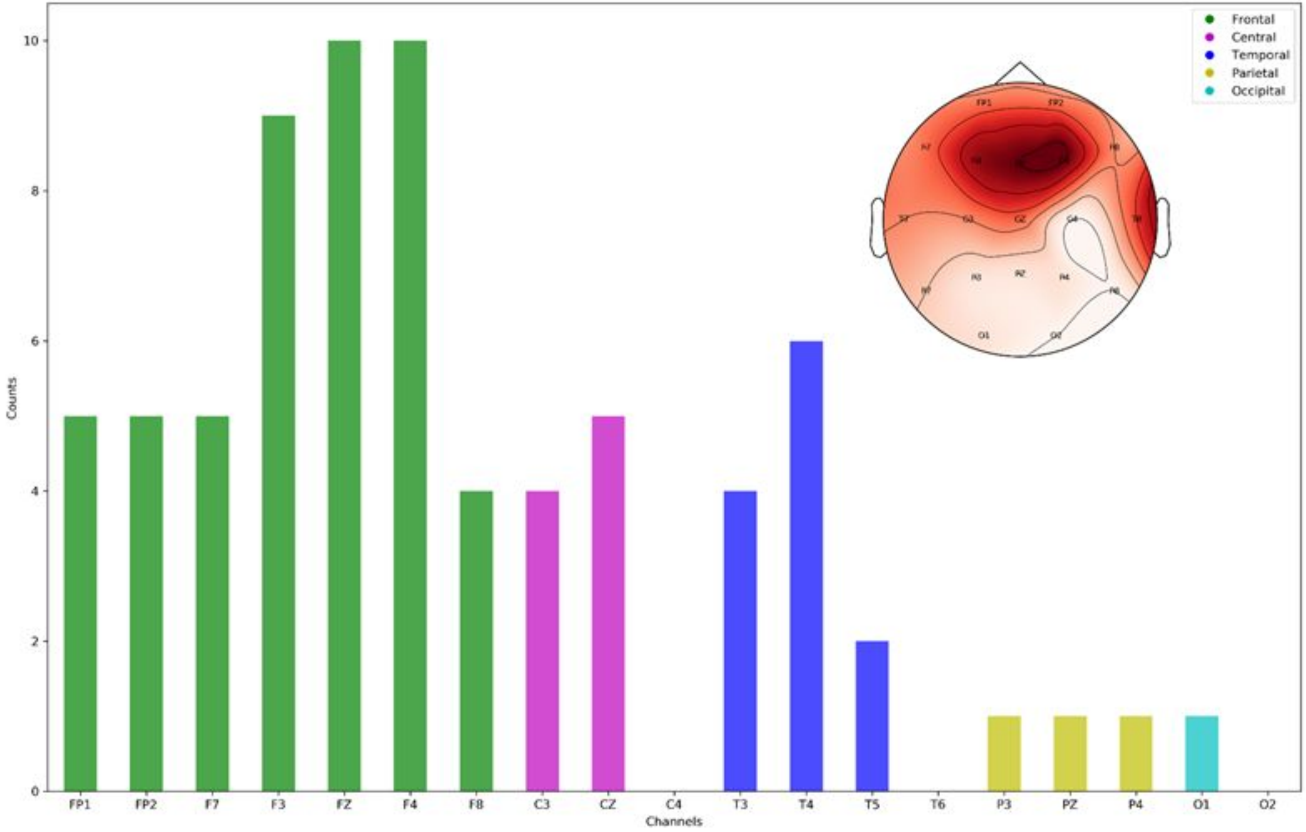
Training

Feature Selection

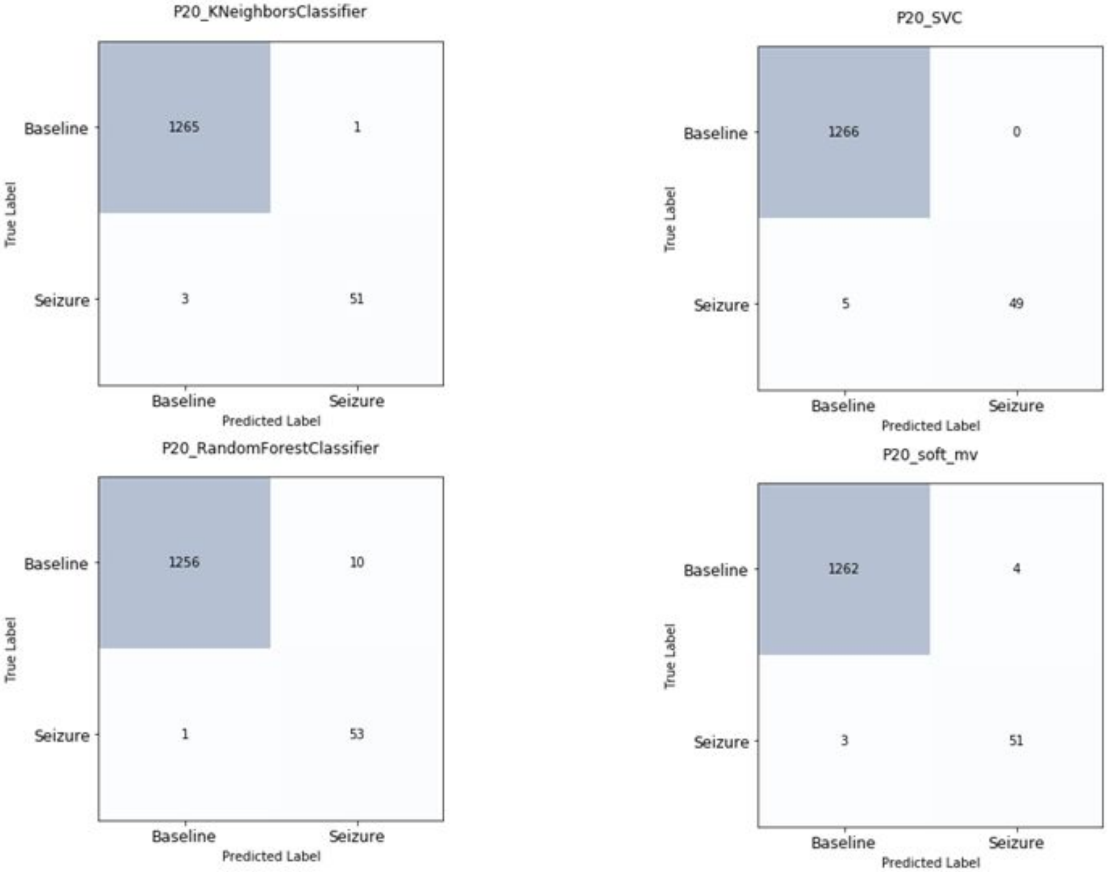


Training

