Predicting change points in multivariate time series

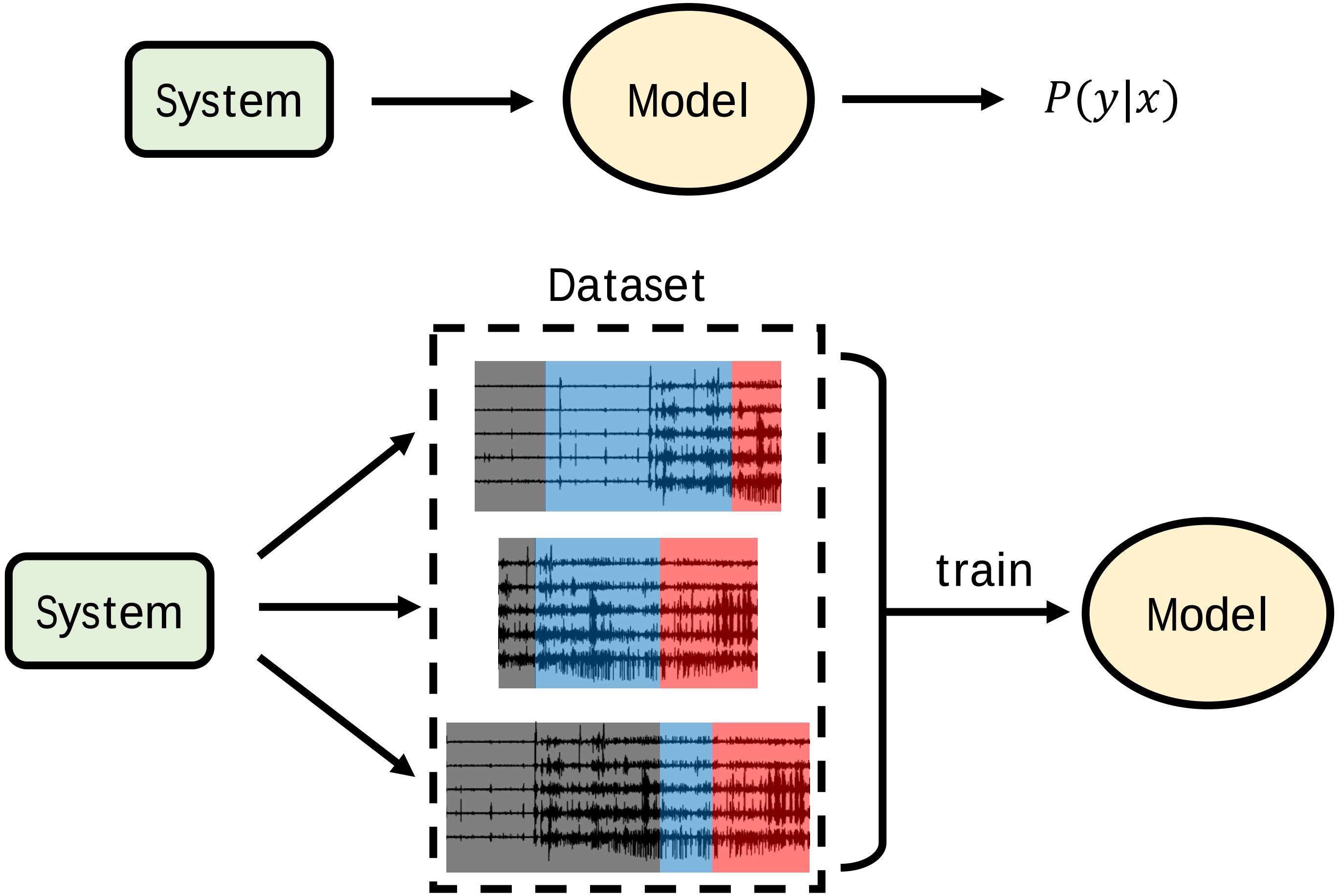
Haidar Khan

Overview

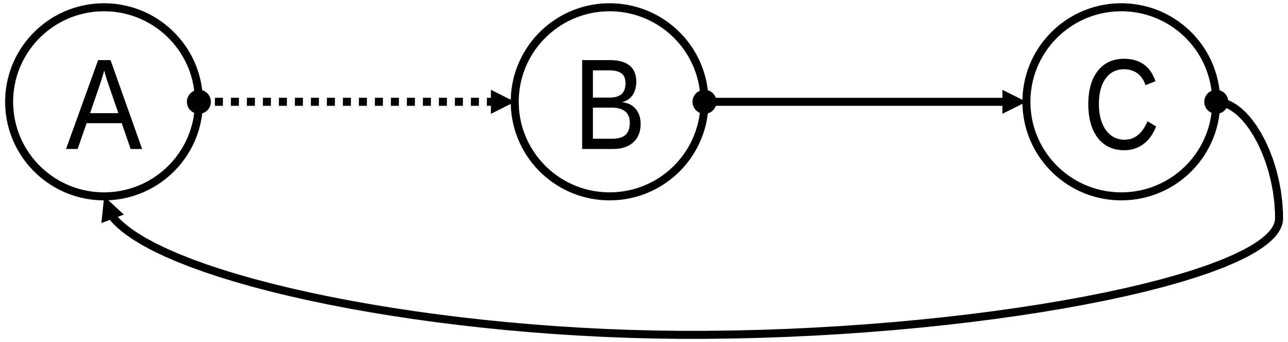
- Change points (what, why)
- Methods for CPD
- Epilepsy and seizures
- Deep virtual classifiers
- Learning spectrograms with wavelets

Setting and Motivation

- Predicting pre-seizure transition in patients with epilepsy*
- Computer network intrusion detection
- Machine failure prediction



High level problem



- Problem: Only some states labeled.
- Need to assign labels to the time series.

