Seizure Prediction in Intracranial EEG Recordings

Ana Maria Sousa, Mariana Xavier, Rui Santos

Abstract — Epilepsy is a neurological disorder that can cause seizures. These seizures can be predicted by detecting preceding events that occur in the preictal state prior to a seizure. The biggest challenge in predicting seizures is to distinguish the normal interictal state from the preictal state of the brain from intracranial EEG recordings. This paper aimed at making this distinction in signals from humans and dogs, whose seizures are very similar. The data was filtered to remove unwanted frequencies in the EEG and the STFT was applied to transform the signals into the frequency domain. For the feature extraction a CNN was chosen, followed by several fully connected layers responsible for the classification task in one model and followed by a logistic regression in another. A balanced accuracy of 56,2% was attained in the test dataset for the first model and of 54,4 % for the second one.

Index Terms – epilepsy, seizure prediction, preictal state, interictal state, intracranial EEG, deep learning, logistic regression, CNN, Short Time Fourier Transform (STFT).

I. INTRODUCTION

Epilepsy is a neurological disorder caused by unusual nerve cell activity in the brain. This disorder causes recurrent seizures which may or may not go unnoticed and with various durations, depending on its severity [1]. Around 50 million people worldwide have epilepsy, making it one of the most common neurological diseases globally and it is estimated that up to 70% of people living with epilepsy could live seizure-free if properly diagnosed and treated [2].

Seizure prediction systems can help patients with epilepsy by identifying the periods which are most likely to be followed by seizures. Designing a device that could alert the patient to the high probability of a seizure could be a powerful tool for that patient, who could take measures to maintain his own safety, avoiding potentially dangerous activities like driving or swimming, and try to prevent the seizure from occurring via the administration of medication.

Electroencephalogram (EEG) signals are recorded to monitor the electrical activity inside the brain. These signals can be recorded by implantation of electrodes inside the brain tissues called intracranial EEG (iEEG) signals [3].

In case of a neurological disorder, an abrupt change in the electrical signals inside the brain can be observed through EEG recordings. This brain activity can be identified and divided into four different states: before the seizure (preictal), during the seizure (ictal), between seizures or baseline being the normal state of the brain (interictal) and after the seizure (post-ictal). Figure 1 shows the plot of three channels of recordings of EEG signals where these states are illustrated.

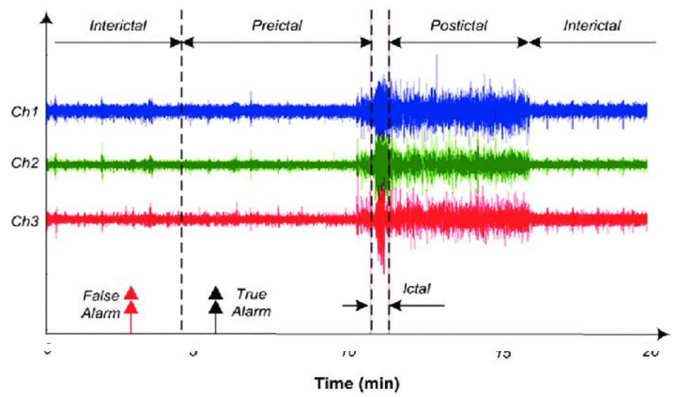

Figure 1 - Epileptic brain states [4].

Also, epilepsy is the most common neurological disorder seen in dogs and has been estimated to affect approximately 75% of the canine population [5]. Epilepsy in dogs and humans is similar enough that canine epilepsy research also has the potential to facilitate human epileptic research [6].

Therefore, the goal of this work was to identify with maximum accuracy the preictal state in dogs and humans with naturally occurring epilepsy.

II. METHODS

A. Dataset

This work considers the dataset from the American Epilepsy Society Seizure Prediction Challenge database [7].

It contains intracranial EEG signals recorded from two different patients. These recordings have a varying number of electrodes (15 for one patient and 24 for the other) and a sampling frequency of 5000 Hz.

In addition, it also presents iEEG signals recorded from four dogs with naturally occurring epilepsy using an ambulatory monitoring system. The signals were recorded with a sampling frequency of 400 Hz and from 15 or 16 electrodes, depending on the dog.