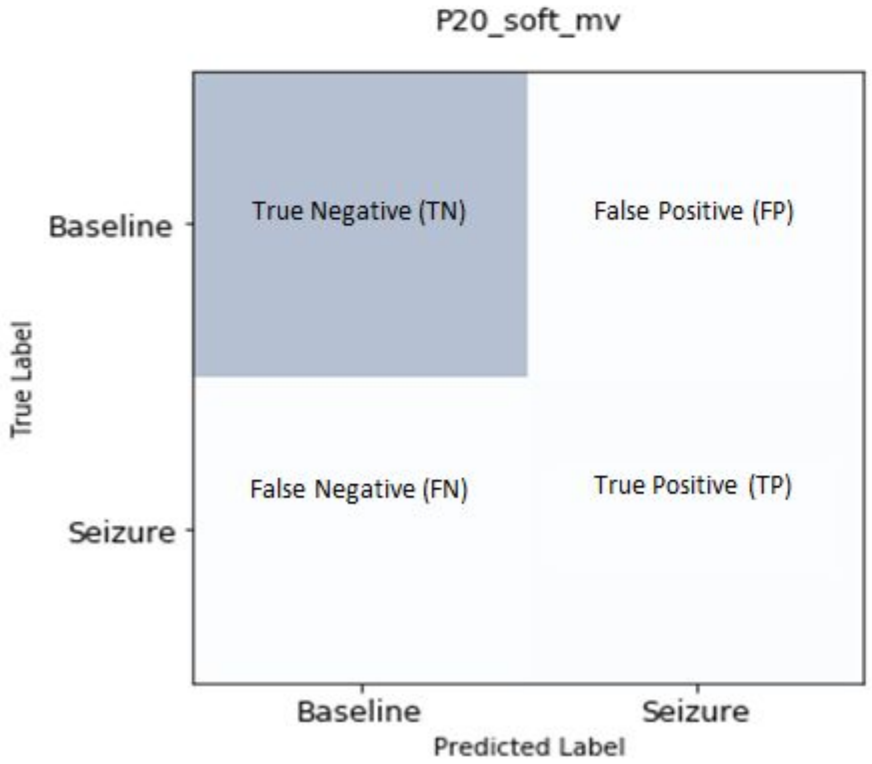
Feature Selection



Test Results



Test Results



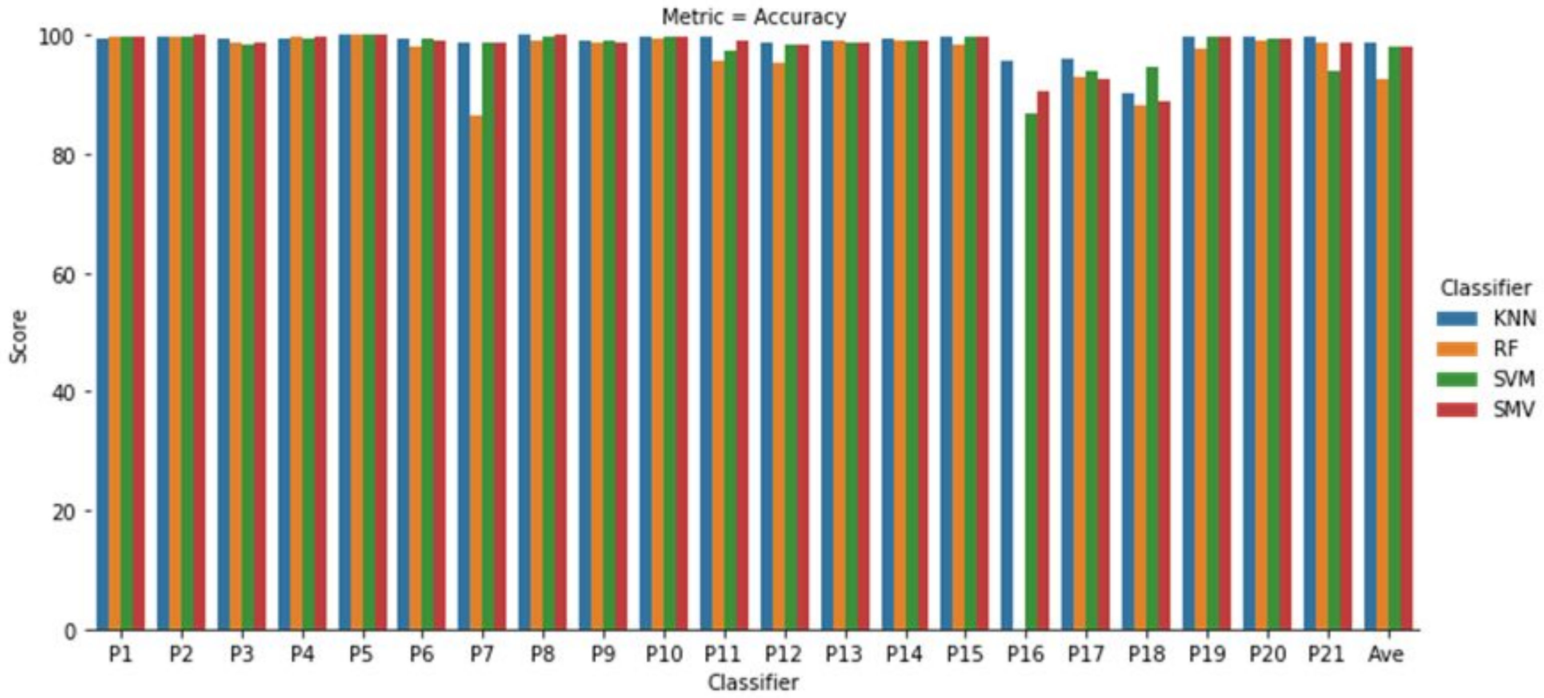
$$\text{Accuracy} = 1 - \frac{FP+FN}{FP+FN+TP+TN}$$

$$\text{Sensitivity} = \frac{TP}{TP+FN}$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