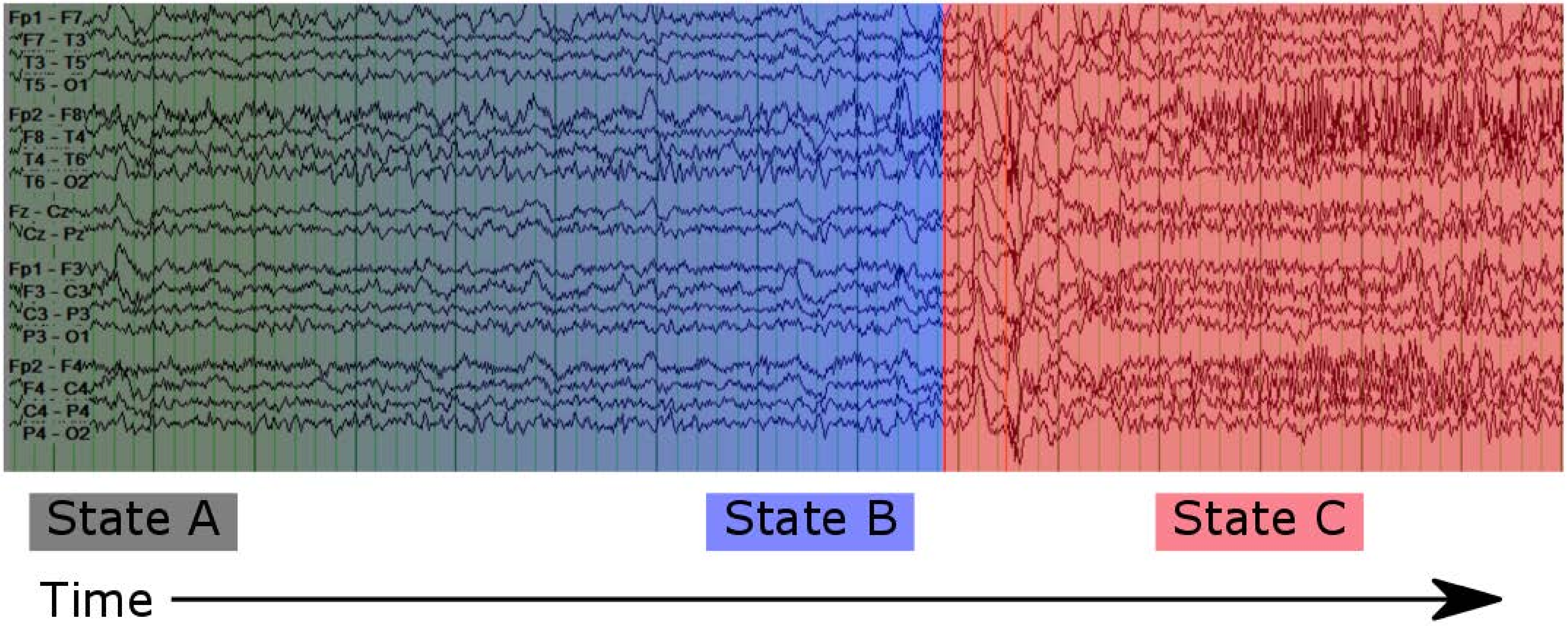
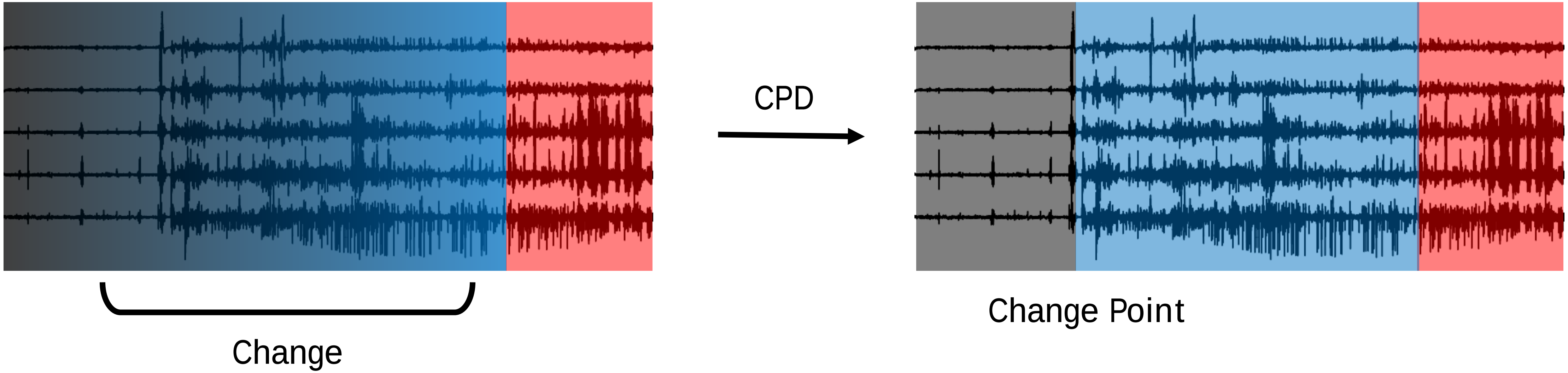
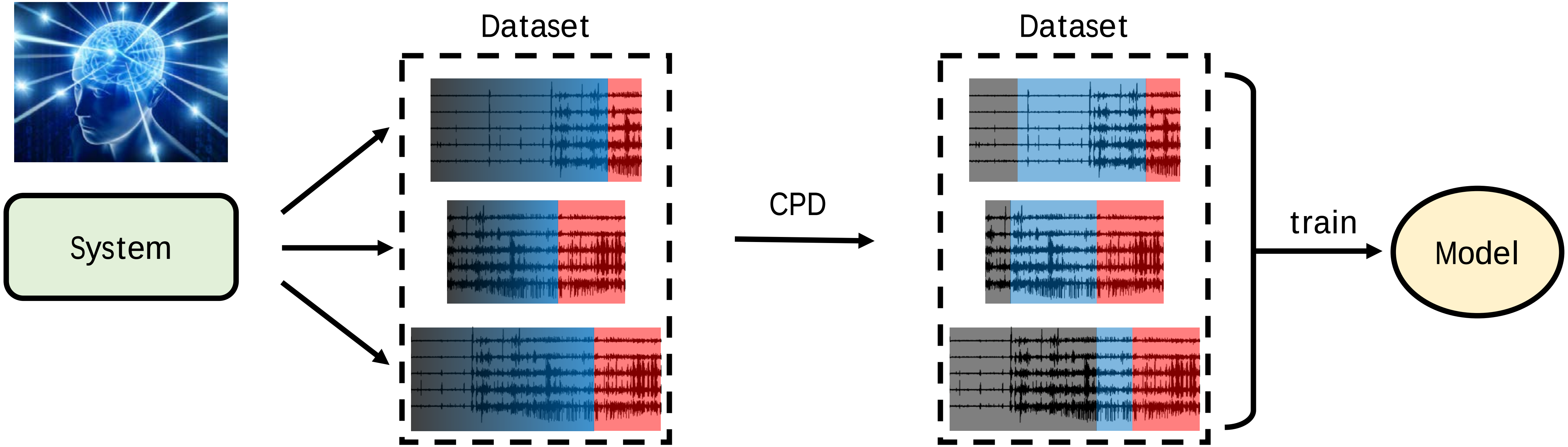


Figure 1. Example of a system generating a multichannel signal transitioning between states. The transition from State B to State C can be easily marked, but the transition from State A to State B cannot be marked. This results in a region of uncertainty about the state of the system.

Change point detection (CPD)

- Change point detection is determining **when** the transition occurs.





System
Dataset
CPD
Dataset
train
Model

Problem definition

- multivariate time series $\mathrm{X} = \{x_1, x_2, \dots x_T\}$
  - $x_t \in \mathbb{R}^d$
- Assume X is generated by a process which undergoes a transition from state A to state B,
  - with probability distributions P_A and P_B respectively and $P_A \neq P_B$.
- A time τ is a change point if:

$$\{x_1, x_2, \dots x_\tau\} \sim P_A$$
$$\{x_{\tau+1}, x_{\tau+2}, \dots x_T\} \sim P_B$$

Related work

- Hypothesis testing: (Kuncheva, 2013)
  - H_0 - x_t and x_{t-1} drawn from the same multivariate Gaussian distribution
- CUSUM (Jeske et al., 2009)
  - monitor cumulative sum which measures accrued deviations
- Bayesian change-point detection (Adams and MacKay, 2007)
  - Estimate posterior probability of the "run-time" distribution
  - "run-time": length of time since last change point
- KLIEP (Sugiyama et al., 2007; Kawahara et al. 2012)
  - approximate density ratio to measure change in distribution
- **Virtual classifiers** (Desobry et al., 2005; Hido et al, 2008, Yamada et al., 2013)
  - **measure likelihood of change point using classification accuracy**

Unsupervised

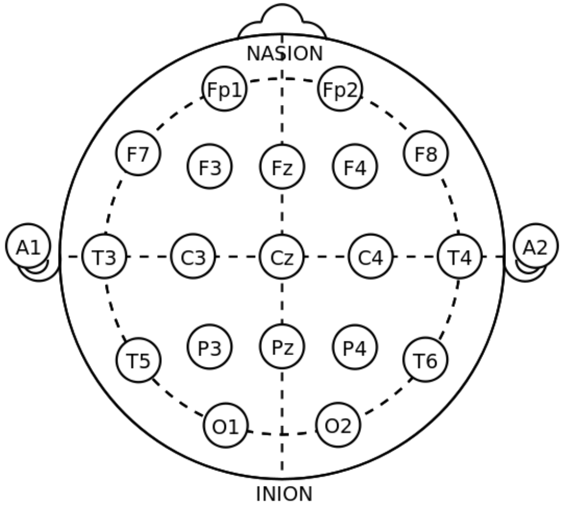
Semisupervised

Deep virtual classifiers for seizure prediction

Khan, H., Marcuse, L., Fields, M., Swann, K., & Yener, B., (2017). Focal onset seizure prediction using convolutional networks. *IEEE Transactions on Biomedical Engineering*.

What is Epilepsy?