The training set is organized into ten-minute EEG clips labeled "Preictal" for pre-seizure data segments, or "Interictal" for non-seizure data segments. Training data segments are numbered sequentially, while testing data are in random order.

Nevertheless, the dataset was already divided into training and testing. The training dataset presented 204 clips for preictal state and 2962 segments for interictal state while the test set presented 194 clips for preictal state and 2751 clips for interictal state. Therefore, the dataset is extremely unbalanced, which had to be taken into consideration.

Each file contained information regarding the data of the EEG of the different channels, the name of the channels used, the sampling frequency, the length of the signal in seconds and the identifying number of the signal in the recorded sequence.

B. Preprocessing

For preprocessing the data, the different channels were considered as representative of the brain status. Therefore, they were extracted separately from each file. Also, the sampling frequency was retrieved from the file.

With this separation in different channels, we ended up with 50588 clips for training and 47934 clips for testing.

Then, the data was filtered using three different butterworth filters: a 59-61 Hz band-stop filter of second order to remove power line noise, a 120 Hz low-pass filter and a 0.5 Hz high-pass filter to remove the dc component as proposed in [8].

Furthermore, the Short Time Fourier Transform (STFT) was used to transform the signals from the time domain into the frequency domain. The STFT was chosen because it gives better results for signals with non-stationary nature like EEG signals since it conserves the short duration changes in the signals. Therefore, the STFT was applied to a non-overlapping window of 30 seconds as proposed in [9].

C. Undersampling

Due to the limited time and resources for the development of this work, there was the need to perform undersampling in order to decrease the amount of data clips to process.

Therefore, a random undersampling was performed to the data files of each patient or dog, keeping the ratio preictal:interictal signals equal to 1:2 in each one of them.

This operation was performed over the training dataset, having ended up with 10080 files for training (3360 preictal and 6720 interictal).

D. Convolutional Neural Network

Convolutional Neural Networks (CNN) have been showing promising results in the field of signal analysis, especially in the frequency domain [10]. This type of network is capable of performing an automatic feature extraction through the application of several convolutional filters, decreasing processing time and complexity.

The CNN implemented in this work consists of 3 convolutional layers with increasing number of filters (16, 32 and 64). Kernel sizes are 3x3, 3x3 and 5x5, respectively. Zero-padding is performed prior to convolutions for two reasons:

i) to avoid loss of information in the edges;
ii) to avoid size reduction during convolutions since input dimensions are only 3360x21.

A 2x2 max-pooling is performed after each convolutional layer to reduce dimensionality. This also reduces the computational cost and introduces some translation invariance. Rectified Linear Unit (ReLU) activation function is used for all convolutional layers. Batch normalization is performed before the first convolutional layer to stabilize and speed up the training process.

E. Classification

To perform the classification task, the output from the convolutional layers were passed through two different models for further comparison. The output from the convolutional layers was flattened and used as input for:

1. two hidden layers with decreasing number of neurons - 128 and 64, respectively. An output layer with 2 neurons followed. ReLU activation function was used for both hidden layers and a Log-Softmax activation function was used in the output layer.

2. a logistic regression process in order to be classified as one of the two classes.

F. Training parameters

A batch size of 32 was selected for the training process. For training, the Adam optimizer was used with a learning rate of 1e-4 and L2-regularization with a weight decay of 1e-2 was applied to decrease overfitting during the training process.

Negative log likelihood (NLL) loss function was selected to compute the loss after each batch applying different costs to each class in order to counteract data imbalance. The costs were defined as an inverse of the frequency of each class as:

$$costs = [\frac{batch\ size}{2 * interictal} \quad \frac{batch\ size}{2 * preictal}]$$

The final optimal number of epochs given by early stopping was 42, comparing the validation accuracy in each epoch (validation segments chosen randomly from the training set).

The flowcharts of both models and a schematic representation of the convolutional neural network adopted for this work are present in the appendix I and II, respectively.

III. Results

Before reaching the first model (without logistic regression) previously described different models with distinct characteristics were tested in 25% of the training data which was set aside in order to define the best model for the task at hand. The different attempts and its results are summarized in the following sections.

