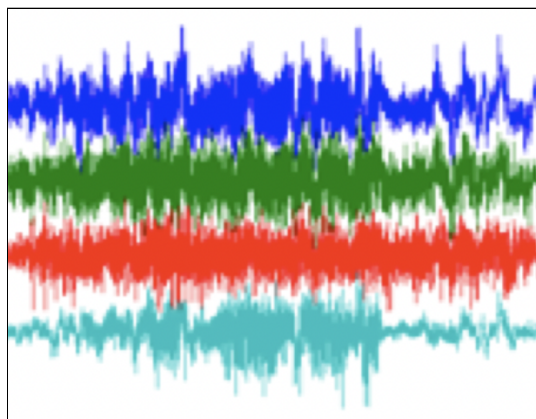


Aprendizagem Computacional

## Relatório do Trabalho Prático 2b



## Prediction and Detection of Epileptic Seizures

2021/2022

Mestrado em Engenharia Informática

PL3  
PL3

Pedro Rodrigues  
Miguel Rabuge

2018283166  
2018293728

[pedror@student.dei.uc.pt](mailto:pedror@student.dei.uc.pt)  
[rabuge@student.dei.uc.pt](mailto:rabuge@student.dei.uc.pt)

# Índice

<b>Prediction and Detection of Epileptic Seizures</b>	<b>3</b>
Datasets	3
Carregamento dos Dados	4
Divisão do Dataset	5
Pré-Processamento	7
Pós-Processamento	7
Arquitetura das Redes Neurais	8
FeedForwardNet	8
LayRecNet	8
Autoencoder e Stacked Autoencoders	9
CNN	9
LSTM	10
Resultados	10
Metricas consideradas	10
Predição de Convulsões	10
Deteção de Convulsões	12
Conclusões	14

# 1. Prediction and Detection of Epileptic Seizures

Neste segundo capítulo, o principal objetivo deste problema foi o reconhecimento, detecção e previsão de convulsões (“seizures”) recorrendo para isto a modelos baseados em redes neurais de diversos tipos, com o objetivo de tentar perceber quais os modelos apresentam melhor performance. Nas secções seguintes fazemos uma explicação detalhada acerca do dataset utilizado bem como todo o processamento feito neste por forma a prepará-lo para o seu uso nos modelos. Discutimos também as arquiteturas dos modelos treinados, setup experimental, resultados obtidos e terminamos extraindo algumas conclusões.

## 1.1. Datasets

Para a análise realizada neste trabalho foram-nos fornecidos datasets que contém resultados de medições de sinais elétricos cerebrais durante um dado período temporal (EEG - Eletroencefalograma) efetuadas a pacientes que sofrem de epilepsia. Estes pacientes apresentam frequentemente, fruto da doença, fenômenos de convulsão “seizures” com uma duração variável na ordem de segundos ou até mesmo minutos. No decorrer do tempo em que são efetuadas as medições podemos distinguir 4 classes / estados cerebrais distintos: “pré-ictal”, “ictal”, “pós-ictal” e “inter-ictal”.

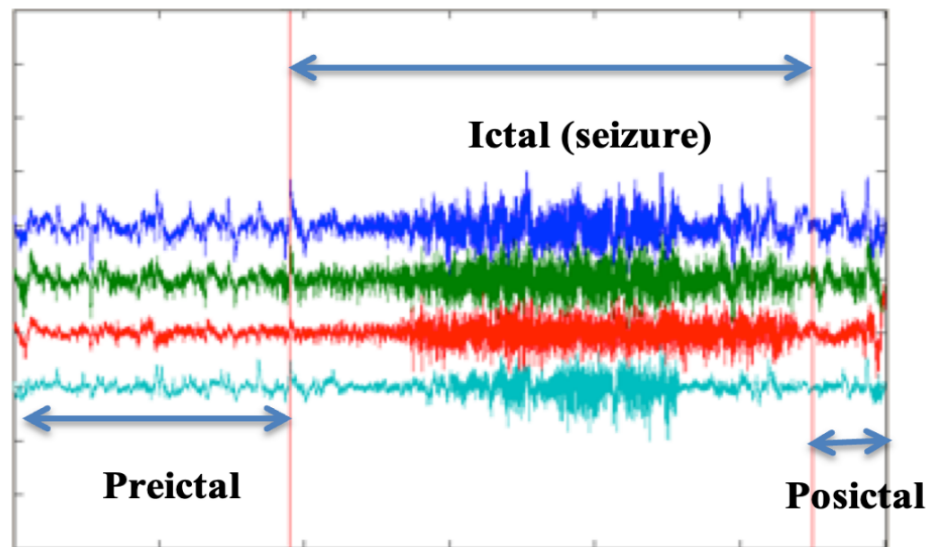


Figura 1 - EEG de um paciente epileptico a sofrer de uma convulsão.