- Epilepsy is a neurological disorder characterized by the unpredictable occurrence of seizures.
  - Affects 65M people in the world, 3.4M in the US
- Symptoms of seizures:
  - Convulsions
  - Auras
  - Forgetfulness



Seizure prediction horizon

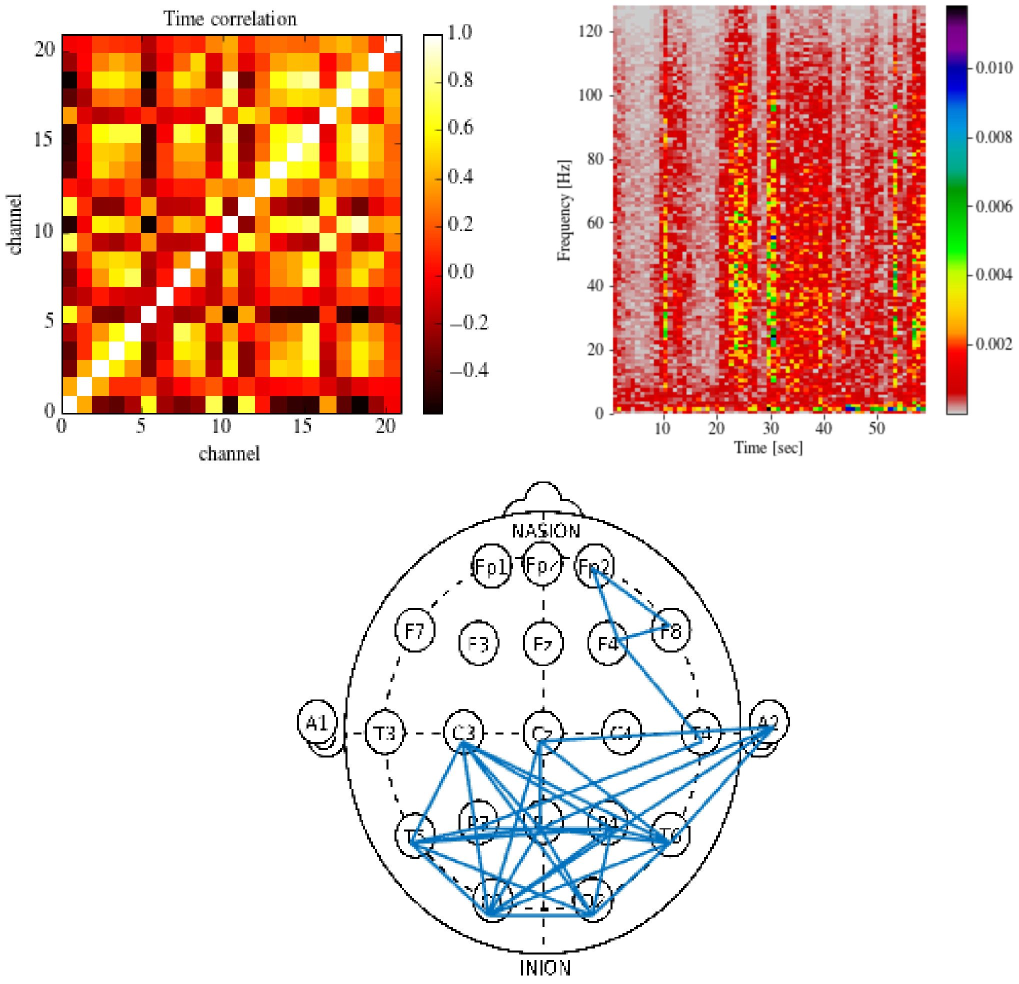
- Changes occur in the brain prior to seizure onset that make the seizure inevitable.
  - **Seizure prediction horizon (PH), preictal state/period**
- Central question: When do the pre-seizure changes occur?

Assumed pre-ictal period (min)	Sensitivity (%)	False-positive rate (FP/h)	Mean prediction time (min)	Statistical validation of performance
30	94	0	12	No
20	89	n.a.	3	No
20	83	n.a.	6	No
20	94	n.a.	4	No
n.s.	100	0	n.s.	No
60	100	0	n.s.	No
262.5	100	n.a.	52	No
60	96	n.a.	7	No
Variable	91	n.s.	49	No
180	90	0.12	19	No
n.s.	77	n.s.	Several min	No
n.s.	47	0	19	No
60	95	0	n.s.	No
3	100	n.a.	2	No
90	83	0.31[c]	8	No
Variable	100	n.s.	83	No
240	86	0	86/102[h]	Yes
240	81	0	4–221	No
60	0	n.a.	–	No
2	94	0.08[f]	5–80 s	No
90	n.s.	n.s.	>>30	No
60	88	0.02	35	No

Table from (Mormann et al. 2007)

Features for seizure prediction

- Seizure prediction horizon (PH)
  - Previous studies assume PH in the range 2 minutes to 262.5 minutes (Mormann et al., 2016)
  - PH reported varies based on features extracted
- Features:
  - Time/frequency domain features (Karoly et al., 2016)
  - Multivariate features (Cho et al., 2017; Dhulekar et al., 2016)
  - Model based features (Arabi and He, 2014)



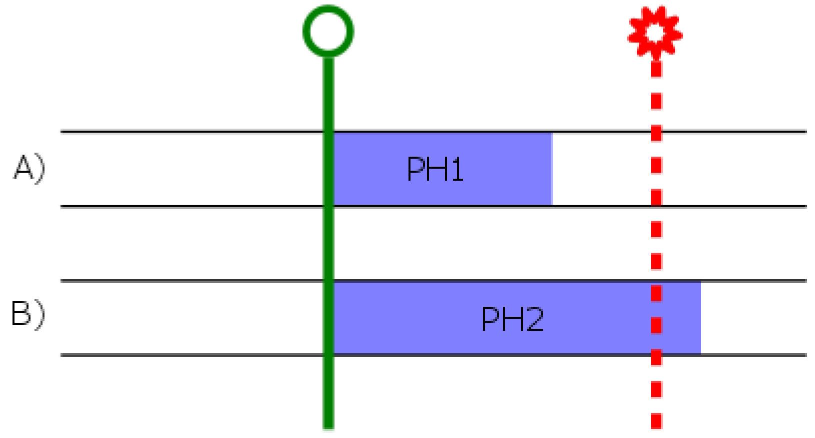
Examples of features extracted for seizure prediction.

State of the art

- “Crowdsourcing reproducible seizure forecasting in human and canine epilepsy” (Brinkman et al., 2016)
  - Results of Kaggle competition on seizure prediction
  - Winning submissions used time/frequency domain features extracted from intracranial human and dog EEG
  - PH – 60 mins
- “Prediction of seizure likelihood with a long-term, implanted seizure advisory system in patients with drug-resistant epilepsy: A first-in-man study” (Cook et al., 2013)
  - Implanted seizure prediction device
  - Three energy measures in filtered intracranial EEG as features
  - PH – 6 – 30 minutes (optimized per patient)
- “On the proper selection of preictal period for seizure prediction” (Bandarabadi et al., 2015)
  - Measure common area (C) between preictal and interictal feature histograms
  - Define optimal prediction horizon for a single feature as minimum C

Evaluating prediction systems

- Sensitivity: percentage of events predicted within prediction horizon
- Specificity: false prediction rate
- Comparison to random predictor (Schelter et al., 2006)



The prediction horizon is a critical parameter for a prediction system as it can be increased arbitrarily to achieve perfect sensitivity.

Is a system better than random?

- Analytical model given by (Schelter et al., 2006)
- Predictions are generated with probability $P \approx \mathrm{FPr} * \mathrm{PH}$
- To perform better than random, sensitivity must be greater than:

$$\sigma > \frac{\max_{k}\left\{\left(1 - \left(\sum_{j<k}\binom{K}{j} P^{j}(1-P)^{K-j}\right)^{d}\right) > \alpha\right\}}{K}$$

- Where K is the number of analyzed events, d is the dimension of the feature space, and α is a significance level.