A. *Fighting Data Imbalance*

To fight data imbalance present in the dataset, two approaches were tested:

1. Introducing weights in the negative log likelihood loss in order to penalize more the errors in the preictal class.
2. Performing another undersampling, keeping the ratio preictal:interictal equal to 1:1, therefore eliminating the data imbalance but also losing more data.

The weights for the first approach were first defined as:

$$weights = [\frac{batch\ size}{2 * interictal} \quad \frac{batch\ size}{2 * preictal}]$$

These weights were chosen since they give each class a weight which is inversely proportional to its frequency in the dataset. Since the ratio preictal:interictal is 1:2, the weight for preictal class will be equal to 66% and 33% for the interictal class.

Also, in case these weights weren't enough for the task, the values [0.2 0.8] were also tested, giving even a higher importance for the errors in the preictal class.

This comparison was performed with a model without any of these transformations.

For these comparisons, the performance of the four models (without weights, with calculated weights, with defined weights, with undersampling) was compared in terms of balanced accuracy, preictal accuracy and interictal accuracy.

The results obtained can be seen in the following figures:

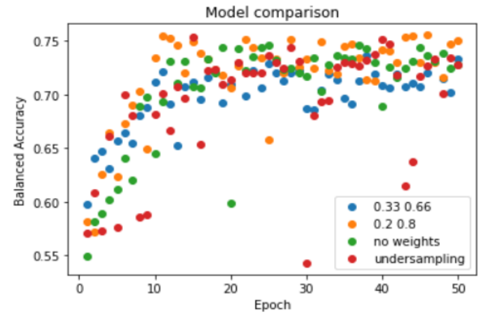

Figure 2 - Balanced accuracy in the test set for the four models.

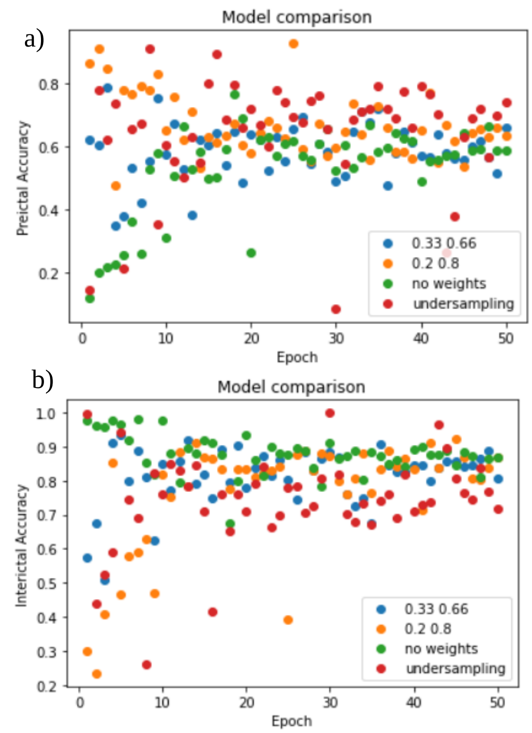

Figure 3 – a) Preictal and b) interictal accuracies in the test set for the four models.

From these figures we can conclude that the weights defined as [0.2 and 0.8] provide a higher balanced accuracy. However, looking at the separate accuracies, the classes do not look as evenly balanced. This fact made us discard this option.

Moving to the next higher balanced accuracy we find the model in which was performed a random undersampling. Despite its relatively good accuracies, this model leads to losing even more signals from the dataset, which was unwanted.

Ending up with the model without weights or the one with weights derived from the frequency of each class, the second one brings an advantage since it is able to slightly compensate for the present imbalance. Therefore, we moved on with this model.

B. Filter Size Selection in CNN

Smaller filters allow to differentiate objects from smaller and more local features. Therefore, its sizes are crucial in the feature extraction stage.

Up until now, we were using a decreasing size of filters: 5x5 in the first convolutional layer and 3x3 in the following two layers. However, we decided to test the influence of the size, making another model whose order was inverted: 3x3 in the first two convolutional layers and 5x5 in the last one, so that the model started with learning more local features and only then broadening for less local ones.

For the comparison we took again 25% of the training set aside, to then use as a test set and check the model's accuracy.