Como podemos observar na imagem o período temporal que precede uma convulsão é denominado de “pré-ictal” e tem uma duração de sensivelmente 15 minutos. Desta forma, o período temporal em que o fenómeno de convulsão toma lugar é denominado período “ictal”. O período imediatamente a seguir à convulsão é denominado de “pós-ictal” podendo ainda ser chamado de

“inter-ictal” caso se trate de um período “pós-ictal” que ocorra antes do período “pré-ictal” da convulsão seguinte. Para efeitos de classificação e referência no resto deste documento considerámos algumas classes que correspondem a cada um destes períodos e para as quais é esperado que os modelos de classificação (redes neurais) desenvolvidos devolvam um dado valor categórico. As classes correspondentes a cada período e o valor esperado são mostrados na tabela abaixo.

Classes	Periodos	NN Output
1	“pós-ictal” + “inter-ictal”	[ 1 0 0 ]
2	“pré-ictal”	[ 0 1 0 ]
3	“ictal”	[ 0 0 1 ]

Tabela 1 - Classes e resultados esperados.

Para efeitos do nosso estudo foram nos atribuídos os pacientes com os identificadores “54802” e “112502”. Em cada um dos ficheiros de dados correspondentes a cada um dos pacientes é possível encontrar duas matrizes: “FeatVectSel” e “Trg”.

No caso da primeira matriz, esta contém os dados relativos a todas as medições realizadas durante a sessão de EEG, sendo que cada coluna corresponde a uma “feature” (tendo a matriz contendo estes dados de ser transposta para posterior tratamento) . As 29 “features” obtidas em cada instante de são na verdade resultantes de amostragem dos dados “raw” obtidos através do EEG em segmentos 2 segundos e 50% de sobreposição entre segmentos, ou seja cada instante temporal ao qual correspondem 29 features partilha 50% da informação obtida no instante anterior.

No caso da segunda matriz, esta apenas contém para cada instante temporal a que correspondem 29 “features” na matriz anterior, a classe / estado cerebral em que o paciente se encontra de momento, sendo o valor “0” nesta matriz representativo da situação em que o paciente se encontra no estado “pré-ictal”, “pós-ictal” e “interictal” e o valor “1” caso este se encontre num período onde está a sofrer de uma convulsão “ictal”.

### 1.1.1. Carregamento dos Dados

Em termos de carregamento de dados, como é possível ver na figura abaixo, é possível selecionar na GUI disponibilizada o dataset associado ao paciente que

se pretende analisar, sendo posteriormente o ficheiro de dados que lhe está associado importado.

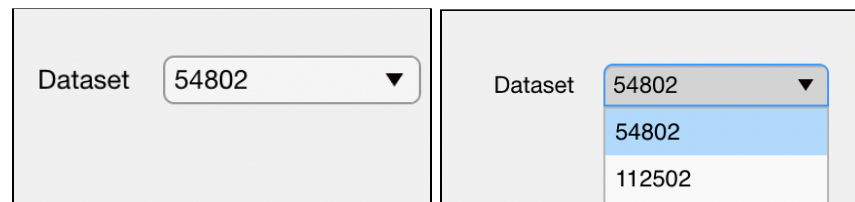


Figura 2(a) e 2(b) - Seleção do dataset na GUI

Após importar dos ficheiros as matrizes relativos a cada um dos pacientes foi necessário então aplicar uma transformação nestes para que ficassem com o aspeto necessário para posteriormente serem divididos e processados.

No caso da matriz “FeatVectSel” contendo as 29 “features” para cada instante temporal, foi necessário aplicar uma transposição por forma a colocar as “features” um por linha, sendo cada coluna representativa de uma “unidade de tempo”.