Virtual classifiers (VC) - Theory

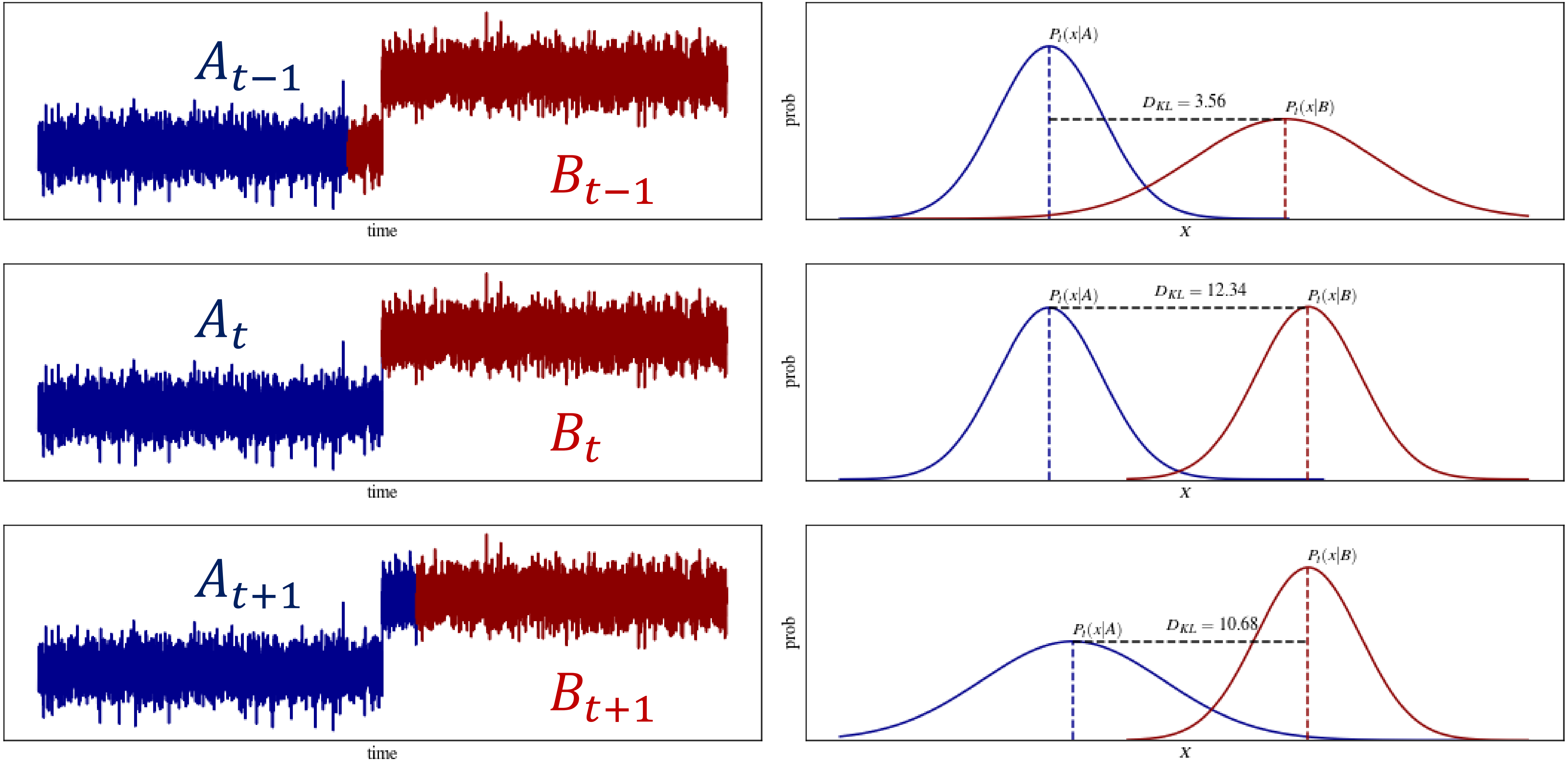
Time series of feature vectors $\{x_k\}_{k=1}^{T}$ with state space $\mathcal{X} = \mathbb{R}^d$.
Time t defines a split of the time series into disjoint sets:

$$A_t = \{x_1, x_2, \dots, x_t\}$$
$$B_t = \{x_{t+1}, x_{t+2}, \dots, x_T\}$$

- Consider change point detection as an optimization problem:

$$\max_t D(P(x|A_t), P(x|B_t))$$

- Idea is to approximate $D(P(x|A_t), P(x|B_t))$ with classification accuracy



Example of a Gaussian noise signal undergoing a mean shift. By splitting the signal into segments and approximating the conditional probability distributions with Gaussians, we see the KL-divergence is maximal when the split matches the change point.

Approximating KL-divergence with VC

$$\max_t D(P(x|A_t), P(x|B_t))$$

- Using the KL-divergence for $D(\cdot,\cdot)$ yields:

$$\max_t \sum_{x \in \mathcal{X}} P(x|A_t) \log\left(\frac{P(x|A_t)}{P(x|B_t)}\right)$$

$$\max_t \sum_{x \in \mathcal{X}} P(x|A_t) \log P(x|A_t) - \sum_{x \in \mathcal{X}} P(x|A_t) \log P(x|B_t)$$

Bayes rule to isolate posterior distribution

- Applying Bayes rule to $P(x|B_t)$ yields:

$$\max_t \sum_{x\in\mathcal{X}} P(x|A_t)\log P(x|A_t) - \sum_{x\in\mathcal{X}} P(x|A_t)\log P(x) - \sum_{x\in\mathcal{X}} P(x|A_t)\log P(B_t|x) + \sum_{x\in\mathcal{X}} P(x|A_t)\log P(B_t)$$

$$\max_t D\big(P(x|A_t), P(x)\big) - \sum_{x\in\mathcal{X}} P(x|A_t)\log P(B_t|x) + \log P(B_t)$$

$$\max_t - \sum_{x\in\mathcal{X}} P(x|A_t)\log P(B_t|x)$$

- $D\big(P(x|A_t), P(x)\big) \geq 0$
- $P(B_t|x)$ as a classifier $f(x)$
- $P(x|A_t)$ as set of labels y

$$-\sum_{x\in\mathcal{X}} P(x|A_t)\log P(B_t|x) \approx \frac{1}{T}\sum_{i=1}^{T} y_i \log f(x_i)$$

Deep Virtual Classifiers

- Use deep feedforward network instead of a simpler classifier
  - Optimization over feature transform with parameters θ

$$\max_t - \sum_{x \in \mathcal{X}} P(x|A_t) \log P(B_t|x) \approx \min_{t,\theta} - \frac{1}{T} \sum_{i=1}^{T} y_i \log f(x_i; \theta)$$