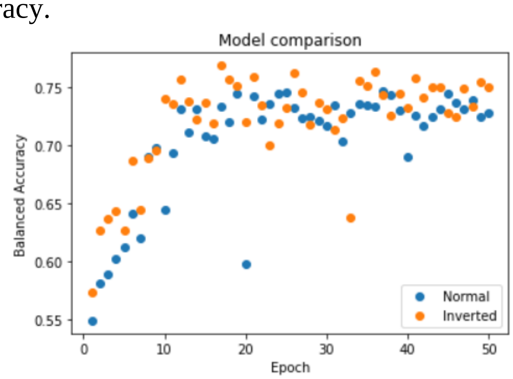

Figure 4 - Balanced accuracy in the test set for the comparison with the inverted model.

From observing the image above, it is now clear that using the smaller filters first is advantageous since the local features are crucial to the task. Therefore, the inverted model was chosen.

C. Number of Linear Layers

To evaluate the influence of the linear layers on the accuracy of the training set, two models were evaluated.

The first model was the one with two linear layers with 64 and 2 neurons, respectively.

The second model added a third linear layer, having 128, 64 and 2 neurons in each consecutive layer.

Increasing the number of linear layers from two to three showed some improvements in the test accuracies of the model, as they became slightly higher but more stable with the increasing epochs.

Therefore, this number of layers was adopted for further studies.

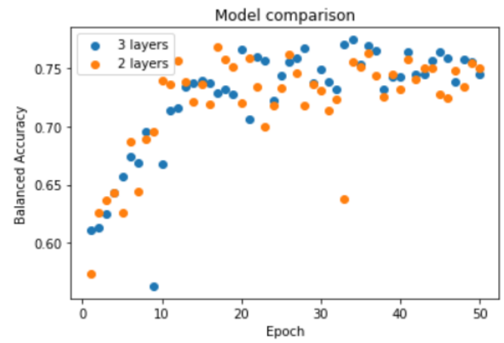

Figure 5 - Balanced accuracy in the test set for the comparison between the models with 2 and 3 fully connected layers.

D. Checking for underfitting or overfitting

At this stage it was crucial to evaluate if the model was either underfitting or overfitting. We used 75% of the training set of the model to actually train the model and 25% to evaluate it, determining the balanced accuracies for each one of the sets.

The accuracies for training and testing the model are present in the figure below:

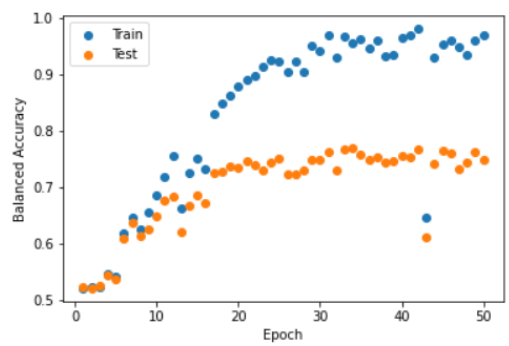

Figure 6 - Balanced accuracy in the training and testing phases.

On the one hand, for underfitting the accuracy in the training set should be lower than in the test set, since the model was not able to fit well to the data it was receiving. On the other hand, for overfitting the accuracy in the training set should be higher than in the test set, since the model was fitting too well the training data and it wasn't able to generalize what it was learning.

It is clear now that the model is overfitting, due to the huge differences observed in the accuracies. Also, this overfitting is also due to the preictal class.

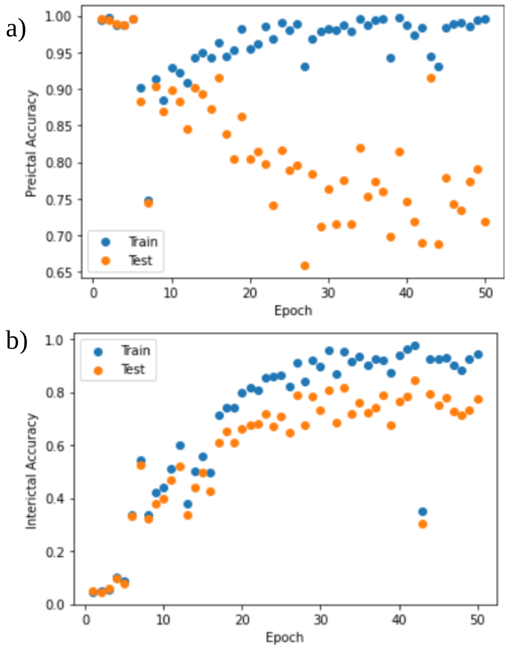

Figure 7 – a) Preictal and b) interictal accuracies in the training and testing phases.