No caso da matriz “Trg” contendo a identificação do periodo em que está a ocorrer uma convulsão foi necessário transformar esta por forma a obter outra matriz que classifica cada instante de tempo como pertencente a cada uma das classes representadas na tabela acima, seguindo a representação que as redes neuronais devem devolver. Simplifica-se assim a comparação dos resultados devolvidos durante a execução de uma rede com os esperados (contidos nesta nova matriz).

A classificação das “features” realizada assenta no facto de sabermos claramente os pontos correspondentes aos instantes em que o paciente se encontra no período “ictal” - classe 3; (representado com “1” na matriz Trg) e aos instantes “pré-ictal” - classe 2; (15min antes da convulsão o que se traduz em 900 pontos anteriores à convulsão na matriz “FeatVectSel” ), os restantes pontos são considerados como “pós-ictal” + “inter-ictal” - classe 1.

### 1.1.2. Divisão do Dataset

Para efetuar a divisão do dataset em dados de treino das redes neuronais e dados de teste das mesmas, começamos por fazer uma análise dos dados fornecidos por forma a perceber como estavam distribuídas as convulsões epiléticas, tornando-se assim mais fácil fazer uma divisão que contenha exemplos de todas as classes em ambas as partes. A análise exploratória dos

dados dos pacientes pode ser visualizada nos gráficos apresentados abaixo (fig. 3 e 4).

Chegamos à conclusão que a divisão dos dados mais lógica seria considerando o rácio 70% treino / 30% teste para o paciente “54802”, e 35% treino / 65% teste para o paciente “112502”. Não foi utilizado o conjunto de validação no decorrer das nossas experiências