Virtual classifiers summary

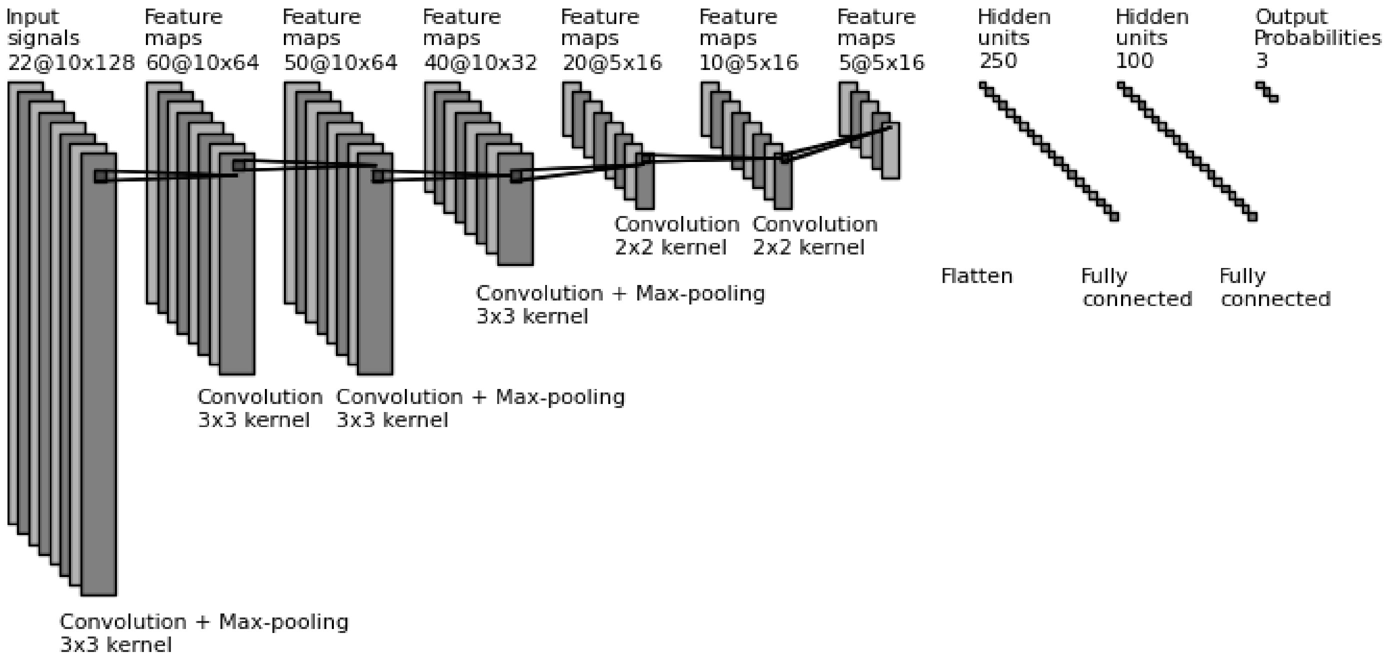
- Given:
  - a set of candidate change points $\{\tau_1, \tau_2, \dots \tau_m\}$
  - a set of time series $\{X_i\}_{i=1}^{n}$
- Construct a set of binary labels $\{Y_j\}_{j=1}^{m}$
- Each Y_j is a vector of length T with:

$$Y_{jk} = \begin{cases} -1 \; if \; k \leq \tau_j \\ 1 \; if \; k > \tau_j \end{cases} \quad for \; k = 1,2, \dots T$$

- Copies of each of these label vectors Y_j are paired with every time series in $\{X_i\}_{i=1}^{n}$ forming the pseudo-labeled dataset $D_j = \{(X_i, Y_j)\}_{i=1}^{n}$.
- A classifier is trained on each dataset D_j, resulting in m classifiers each trained on a different labeling of the data.
- Accuracy on a validation set of each of the classifiers is measured as $p_1, p_2, \dots p_m$.

CNN for feature extraction

- Use CNN to learn features from EEG
- Convolutions over time and frequency domain via wavelets

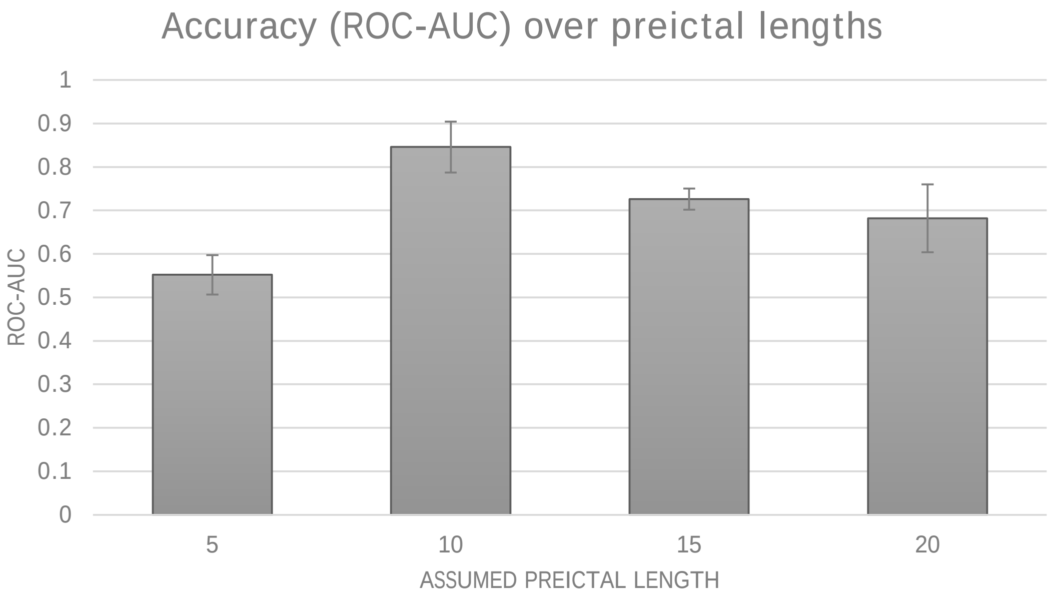


Convolutional neural network trained on EEG to predict brain states from wavelet transformed EEG.

VC for preictal period length

- Candidate preictal lengths were 5, 10, 15, and 20 mins
- A CNN was trained for each labelling of the data
- Preictal length of 10 mins was chosen based on significant improvement in accuracy

ROC-AUC between interictal and preictal classes for different assumed preictal lengths. Averaged over 10-folds of validation data, error bar shows 1 standard deviation.

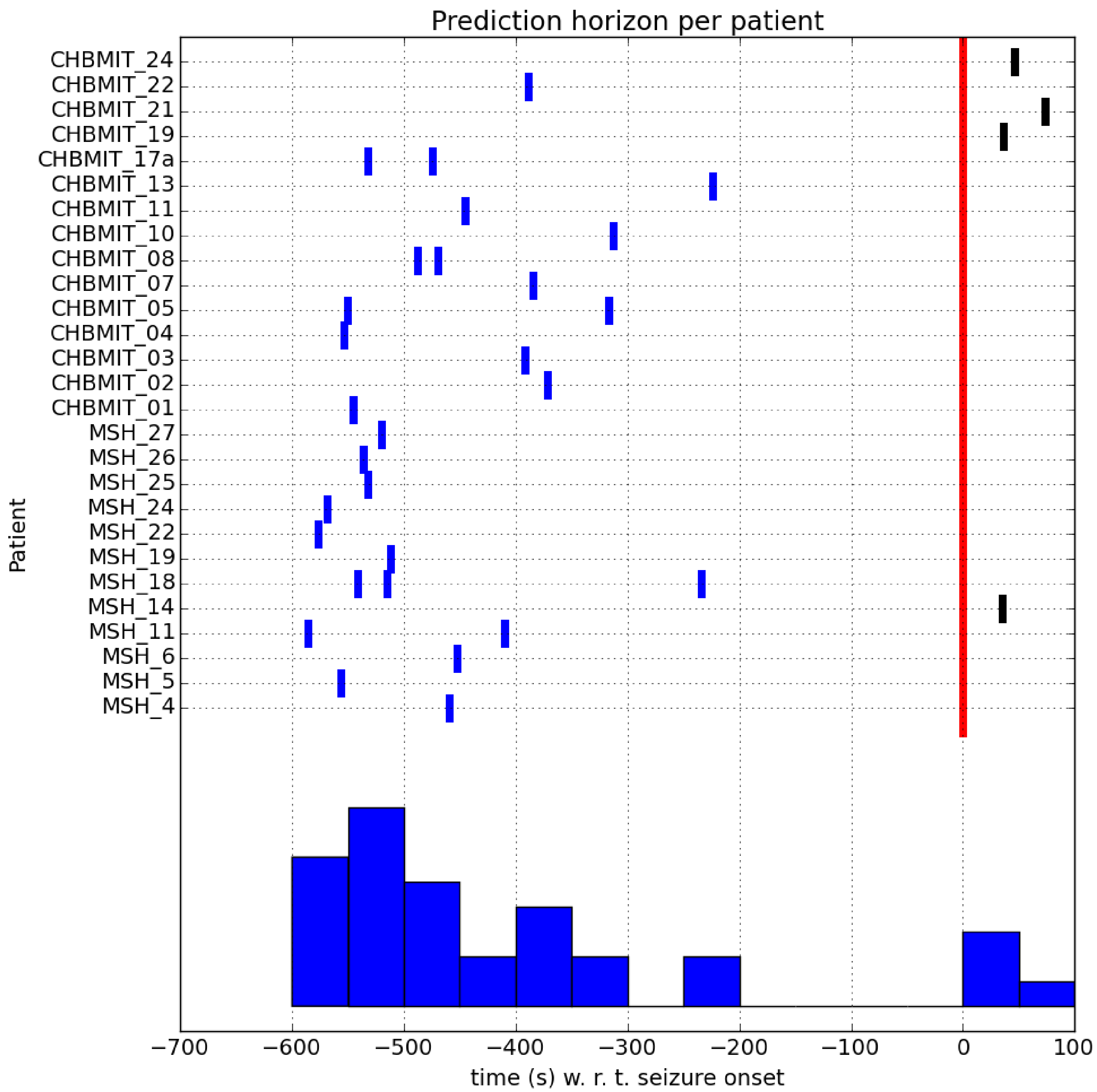


Results and Comparison

- Dataset of 500+ hours of 22-channel EEG with 200+ seizures
- We compared our results to:
  - 2 top performing algorithms from Kaggle (Brinkmann et al., 2016)
  - Algorithm from (Cook et al., 2013)