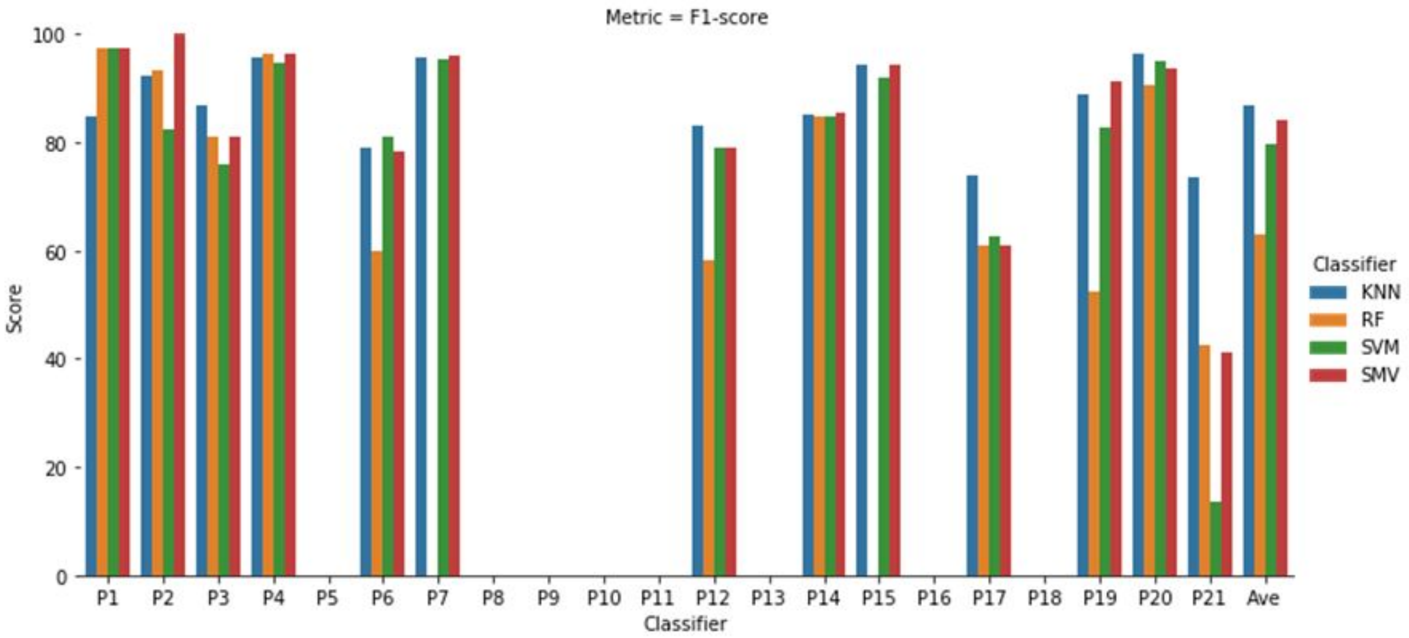
$$\text{F1} = 2 \times \frac{\text{Sensitivity} \times \text{Precision}}{\text{Sensitivity} + \text{Precision}}$$

Test Results



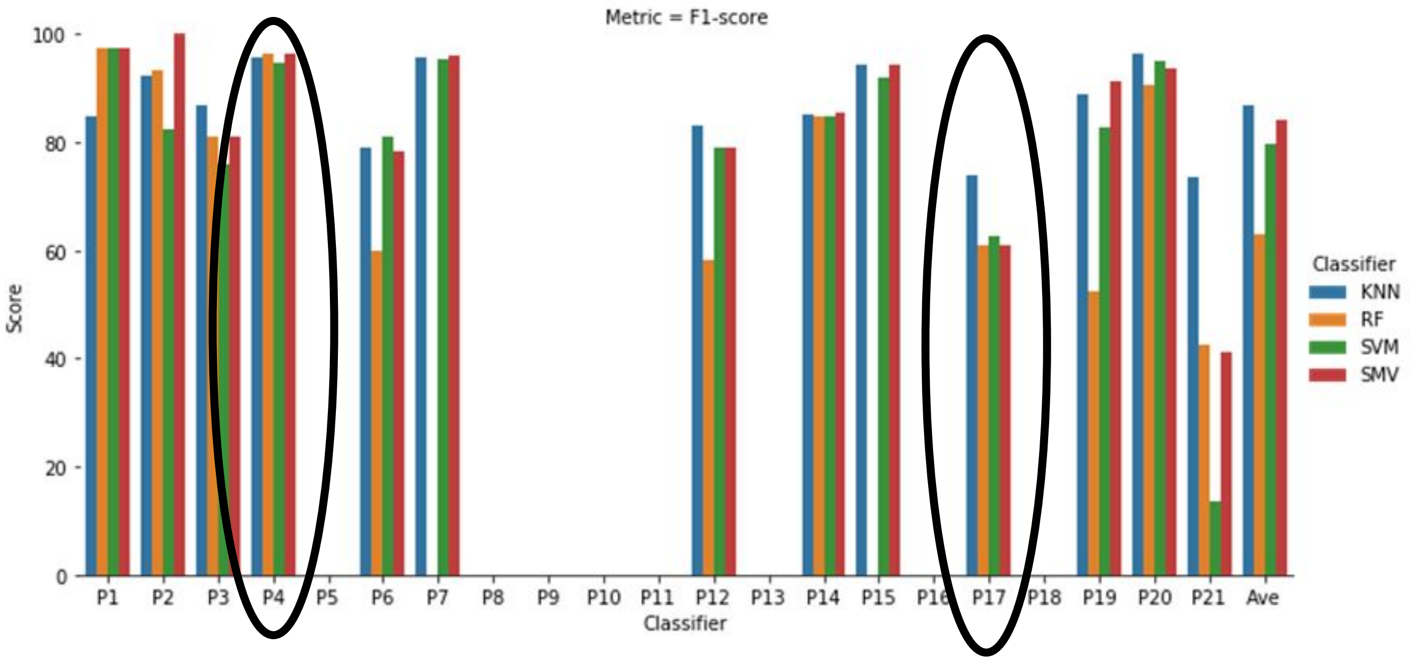
$$\text{Accuracy} = 1 - \frac{FP+FN}{FP+FN+TP+TN}$$