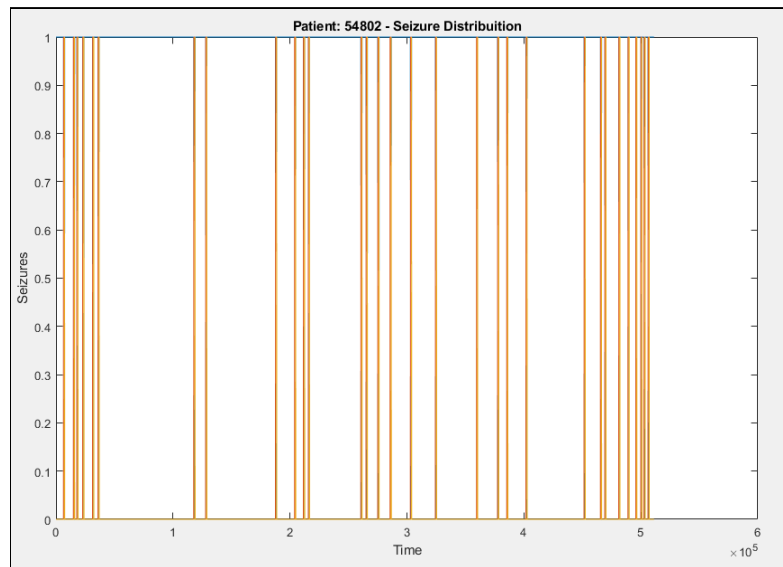


Figura 3 - Distribuição das convulsões do paciente 54802

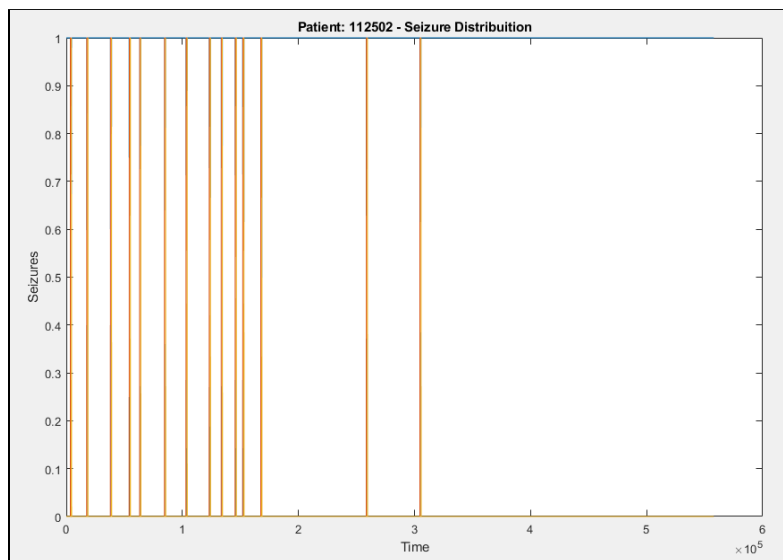


Figura 4 - Distribuição das convulsões do paciente 112502

Em termos de normalização dos dados dentro de cada um destes “splits”, para que exista um igual número de amostras de cada classe mantendo na mesma a ordenação temporal dos pontos a estratégia utilizada foi:

- Manutenção e contagem do número de pontos, “N”, pertencentes às classes 2 (“pré-ictal”) e 3(“ictal”), que são bastante relevantes e que constituem a minoria no dataset fornecido.
- “Sub-sampling” aleatório de “N” pontos pertencente à classe 1 (“pós-ictal + “inter-ictal”) que constitui a maioria no dataset fornecido.

### 1.1.3. Pré-Processamento

Em termos de pré-processamento, sempre que a rede seja uma CNN ou uma LSTM é utilizada a função “*transform*” que recebe os dados (X) e labels (y) de treino/teste, já divididos, formando N blocos de N\_features x 29, para o X, e um array categórico Nx1 para o y, cujos elementos pertencem a {1, 2, 3}.

Estes blocos são contínuos, isto é, são formados por amostras temporais com a mesma classe. Na construção dos mesmos, caso se detete um elemento diferente do inicial, num dado bloco, os pontos constituintes desse bloco até esse momento são descartados e o bloco é reconstruído, começando nesse índice cujo valor é diferente.

Dependendo da rede, caso seja uma CNN, os blocos são *reshaped* para um array 4D de dimensões 29x29x1xN, dado que os N\_features são fixados a 29, uma vez que o enunciado não pede redução da dimensionalidade das features através de autoencoders, para a mesma. No caso de uma LSTM, X passa a ser um *cell array* 1xN, onde cada célula é também um *cell array* com tamanho N\_features x 29, admitindo redução da dimensionalidade das features com autoencoders. Os labels resultantes deste processo são iguais para ambas as redes.

### 1.1.4. Pós-Processamento

Em termos de pós-processamento, é dada a opção na GUI de escolher a *tick box* “Post Processing”. Este pós processamento leva em consideração a hipótese relaxada mencionada no enunciado, e é implementado pela função *nested “postproc”*, da função “*evaluate*”. A implementação baseia-se em varrer o output da rede neuronal com uma janela deslizante de tamanho 10, sem sobreposição, com um threshold de 5. Numa dada janela, caso exista uma classe com uma frequência maior do que o threshold, todos os 10 elementos da

janela são transformados para essa classe. Os pontos finais, quando não é possível aplicar mais uma janela de tamanho 10, não sofrem alteração.



Figura X - Tick Box de Pós-Processamento na GUI

## 1.2. Arquitetura das Redes Neurais

Relativamente a as redes neurais feedforward fizemos várias experiências para tentar averiguar quais os modelos e estruturas de redes que conseguem garantir uma boa performance quer para detecção quer para previsão. No entanto devido à elevada combinação do problema foi complexo garantir que os parâmetros que utilizamos são os melhores possíveis. Também verificamos que torna-se bastante complexo construir um modelo de redes neurais que seja apropriado para ambas as tarefas. Encontram-se então descritas nas secções seguintes as arquiteturas que utilizamos para o nosso estudo experimental. Todas as redes seguintes foram implementadas recorrendo à toolbox de machine learning e neural networks do matlab.

### 1.2.1. FeedForwardNet

Relativamente a redes neurais feedforward entre várias arquiteturas possíveis acabamos por ficar como um modelo que apresenta os seguintes parâmetros.

- Função de Treino: “traingd” (Gradient Descent)
- Camada de Input: 29 neurónios
- Camada Escondida: 10 neurónios
- Camada de Output: 3 neurónios
- Epochs: 1000
- Funções de ativação (por ordem): Sigmoidal, Linear

### 1.2.2. LayRecNet

Relativamente a redes neurais com delay entre várias arquiteturas possíveis acabamos por ficar como um modelo que apresenta os seguintes parâmetros.