Seizure prediction results

Method	PH (mins)	Sensitivity	FPr (FP/h)	Random pred. $\sigma_{low} - \sigma_{high}$
Kaggle1	60	72.7%	0.285	15.1% - 27.2%
Kaggle 2	60	75.8%	0.230	12.1% - 24.2%
Cook et al.	PS*	66.7%	0.186	12.1% - 21.2%
This work	10	**87.8%**	**0.142**	9.1% - 15.1%



Prediction times generated by the CNN for all test set recordings with seizures grouped by patient. The spread of the prediction times is large indicating a non-uniform transition time within patients and between patients.

Limitations

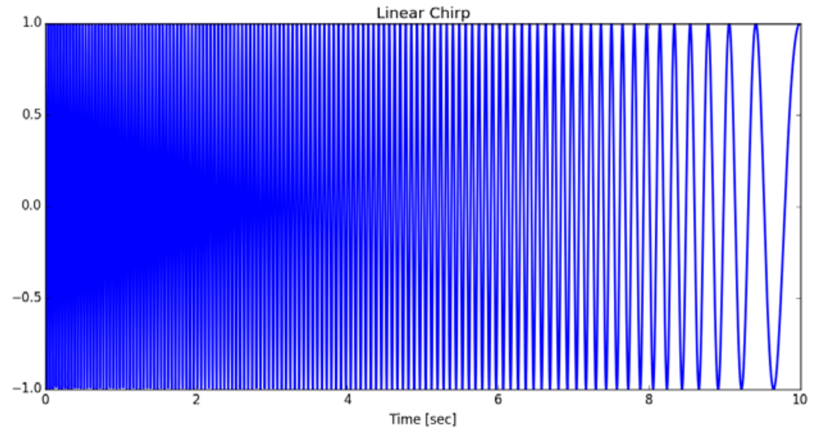
- VC requires a uniform transition time over all time series
  - Otherwise combinatorial explosion of m^n occurs
- VC is also very computationally expensive
  - Requires training m neural nets multiple times.

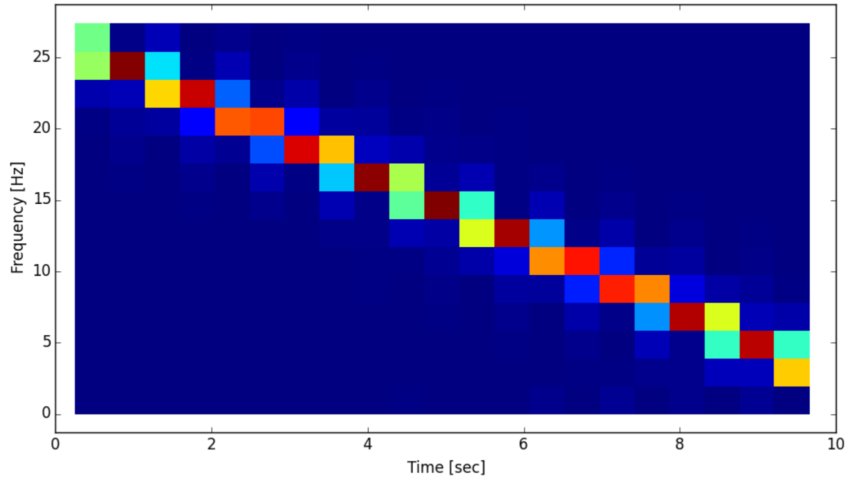
Learning spectral decompositions with wavelets

Khan, H. & Yener B., (2018). Learning filter widths of spectral decompositions with wavelets. *Advances in Neural Information Processing (NeurIPS)*

Spectral decompositions

- Models for time series use a spectral decomposition of signals as input
- Typically use cross validation to pick parameters
- Applications such as automatic speech recognition (Hinton et al., 2012), biological signal analysis (Andreao et al., 2006), and financial time series (Cao et al., 2003)





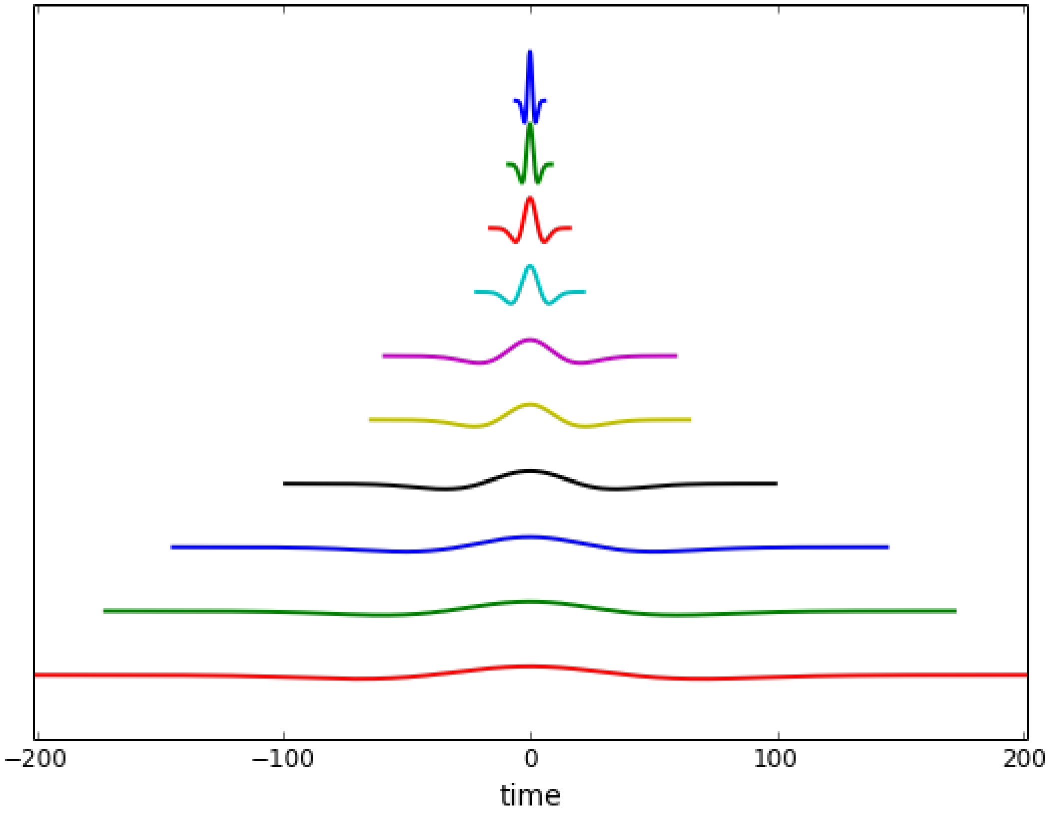
Top: Linear chirp signal. Bottom: Spectrogram of the linear chirp signal.