E. Influence of dropout

To combat overfitting, we studied two different approaches of regularization. The first was dropout which consists of restricting the model by randomly removing neurons during training. The number of nodes that are discarded during dropout is defined by the parameter p. This parameter represents the probability of element being set to zero, the meaning of dropout.

We tested different values for p {0.1; 0.3; 0.5} and applied the technique only after each convolutional layer. The results were compared with the model used up until this stage.

Although the purpose of applying this regularization technique was to reduce the generalization error, it was found that dropout impaired the model's performance without bringing benefit in the reduction of overfitting. As this technique implies a reduction in the capacity of the network, this can be harmful in the case of neural networks that are not large enough, which justifies the reduction in the performance of the model which can be observed in the figure 8. Therefore, this approach was discarded.

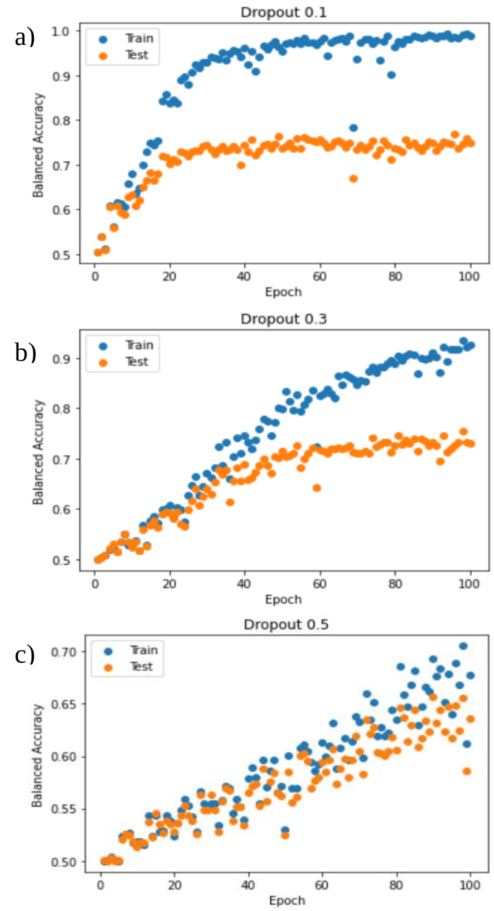

Figure 8 – Dropout a) 0.1, b) 0.3 and c) 0.5 influence on the training and testing accuracies.

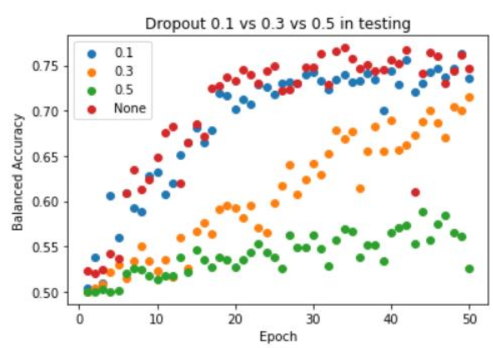

Figure 9 – Dropout comparison over the test accuracies.

F. L2 regularization

The final attempt to combat overfitting is based on L2 regularization, which was applied to the loss function through the weight decay, forcing the weights to be smaller.

The optimal value obtained for weight decay was 0.01, which allowed a closer similarity for the accuracies for more epochs by improving the accuracies in the test set. Although the problem of overfitting was not fixed, this approach was considered for the final model due to the mentioned reason.