- Função de Treino: “traingd” (Gradient Descent)
- Camada de Input: 29 neurónios
- Camada Escondida: 10 neurónios
- Camada de Output: 3 neurónios
- Delay: 2
- Funções de ativação (por ordem): Sigmoidal, Linear
- Epochs: 1000

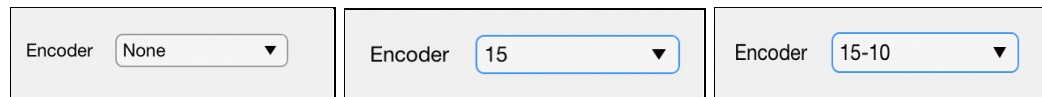


### 1.2.3. Autoencoder e Stacked Autoencoders

Relativamente a autoencoders, foram implementadas duas versões:

- Simples: Um autoencoder (função *autoencoder*)
- Composto: Um pipeline de autoencoders (função *stacked\_encoder*)

A sua implementação tem como base a função “*trainAutoencode*” e “*encode*” do matlab, que treinam um encoder e codificam os dados, respetivamente.



The figure shows three separate GUI panels, each with a label 'Encoder' and a dropdown menu. The first panel has 'None' selected. The second panel has '15' selected. The third panel has '15-10' selected. Each panel is enclosed in a light gray border.

Figura X(a), X(b) e X(c) - Escolha de autoencoders na GUI (sem, simples e composto)

### 1.2.4. CNN

Relativamente a redes neurais profundas, neste caso convulsionária entre várias arquiteturas possíveis, acabamos por ficar como um modelo que apresenta os seguintes parâmetros.

- Camada de Input: 29x29 neurónios (ImageInputLayer([29 29 1]))
- Primeira camada:
  - convolution2dLayer(3,8,'Padding','same')
  - batchNormalizationLayer
  - reluLayer
- Segunda camada: maxPooling2dLayer(2,'Stride',2)
- Terceira camada:
  - convolution2dLayer(3, 16,'Padding','same')
  - batchNormalizationLayer
  - reluLayer
- Quarta camada: maxPooling2dLayer(2,'Stride', 2)
- Quinta camada:
  - convolution2dLayer(3,32,'Padding','same')
  - batchNormalizationLayer
  - reluLayer
- Camada de Output (3 neurónios): f
  - fullyConnectedLayer(3)
  - softmaxLayer
  - classificationLayer
- Ritmo de aprendizagem inicial: 0.01
- Max epochs: 100
- Shuffle: 'every-epoch', ...
- Função de treino: “sgdm” (Stochastic gradient descent with momentum)

### 1.2.5. LSTM

Relativamente a redes lstm neste caso convulsionária entre várias arquiteturas possíveis, acabamos por ficar como um modelo que apresenta os seguintes parâmetros.

- Camada de Input: 29 neurónios (sequenceInputLayer(29))
- Primeira camada: lstmLayer(100, 'OutputMode', 'last')
- Camada de Output (3 neurónios):
  - fullyConnectedLayer(3)
  - softmaxLayer
  - classificationLayer
- Ritmo de aprendizagem inicial: 0.01
- Limiar do gradiente: 1
- Tamanho da Batch: 27
- Tamanho da sequência: 'longest'
- Max epochs: 20
- Shuffle: 'never',
- Função de treino: "adam"

## 1.3. Resultados

Neste subcapítulo detalhamos os resultados experimentais, nomeadamente quais métricas utilizamos, e quais foram os resultados, quer de predição ou de deteção de convulsões epilépticas.