Test Results



$$\text{Sensitivity} = \frac{TP}{TP+FN} \qquad \text{Precision} = \frac{TP}{TP+FP} \qquad F1 = 2 \times \frac{\text{Sensitivity} \times \text{Precision}}{\text{Sensitivity} + \text{Precision}}$$

Test Results

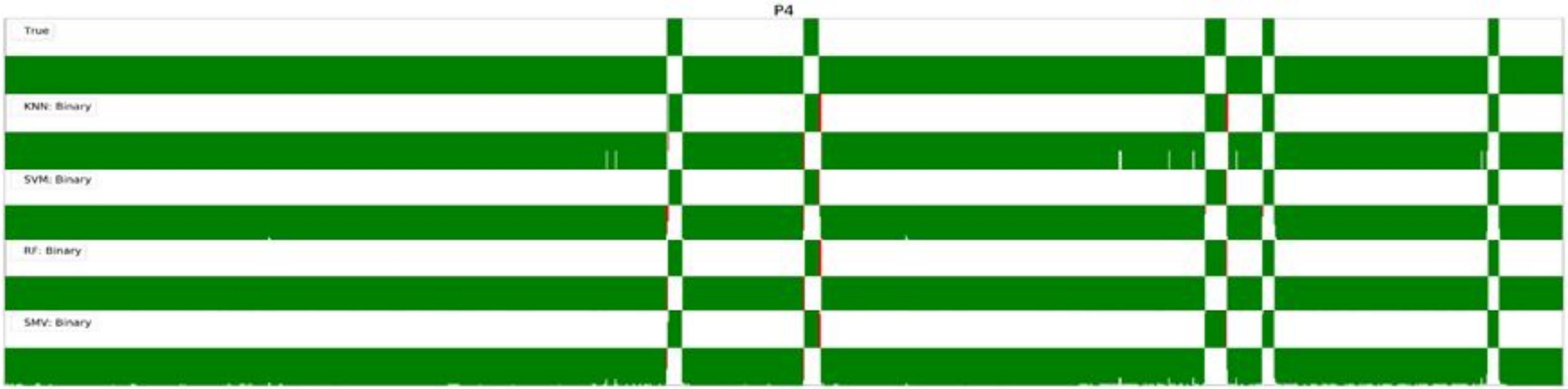


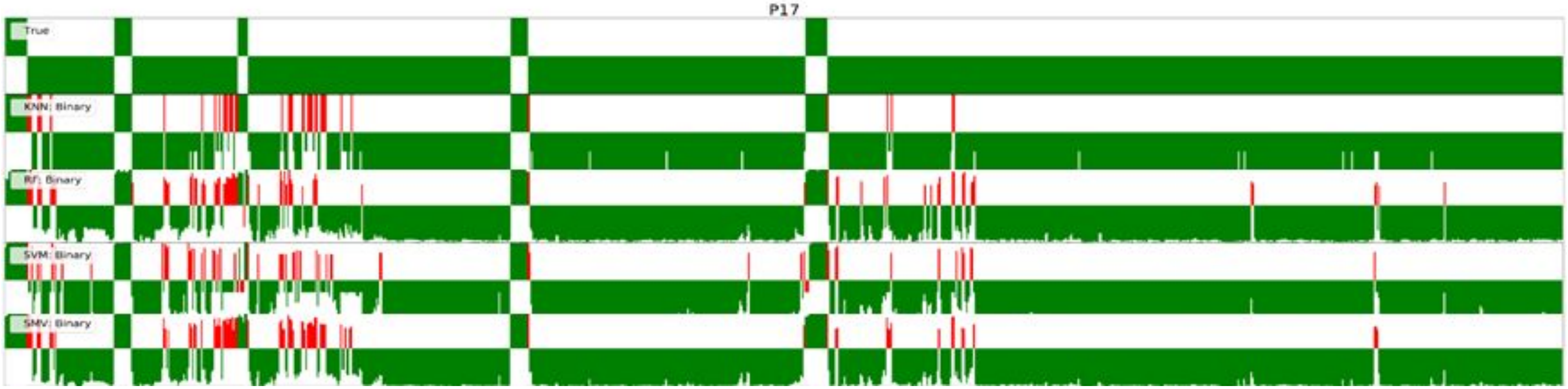
$$\text{Sensitivity} = \frac{TP}{TP+FN}$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$F1 = 2 \times \frac{\text{Sensitivity} \times \text{Precision}}{\text{Sensitivity} + \text{Precision}}$$