Wavelet transform

- Mother wavelet function

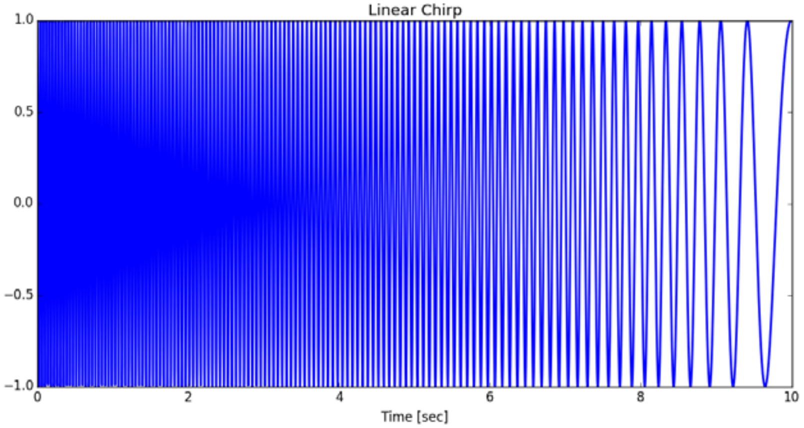
$$\Psi_w(t) = \frac{2}{\sqrt{3w\pi^{\frac{1}{4}}}}\left(1 - \frac{t^2}{w^2}\right)e^{-\frac{t^2}{2w^2}}$$

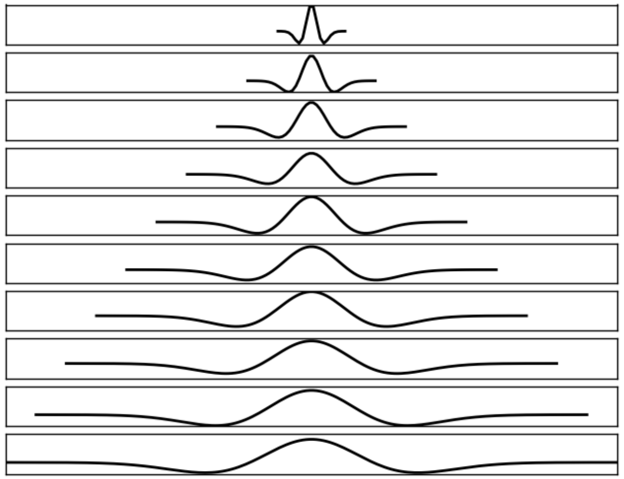
- Scale the mother wavelet and convolve with the signal

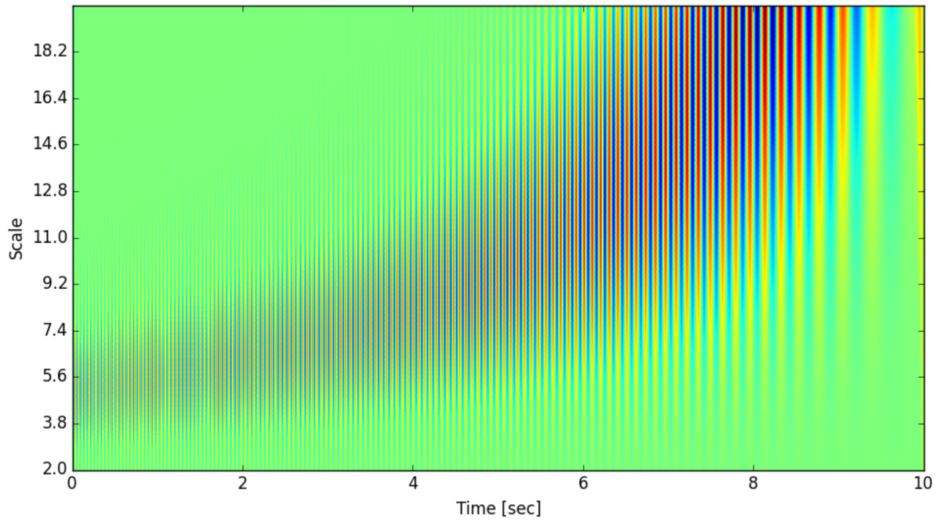


Wavelet filters.