### 1.3.1. Métricas consideradas

Neste trabalho prático, retiramos as nossas conclusões através das métricas:

- Accuracy
- Sensitivity
- Specificity

### 1.3.2. Predição de Convulsões

Paciente		54802		112502	
Encoder	Network	Treino	Teste	Treino	Teste

None	Feed Forward Net	Acc: 0.50 Sen: 0.01 Spe: 0.94	Acc: 0.88 Sen: 0.04 Spe: 0.96	Acc: 0.50 Sen: 0.00 Spe: 0.99	Acc: 0.99 Sen: 0.00 Spe: 0.99
	LayRec Net	Acc: 0.44 Sen: 0.99 Spe: 0.04	Acc: 0.09 Sen: 0.99 Spe: 0.02	Acc: 0.44 Sen: 0.99 Spe: 0.00	Acc: 0.01 Sen: 1.00 Spe: 0.00
	CNN	Acc: 1.00 Sen: 1.00 Spe: 1.00	Acc: 0.49 Sen: 0.44 Spe: 0.53	Acc: 0.56 Sen: 1.00 Spe: 0.21	Acc: 0.027 Sen: 0.98 Spe: 0.028
	LSTM	Acc: 0.78 Sen: 0.91 Spe: 0.68	Acc: 0.45 Sen: 0.48 Spe: 0.51	Acc: 0.57 Sen: 0.33 Spe: 0.79	Acc: 0.91 Sen: 0.00 Spe: 0.93
15	Feed Forward Net	Acc: 0.46 Sen: 0.11 Spe: 0.78	Acc: 0.61 Sen: 0.45 Spe: 0.65	Acc: 0.50 Sen: 0.01 Spe: 0.95	Acc: 0.98 Sen: 0.00 Spe: 0.99
	LayRec Net	Acc: 0.50 Sen: 0.00 Spe: 0.99	Acc: 0.79 Sen: 0.00 Spe: 0.99	Acc: 0.50 Sen: 0.00 Spe: 0.99	Acc: 0.99 Sen: 0.00 Spe: 1.00
	LSTM	Acc: 0.54 Sen: 0.95 Spe: 0.26	Acc: 0.40 Sen: 0.45 Spe: 0.41	Acc: 0.51 Sen: 0.00 Spe: 1.00	Acc: 0.99 Sen: 0.00 Spe: 0.99
10-3	Feed Forward Net	Acc: 0.49 Sen: 0.01 Spe: 0.93	Acc: 0.87 Sen: 0.05 Spe: 0.94	Acc: 0.50 Sen: 0.00 Spe: 0.99	Acc: 0.99 Sen: 0.00 Spe: 0.99
	LayRec Net	Acc: 0.50 Sen: 0.00 Spe: 1.00	Acc: 0.92 Sen: 0.00 Spe: 1.00	Acc: 0.50 Sen: 0.00 Spe: 1.00	Acc: 0.99 Sen: 0.00 Spe: 1.00
	LSTM	Acc: 0.50 Sen: 0.00 Spe: 1.00	Acc: 0.92 Sen: 0.00 Spe: 1.00	Acc: 0.50 Sen: 0.00 Spe: 1.00	Acc: 0.99 Sen: 0.00 Spe: 1.00

Tabela 2 - Resultados de Predição s/ Pós-Processamento

Paciente		54802		112502	
Encoder	Network	Treino	Teste	Treino	Teste
	Feed	Acc: 0.50	Acc: 0.91	Acc: 0.50	Acc: 0.99