Test Results





Test Results

Reference	Participants	Data Length	Seizures	Channel number	Features	Label Classification	Classifier	Evaluation Method	ACC	SEN	SPEC	PREC	F1	FPR/h
Alkan2005	5 In-Clinic Patients 7 Controls	- -	20 N/A	4	3 Frequency	Binary	Logistic Regression Multilayer Perceptron	40% Hold-out	90.5 92	87.9 90	92.6 93.6	- -	- -	- -
Subasi2007	5 In-Clinic Patients 7 Controls	- -	20 N/A	128	4 Time-Frequency	Binary	Multilayer Perceptron Adaptive Neuro-Fuzzy Inference System	40% Hold-out	92 94	92 94.3	91.9 93.7	- -	- -	- -
Liang2010a	3 Rats	1288 secs	44%	1 (Inter-cranial)	2 Time 1 Frequency	Binary	Linear Least Squares Linear Discriminate Analysis Backpropagation Neural Network Support Vector Machine	Leave-One-Out Cross-Validation	95.39 95.81 97.37 97.5	90.33 91.37 96.8 97.03	99.07 98.93 97.83 97.83	- - - -	- - - -	- - - -
Petersen2011	19 In-Clinic Patients	11hrs 48m	111	18 (F7-FP1 reported)	1 Time-Frequency	Binary	Support Vector Machine	Leave-One-Out Cross-Validation	-	99.1	-	94.8	-	0.5
Duun-Henriksen2012a	20 In-Clinic Patients 1 Ambulatory Patient	11hrs 23m 4 Days	125 -	19 4 (F7-FP1 reported)	2 Time 1 Time-Frequency	Binary	Support Vector Machine	Triple-Repeated Fivefold Cross-Validation 1 Day Training, 3 Test	- -	97.2 95	- -	- -	- -	0 0.037
Zeng2016	9 In-Clinic Patients	600 secs	33%	19	3 Time 1 Frequency 1 Time-Frequency	Multiclass (Ictal, Interictal, Pre-Ictal)	Decision Tree K-nearest Neighbor Discriminant Analysis Support Vector Machine	10-fold Cross-Validation	71.8 72.1 76.7 74.3	- - - -	- - - -	- - - -	- - - -	- - - -
Kjaer2017	6 Ambulatory Patients	96hrs	-	3 (F7-FP1 reported)	1 Time 4 Frequency 1 Time-Frequency 1 Phase	Binary	Support Vector Machine	5-fold Cross-Validation	-	98.4	100	87.1	-	0.23
Our Approach	21 In-Clinic Patients	11hrs	53	19	2 Time 4 Frequency 5 Time-Frequency	Binary	K-Nearest Neighbors Random Forest Support Vector Machine Soft Majority Vote	Leave-One-Out Cross-Validation	99.39 93.44 98.93 98.95	88.14 80.39 89.07 93.85	99.55 94.22 99.11 99.05	93.78 65.15 86.57 87.73	90.36 70.64 85.28 89.69	15.77 207.49 31.65 33.63
						Multiclass (Ictal, Interictal, Artefact)	K-Nearest Neighbors Random Forest Support Vector Machine Soft Majority Vote	Leave-One-Out Cross-Validation	99.05 99.11 99.09 98.88	91.36 92.62 83.14 93.82	99.17 99.21 99.49 98.98	87.71 85.02 85.55 84.44	88.59 87.78 81.66 87.83	29.21 27.97 17.94 36.3

Test Results

ACC	SEN	SPEC	PREC	F1	FPR/h
90.5	87.9	92.6	-	-	-
92	90	93.6	-	-	-
92	92	91.9	-	-	-
94	94.3	93.7	-	-	-
95.39	90.33	99.07	-	-	-
95.81	91.37	98.93	-	-	-
97.37	96.8	97.83	-	-	-
97.5	97.03	97.83	-	-	-
-	99.1	-	94.8	-	0.5
-	97.2	-	-	-	0
-	95	-	-	-	0.037
71.8	-	-	-	-	-
72.1	-	-	-	-	-
76.7	-	-	-	-	-
74.3	-	-	-	-	-
-	98.4	100	87.1	-	0.23
99.39	88.14	99.55	93.78	90.36	15.77
93.44	80.39	94.22	65.15	70.64	207.49
98.93	89.07	99.11	86.57	85.28	31.65
98.95	93.85	99.05	87.73	89.69	33.63
99.05	91.36	99.17	87.71	88.59	29.21
99.11	92.62	99.21	85.02	87.78	27.97
99.09	83.14	99.49	85.55	81.66	17.94
98.88	93.82	98.98	84.44	87.83	36.3

General Conclusions

Datasets with lots of artifactual (noisy) data, means there's likely to be increased instances of false positives.