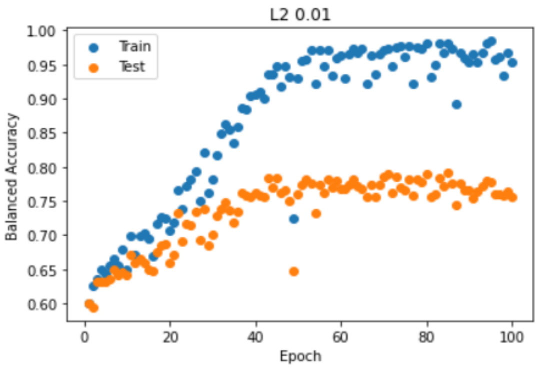

Figure 10 – Influence of L2 regularization.

G. *Final Model Performance*

Before assessing test accuracy, we trained the proposed models. 20% of the training set was set aside to check for validation accuracy at every epoch of the training process. An early-stopping strategy was adopted, in which we saved the best model and if no improvement was made in the following 10 epochs the training process would cease. The best model was obtained for epoch 42, as shown in the image below.

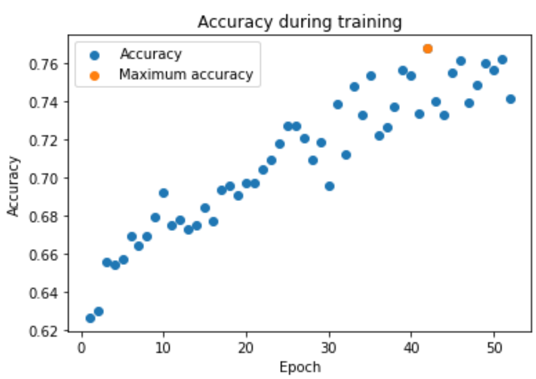

Figure 11 - Early-stopping strategy and accuracies.

After training, we computed the test accuracy. Test data underwent the same preprocessing as the train set before use. Then, each signal passed through the network and the model predictions were obtained. Since original labels are attributed to iEEG signals composed of several channels and not a single channel, predictions corresponding to the different channels of the same iEEG segment were combined by using the majority of those predictions for each segment.

For the first model (without logistic regression), a final balanced accuracy of 56,2% was obtained and an accuracy of 38,7% and 73,8% for preictal and interictal, respectively.

For the logistic regression model, a final balanced accuracy of 54,4% was obtained, and an accuracy of 35,1% and 74,4% for preictal and interictal, respectively.

Since the presented accuracies are close, the two models behave in a similar way in the task of classification.

In addition, for a better analysis of the model (which presented better balanced accuracy), we represented the results using a confusion matrix present on appendix III. Attached we can also find a more exhaustive analysis of the results for the same model in the form of a table with various statistical values such as recall and precision (appendix IV).

IV. Conclusion

Despite providing a starting point for seizure prediction, the model here proposed had unsatisfactory results. Several improvements can be done to improve model performance, namely the inclusion of more training data and the oversampling of the minority class to eliminate data imbalance. Also, since some channels of the preictal iEEG signals might be located in non-focal areas, these might act as confounding samples that increase the intraclass variability, decreasing the capacity of the model to learn from these signals. Therefore, more information about seizure heterogeneity and pathophysiology should be considered upon dataset selection and model development.

References

[1] Pietrangelo A., Morrison J. *Everything you need to know about Epilepsy*. Healthline (2018)

[2] *Epilepsy*. Word Health Organization (2019)

[3] M. Le Van Quyen, J. Martinerie, V. Navarro, P. Boon, M. D'Havé, C. Adam, B. Renault, F. Varela, M. Baulac. *Anticipation of epileptic seizures from standard EEG recordings*. Lancet, vol. 357, no. 9251, pp. 183–188 (2001).

[4] Rasekhi J., Mollaei M., Bandarabadi M., Teixeira C., Dourado A. *Epileptic Seizure Prediction based on Ratio and Differential Linear Univariate Features*. Journal of Medical Signals and Systems (2015).

[5] Heske L., Nodvedt A., Jaderlund K.H., Berendt M., Egenvall A.. *A cohort study of epilepsy among 665,000 insured dogs: incidence, mortality and survival*. Vet J, 202:471-6 (2014).