None	Forward Net	Sen: 0.00 Spe: 0.96	Sen: 0.00 Spe: 0.99	Sen: 0.00 Spe: 0.99	Sen: 0.00 Spe: 0.99
	LayRec Net	Acc: 0.43 Sen: 0.99 Spe: 0.02	Acc: 0.08 Sen: 0.99 Spe: 0.01	Acc: 0.44 Sen: 1.00 Spe: 0.00	Acc: 0.00 Sen: 1.00 Spe: 0.00
	CNN	Acc: 0.82 Sen: 0.99 Spe: 0.69	Acc: 0.45 Sen: 0.47 Spe: 0.47	Acc: 0.44 Sen: 0.98 Spe: 0.06	Acc: 0.01 Sen: 1.00 Spe: 0.01
	LSTM	Acc: 0.79 Sen: 0.96 Spe: 0.67	Acc: 0.44 Sen: 0.50 Spe: 0.49	Acc: 0.52 Sen: 0.29 Spe: 0.79	Acc: 0.99 Sen: 0.00 Spe: 0.99
15	Feed Forward Net	Acc: 0.48 Sen: 0.05 Spe: 0.86	Acc: 0.67 Sen: 0.43 Spe: 0.70	Acc: 0.50 Sen: 0.00 Spe: 0.97	Acc: 0.99 Sen: 0.00 Spe: 0.99
	LayRec Net	Acc: 0.53 Sen: 0.00 Spe: 1.00	Acc: 0.86 Sen: 0.00 Spe: 0.99	Acc: 0.50 Sen: 0.00 Spe: 1.00	Acc: 0.99 Sen: 0.00 Spe: 1.00
	LSTM	Acc: 0.48 Sen: 0.98 Spe: 0.11	Acc: 0.35 Sen: 0.44 Spe: 0.35	Acc: 0.50 Sen: 0.00 Spe: 1.00	Acc: 0.99 Sen: 0.00 Spe: 1.00
10-3	Feed Forward Net	Acc: 0.50 Sen: 0.00 Spe: 0.95	Acc: 0.90 Sen: 0.00 Spe: 0.98	Acc: 0.50 Sen: 0.00 Spe: 1.00	Acc: 0.99 Sen: 0.00 Spe: 1.00
	LayRec Net	Acc: 0.50 Sen: 0.00 Spe: 1.00	Acc: 0.92 Sen: 0.00 Spe: 1.00	Acc: 0.50 Sen: 0.00 Spe: 1.00	Acc: 0.99 Sen: 0.00 Spe: 1.00
	LSTM	Acc: 0.50 Sen: 0.00 Spe: 1.00	Acc: 0.92 Sen: 0.00 Spe: 1.00	Acc: 0.50 Sen: 0.00 Spe: 1.00	Acc: 0.99 Sen: 0.00 Spe: 1.00

Tabela 3 - Resultados de Predição c/ Pós-Processamento

### 1.3.3. Detecção de Convulsões

Paciente		54802		112502	
Encoder	Network	Treino	Teste	Treino	Teste
	Feed	Acc: 0.50	Acc: 0.88	Acc: 0.50	Acc: 0.99

None	Forward Net	Sen: 0.03 Spe: 0.99	Sen: 0.08 Spe: 0.99	Sen: 0.00 Spe: 0.99	Sen: 0.00 Spe: 0.99
	LayRec Net	Acc: 0.44 Sen: 0.02 Spe: 0.99	Acc: 0.09 Sen: 0.05 Spe: 0.99	Acc: 0.44 Sen: 0.00 Spe: 0.99	Acc: 0.01 Sen: 0.00 Spe: 0.99
	CNN	Acc: 0.82 Sen: 1.00 Spe: 0.99	Acc: 0.49 Sen: 0.94 Spe: 0.96	Acc: 0.56 Sen: 0.66 Spe: 0.99	Acc: 0.03 Sen: 0.67 Spe: 0.99
	LSTM	Acc: 0.75 Sen: 0.99 Spe: 1.00	Acc: 0.45 Sen: 1.00 Spe: 0.94	Acc: 0.57 Sen: 0.70 Spe: 1.00	Acc: 0.92 Sen: 0.67 Spe: 0.99
15	Feed Forward Net	Acc: 0.46 Sen: 0.25 Spe: 0.99	Acc: 0.61 Sen: 0.31 Spe: 0.98	Acc: 0.50 Sen: 0.02 Spe: 0.99	Acc: 0.98 Sen: 0.00 Spe: 0.99
	LayRec Net	Acc: 0.50 Sen: 0.52 Spe: 0.95	Acc: 0.79 Sen: 0.44 Spe: 0.86	Acc: 0.50 Sen: 0.00 Spe: 1.00	Acc: 0.99 Sen: 0.00 Spe: 1.00
	LSTM	Acc: 0.54 Sen: 0.56 Spe: 0.99	Acc: 0.40 Sen: 0.86 Spe: 0.98	Acc: 0.51 Sen: 0.16 Spe: 1.00	Acc: 0.99 Sen: 0.00 Spe: 0.99
10-3	Feed Forward Net	Acc: 0.49 Sen: 0.06 Spe: 0.99	Acc: 0.87 Sen: 0.17 Spe: 0.99	Acc: 0.50 Sen: 0.00 Spe: 0.99	Acc: 0.99 Sen: 0.00 Spe: 1.00
	LayRec Net	Acc: 0.50 Sen: 0.00 Spe: 1.00	Acc: 0.92 Sen: 0.00 Spe: 1.00	Acc: 0.5 Sen: 0.00 Spe: 1.00	Acc: 0.99 Sen: 0.00 Spe: 1.00
	LSTM	Acc: 0.50 Sen: 0.00 Spe: 1.00	Acc: 0.92 Sen: 0.00 Spe: 1.00	Acc: 0.50 Sen: 0.00 Spe: 1.00	Acc: 0.99 Sen: 0.00 Spe: 1.00