Some authors…

…group detection's together.

…remove artefacts before training and testing models.

…do not consider a detection false if within 1 minute of a seizure!

General Conclusions

Features selected by random forests reflect the presentation of absence seizures. This may enable…

…a seizure specific EEG channel profile based on the focal area of seizures.

…patient specific limited channel EEG for long term monitoring based on their unique seizure topography.

General Conclusions

Finding optimal parameters is important…

… for model fit

… to ensure differences between models are not because of default/selected parameters

General Conclusions

Model performance tends to be worse on records with no seizures present.

- Algorithms could assist with collecting longer EEG records.
- Another study found 30% of children who had no clinically detected seizures in a standard recording procedure had them detected in 1hr EEG recordings.

General Conclusions

Use of Bayesian hyperparameter optimisation

- Authors often do not make it clear how they arrived at certain hyperparameters

Most complete descriptions of the models performance in raw seconds

General Conclusions

Comparison of Binary and Multiclass models with a new dataset marked freely rather than in windowed bins

First use of NHS clinic data scans collected during diagnostic routine

- They tend to be noisy at the start of the record and while they are asking the patient to breathe heavily.

Whats Next?

Larger datasets

- NHS: Increase to 37 Patients (26hrs) with absence seizures
- TUH: 11 Patients (6hrs) with absence seizures
- TUH: 65 Patients (47hrs) with generalised seizures
- CHB: 24 Patients (980hrs) with generalised seizures

Ensemble Models

- Balanced Bagged KNN
- Balanced Random Forest
- Randomly Undersampled Boosted Trees (RUSBoost)
- LightGBM

Whats Next?

Deep learning architectures

- Multilayer Perceptron (MLP)
- Convolutional Neural Network (CNN)
- Recurrent Neural Network

BOHB – Bayesian Optimization and Hyperband

Varying window sizes based on data (Changepoint)

Computing hardware and software

Hardware

Dell XPS 13 9370 laptop

Lancaster High End Computing Cluster

Google Colab (GPU & TPU)