[6] Potschka H., Fischer A., von Rüden E.-L., Hülsmeyer V., Baumgärtner W. *Canine epilepsy as a translational model?* Epilepsia. 54:571-579. (2013)

[7] American epilepsy society seizure prediction challenge. USA. Available at: https://www.kaggle.com/c/seizure-prediction

[8] P. Mirowski, D. Madhavan, Y. LeCun, R. Kuzniecky. *Classification of patterns of EEG synchronization for seizure prediction*. *Clin*. Neurophysiology, vol. 120, no. 11, pp. 1927–1940 (2009).

[9] Syed M.U., Shehzad K., Muhammad H.A. *Epileptic Seizures Prediction Using Deep Learning Techniques*. IEEE Access, vol. 8, pp. 39998-40007 (2020).

[10] Zhou Mengni, Tian Cheng, Cao Rui, Wang Bin, Niu Yan, Hu Ting, Guo Hao, Xiang Jie. *Epileptic Seizure Detection Based on EEG Signals and CNN*. Frontiers in Neuroinformatics, vol. 12 (2018).

APPENDIX

I – Flowcharts of the two models.

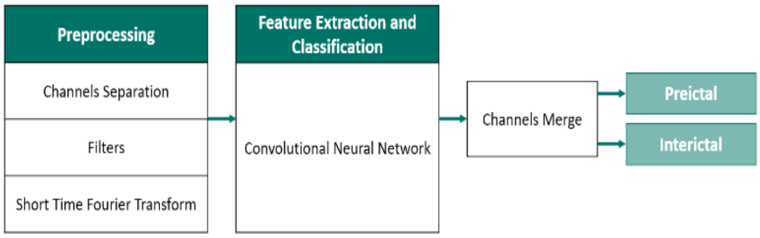

Figure 1 – Flowchart for the first model (without logistic regression).

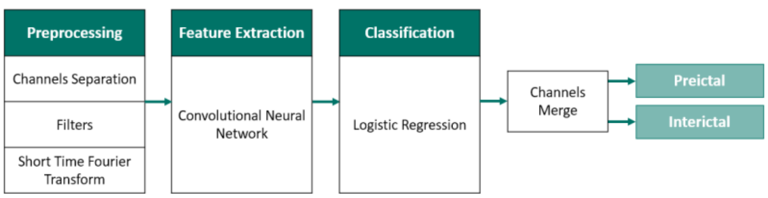

Figure 2 – Flowchart for the second model (logistic regression).

II. Schematic representation of the CNN.

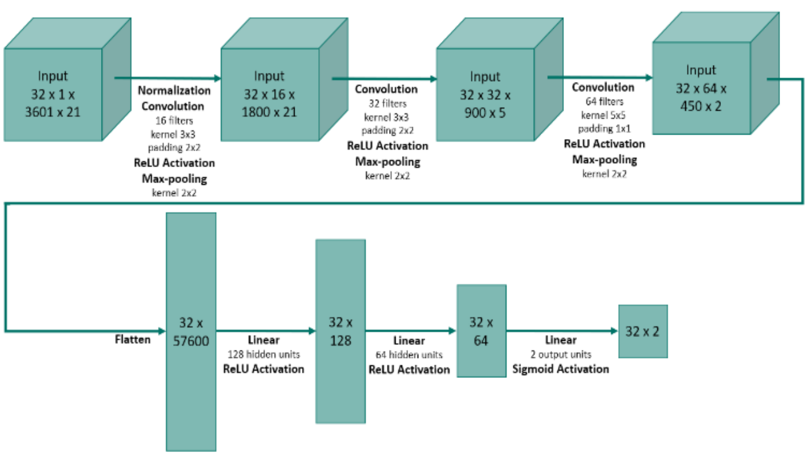

Figure 1 – Schematic for the first model (without logistic regression). For the logistic regression one, only the steps until "Flatten" (included) are considered.

III – Confusion matrix for the first models.

	Predicted	
Actual	Interictal	Preictal
Interictal	2029	722
Preictal	119	75

Figure 1 – Flowchart for the first model (without logistic regression).

IV - Statistical analysis.

	precision	recall	f1-score	support
0	0.94	0.74	0.83	2751
1	0.09	0.39	0.15	194
accuracy			0.71	2945
macro avg	0.52	0.56	0.49	2945
weighted avg	0.89	0.71	0.78	2945

Figure 1 – Statistical analysis for the first model (without logistic regression).