Tabela 4 - Resultados de Detecção s/ Pós-Processamento

Paciente		54802		112502	
Encoder	Network	Treino	Teste	Treino	Teste
	Feed Forward Net	Acc: 0.50 Sen: 0.02 Spe: 0.99	Acc: 0.91 Sen: 0.07 Spe: 0.99	Acc: 0.50 Sen: 0.00 Spe: 1.00	Acc: 0.99 Sen: 0.00 Spe: 0.99

None	LayRec Net	Acc: 0.44 Sen: 0.02 Spe: 1.00	Acc: 0.08 Sen: 0.05 Spe: 0.99	Acc: 0.44 Sen: 0.00 Spe: 1.00	Acc: 0.05 Sen: 0.00 Spe: 0.99
	CNN	Acc: 0.81 Sen: 0.69 Spe: 0.99	Acc: 0.45 Sen: 0.78 Spe: 0.98	Acc: 0.44 Sen: 0.00 Spe: 1.00	Acc: 0.01 Sen: 0.00 Spe: 0.99
	LSTM	Acc: 0.76 Sen: 0.68 Spe: 0.99	Acc: 0.44 Sen: 0.56 Spe: 0.95	Acc: 0.57 Sen: 0.10 Spe: 1.00	Acc: 0.99 Sen: 0.00 Spe: 0.99
15	Feed Forward Net	Acc: 0.48 Sen: 0.25 Spe: 0.99	Acc: 0.67 Sen: 0.30 Spe: 0.99	Acc: 0.50 Sen: 0.02 Spe: 1.00	Acc: 0.99 Sen: 0.00 Spe: 1.00
	LayRec Net	Acc: 0.53 Sen: 0.51 Spe: 0.99	Acc: 0.86 Sen: 0.43 Spe: 0.94	Acc: 0.50 Sen: 0.00 Spe: 1.00	Acc: 0.99 Sen: 0.00 Spe: 1.00
	LSTM	Acc: 0.49 Sen: 0.28 Spe: 1.00	Acc: 0.35 Sen: 0.58 Spe: 0.99	Acc: 0.50 Sen: 0.00 Spe: 1.00	Acc: 0.99 Sen: 0.00 Spe: 1.00
10-3	Feed Forward Net	Acc: 0.50 Sen: 0.05 Spe: 0.99	Acc: 0.90 Sen: 0.17 Spe: 0.99	Acc: 0.50 Sen: 0.00 Spe: 1.00	Acc: 0.99 Sen: 0.00 Spe: 1.00
	LayRec Net	Acc: 0.50 Sen: 0.00 Spe: 1.00	Acc: 0.92 Sen: 0.00 Spe: 1.00	Acc: 0.50 Sen: 0.00 Spe: 1.00	Acc: 0.99 Sen: 0.00 Spe: 1.00
	LSTM	Acc: 0.50 Sen: 0.00 Spe: 1.00	Acc: 0.92 Sen: 0.00 Spe: 1.00	Acc: 0.50 Sen: 0.00 Spe: 1.00	Acc: 0.99 Sen: 0.00 Spe: 1.00

Tabela 5 - Resultados de Detecção c/ Pós-Processamento

## 1.4. Conclusões

De uma forma geral consideramos que os resultados são bastante positivos, tendo na mesma consciência que poderá estar a ocorrer algum overfitting em alguns casos, algo que poderia ser mitigado tendo um tratamento de dados mais cuidado. Além disso consideramos que há muito espaço para melhorias especialmente fazendo mais testes com mais epochs e com uma maior variedade parâmetros, por forma a garantir que os resultados são melhores e mais bem fundamentado (melhor tuning dos modelos treinados). No geral gostamos do trabalho e achamos bastante complexo e desafiante.