Spectral decomposition with wavelets

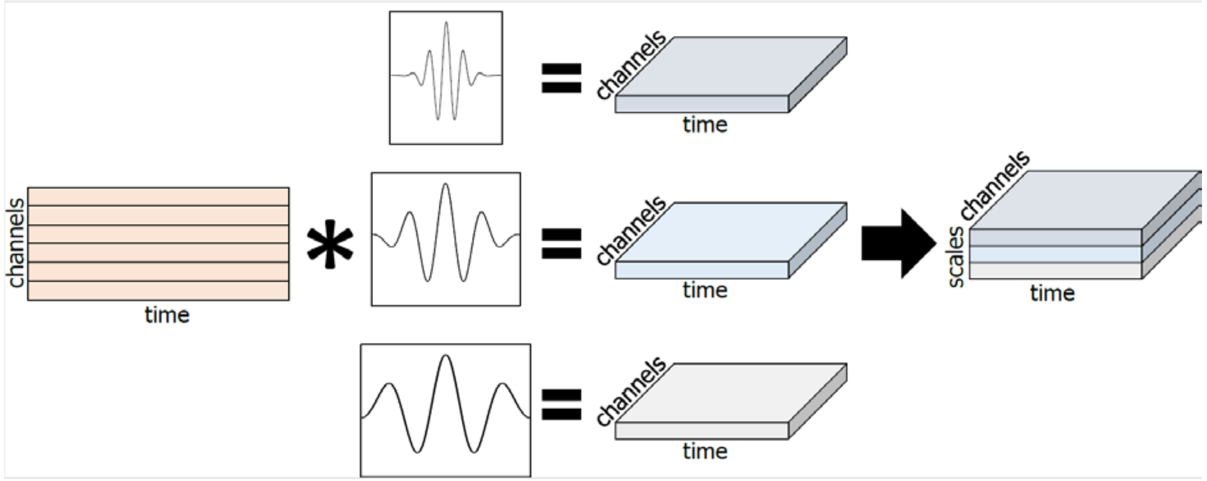




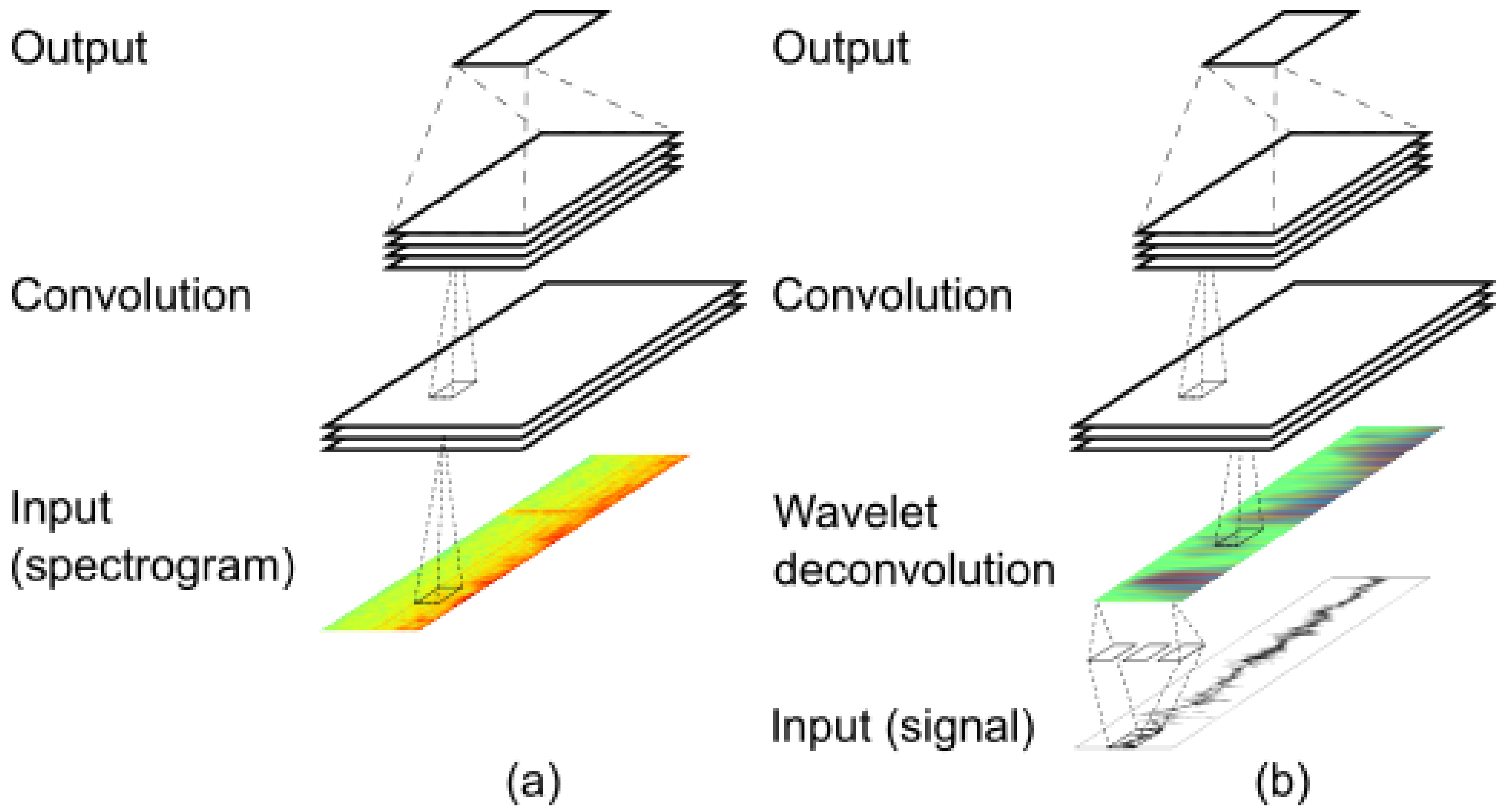


Automatically extracting time/frequency domain features

- Combine the wavelet transform and CNN
- Use backpropagation to learn the scale parameters
- learn the “width” of the filters with gradient descent



The wavelet transform applied to a multichannel signal


Output
Convolution
Input
(spectrogram)
(a)
Output
Convolution
Wavelet
deconvolution
Input (signal)
(b)

Learn scales with backpropagation

The output of the wavelet layer is given by:

$$y_{ij} = \sum_{a=1}^{P} s_{ia}\, x_{j+a} \quad \forall i = 1 \dots M$$

Where the wavelet filter $s_i \in \mathbb{R}^{1xP}$ is the discretized wavelet function over the grid $k = \left\{-\frac{P-1}{2} \dots \frac{P-1}{2}\right\}$:

$$s_{ia} = \frac{2}{\sqrt{3w_i \pi^{\frac{1}{4}}}}\left(1 - \frac{k_a^2}{w_i^2}\right) e^{-\frac{k_a^2}{2w_i^2}} \quad \forall a = 1 \dots P$$

For backpropagation, we want $\frac{\delta E}{\delta w_i}$ where E is some error function:

$$\frac{\delta E}{\delta w_i} = \sum_{a=1}^{P} \frac{\delta E}{\delta s_{ia}} \frac{\delta s_{ia}}{\delta w_i} = \sum_{a=1}^{P} \frac{\delta E}{\delta s_{ia}} \left[A\left(M\frac{\delta G}{\delta w_i} + G\frac{\delta M}{\delta w_i}\right) + MG\frac{\delta A}{\delta w_i}\right]$$

$$\frac{\delta E}{\delta s_{ia}} = \sum_{j=1}^{N} \frac{\delta E}{\delta y_{ij}} \frac{\delta y_{ij}}{\delta s_{ia}} = \sum_{j=1}^{N} \frac{\delta E}{\delta y_{ij}} x_{j+a}$$

$$A = \frac{2}{\pi^{\frac{1}{4}}}(3w_i)^{-\frac{1}{3}}$$

$$M = 1 - \frac{k_a^2}{w_i^2}$$

$$G = e^{-\frac{k_a^2}{2w_i^2}}$$

$$\frac{\delta A}{\delta w_i} = -\frac{6}{\pi^{\frac{1}{4}}}\boxed{(3w_i)^{-\frac{3}{2}}}$$

$$\frac{\delta M}{\delta w_i} = \frac{2k_a^2}{w_i^3}$$

$$\frac{\delta G}{\delta w_i} = \frac{k_a^2}{w_i^3} e^{-\frac{k_a^2}{w_i^2}}$$

$w_i > 0$

Results

- TIMIT Phone recognition dataset
- UCR Haptics dataset

Best reported PER on the Timit dataset without context dependence

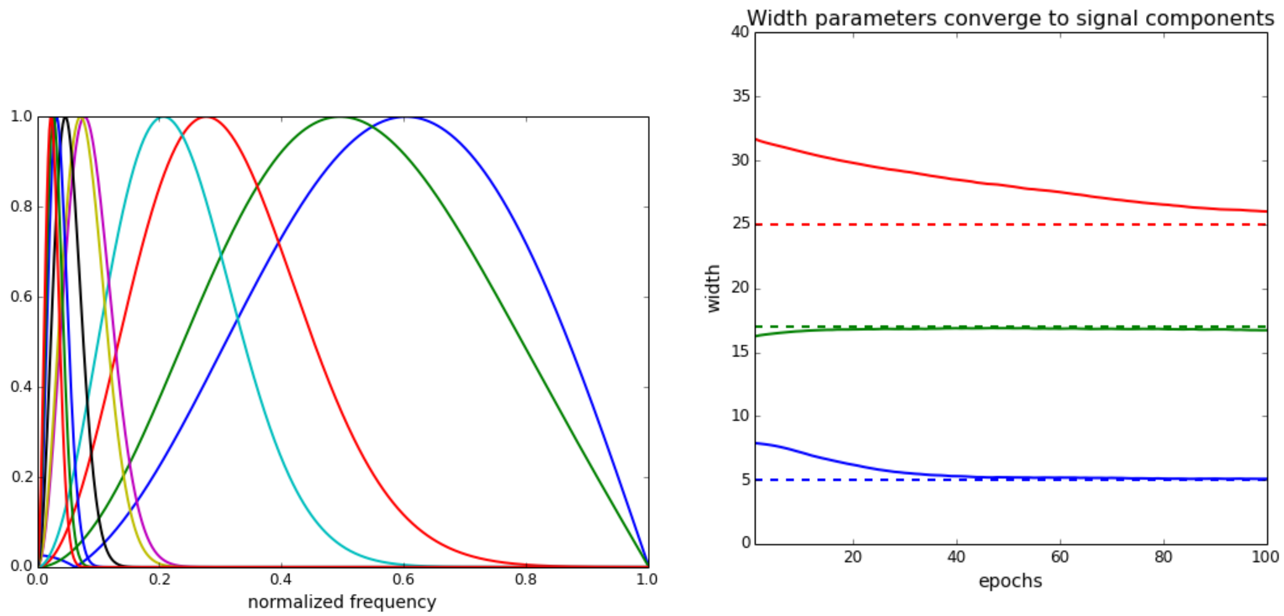
Method	PER (Phone Error Rate)
DNN with ReLU units	20.8
CNN	18.9
DNN + RNN	18.8
WD + CNN (this work)	18.1
LSTM RNN	17.7
Hierarchical CNN	**16.5**

Test error on the Haptics dataset

Method	Test Error
DTW	0.623
BOSS	0.536
ResNet	0.495
COTE	0.488
FCN	0.449
WD + CNN (this work)	**0.425**

Learned filters

- The learned filters resemble engineered filter banks for ASR
- Learning can be slow with vanilla SGD
  - Use ADAM (Kingma and Ba, 2014)



Left: Learned wavelet filter bank for TIMIT phone recognition task. Right: Parameters of the wavelet transform converge to frequencies used to generate artificial data

Acknowledgements

- Prof. Bülent Yener
- Dr. Lara Marcuse and Dr. Madeline Fields