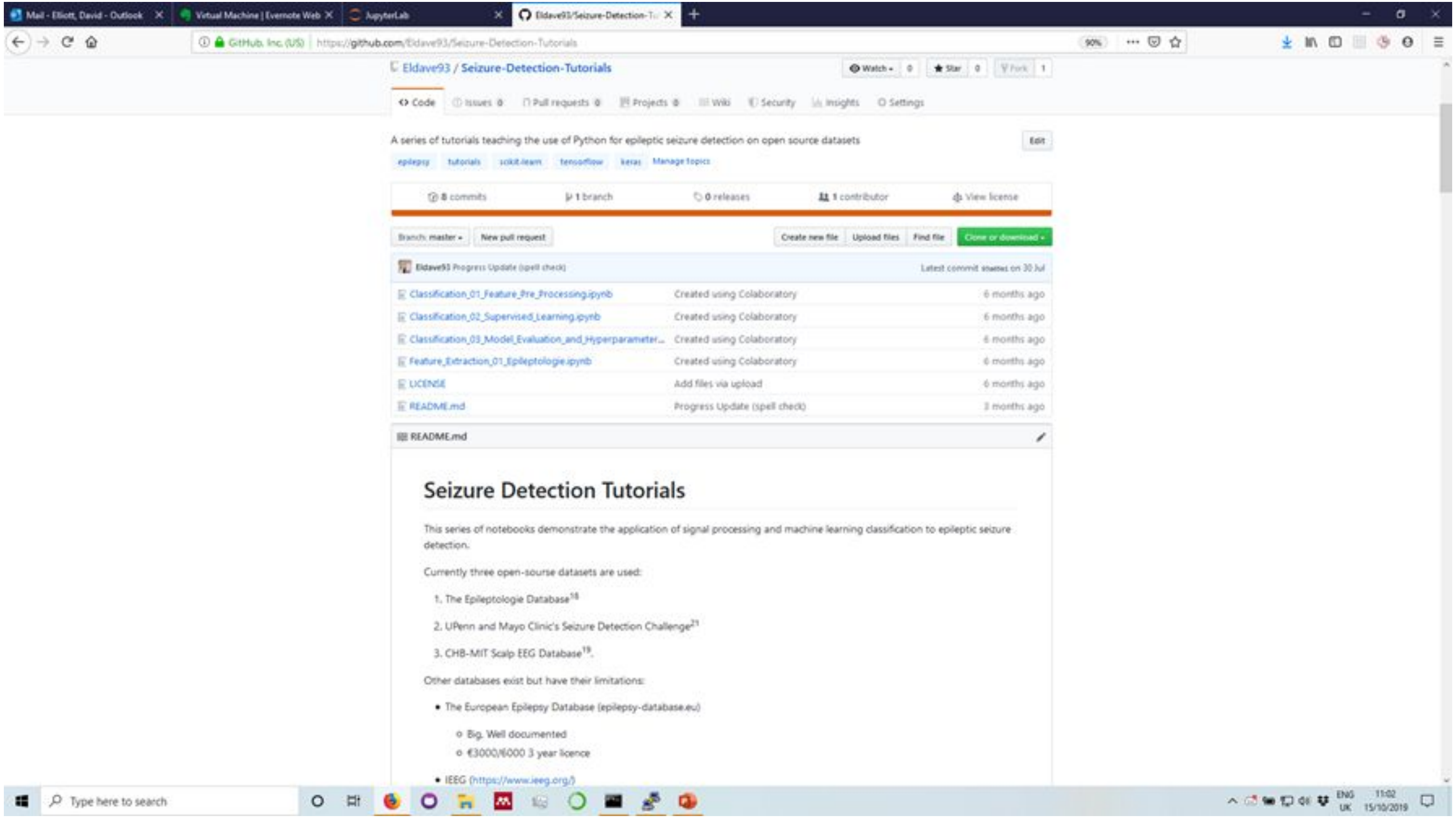
Software

Python 3

- Numpy, Pandas
- PyWavelets, SciPy
- Scikit-learn
- Imbalanced-learn
- Hyperopt, HpBandSter
- LightGBM
- Tensorflow 2.0 (Keras API)

Jupyter Notebooks

GitHub



Eldave93 / Seizure-Detection-Tutorials
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Security
Insights
Settings
A series of tutorials teaching the use of Python for epileptic seizure detection on open source datasets
epilepsy
tutorials
scikit-learn
tensorflow
keras
Manage topics
8 commits
1 branch
0 releases
1 contributor
View license
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
Eldave93 Progress Update (spell check)
Classification_01_Feature_Pre_Processing.ipynb
Created using Colaboratory
6 months ago
Classification_02_Supervised_Learning.ipynb
Created using Colaboratory
6 months ago
Classification_03_Model_Evaluation_and_Hyperparameter...
Created using Colaboratory
6 months ago
Feature_Extraction_01_Epileptologie.ipynb
Created using Colaboratory
6 months ago
LICENSE
Add files via upload
6 months ago
README.md
Progress Update (spell check)
3 months ago
README.md
Seizure Detection Tutorials
This series of notebooks demonstrate the application of signal processing and machine learning classification to epileptic seizure detection.
Currently three open-sourse datasets are used:
1. The Epileptologie Database
2. UPenn and Mayo Clinic's Seizure Detection Challenge
3. CHB-MIT Scalp EEG Database
Other databases exist but have their limitations:
The European Epilepsy Database (epilepsy-database.eu)
Big, Well documented
€3000/6000 3 year licence
IEEG (https://www.ieeg.org/)
Type here to search

Project Team

Clinical Application

- Prof. Vincent Reid (*Professor of Psychology*)
  - David Elliott (*PhD Student)*
  - Aidan Moutrey (*Undergraduate Student*)
- Dr. Judith Lunn (*Lecturer in Medical School*)
- Dr. Christian DeGoede (*Consultant Paediatric Neurologist*)
- Dr. Munni Ray (*Consultant Paediatric Neurologist*)
- Dr. Rosemary Belderbos (*Consultant Paediatrician*)
- Staff from Preston, Blackburn, and Leeds NHS Hospitals
  - Dr. Nicholas Combes
  - Gemma Wilkinson
  - Andrew Lancaster
  - Heather Collier

Hardware

- Barrie Usherwood (*Electronics/Research Technician*)
- Dr. Peter Tovee (*Electronics/Research Technician*)

Software

- Dr. Abe Karnik (*Lecturer in Computing and Communications*)
  - Kristoffer Geyer (*PhD Student*)
  - Nathan Rutherford (*Undergraduate Student*)

Data Analysis

- Dr. Rebecca Killick (*Senior Lecturer Statistics*)
  - David Elliott (*PhD Student)*

Thanks to our sponsors!

MIRALIS

Relative

INSIGHT

Digital
Lancashire/