Prediction of Epileptic Seizures using RealAdaBoost and ANN

Manuel Montoya Catalá [1]

[1] Master in Multimedia and Communications

mmontoya@ing.uc3m.es

Abstract

In this paper, a system for the prediction of Epileptic Seizures is proposed. The system uses the EEG signals of a dog containing samples with interictal periods (those with absence of seizures) and ictal episodes. A wide variety of features is obtained both in the time and the frequency domains from the time sequences that contains each sample vector, then feature selection and feature extraction techniques are used to improve the results. The classifier for the seizure detection a RealAdaBoost ensemble method along with an Artificial Neural Network trained with a variation of the Extreme Learning Machine algorithm as a weak learner. The system compares the performance of the different features and also performs feature and extraction to get the optimal parameters. El evaluation is performed using the hard probability of error and the area under de curve.

Index Terms: Seizure Detection, Artificial Neural Networks, Extreme Learning Machine, Boosting, RealAdaBoost, EEG.

1. Introduction

Epilepsy is a group of neurological disorders characterized by epileptic seizures. Approximately 50 million people worldwide have epilepsy, making it one of the most common neurological diseases globally. About 20% of epileptic patients do not respond to treatments based drugs and continue to experience seizures. For these patients responsive neurostimulation represents a possible therapy capable of aborting seizures before they affect a patient's normal activities.

In order for a responsive neurostimulation device to successfully stop seizures, a seizure must be detected and electrical stimulation applied as early as possible. A seizure that builds and generalizes beyond its area of origin will be very difficult to abort via neurostimulation. Current seizure detection algorithms in commercial responsive neurostimulation devices are tuned to be hypersensitive, and their high false positive rate results in unnecessary stimulation. In this paper we describe a detection system for seizures occurring in dogs.

This project is based on the Lab Session 2 – Prediction of Epileptic Seizures from the subject Applications of Signal Processing [1]. This paper proposes a more flexible a powerful classifier, a RealAdaBoost ensemble using an Artificial Neural Network trained with a variation of the Extreme Learning Machine algorithm for boosting. It also obtains more features and used machine learning techniques to select and extract the best features.

The dataset is composed of intracranial EEG recorded from dogs with naturally occurring epilepsy using an ambulatory monitoring system. EEG was sampled from 16 electrodes at 400 Hz, and recorded voltages were referenced to the group average.

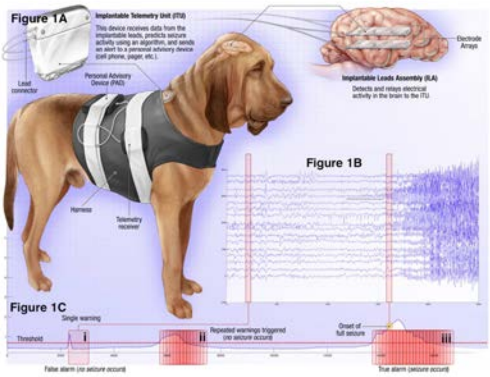
Figure 1A
Implantable Telemetry Unit (ITU)
This device receives data from the implantable leads, predicts seizure activity using an algorithm, and sends an alert to a personal advisory device (cell phone, pager, etc.).
Lead connector
Personal Advisory Device (PAD)
Electrode Arrays
Implantable Leads Assembly (ILA)
Detects and relays electrical activity in the brain to the ITU
Harness
Telemetry receiver
Figure 1B
Figure 1C
Single warning
Threshold
i
ii
iii
Repeated warnings triggered (no seizure occurs)
Onset of full seizure
False alarm (no seizure occurs)
True alarm (seizure occurs)

Figure 1 Scheme of the dataset obtaining process

This paper will start describing the features obtained from the dataset and their importance, since this paper is focused on the classifier, the boosting ensemble will be revisited next, followed by basic introduction to the ANNs and the ELM algorithm. The variation of the ELM algorithm for boosting has been personally derived since it couldn't be found in the literature.

Once all the theoretical background is explained, the experimental setup is described. The parameters of the classifier will be discussed along with some properties of the dataset that are valuable for the tuning and validation of these parameters. Finally, the evaluation of the results, conclusions and future work will be exposed.

2. Features obtained

Seizures can take on many different forms, and seizures affect different people in different ways. Anything that the brain does normally can also occur during a seizure when the brain is activated by seizure discharges. Some people call this activity “electrical storms” in the brain. Seizures have a beginning, middle and end.

Let $x_i(t)$ y $x_j(t)$ be the time sequence of any of the 16 chnanels. Every time sequence contains 400 samples, obtained at 400 Hz rate, so every sequence represents one second. The simplest feature obtained is the total power of the signal for every channel. Then, more complex features were obtained.

The figure below shows a representation of the EEG signals sampled from the 16 electrodes during the seizure of a dog.

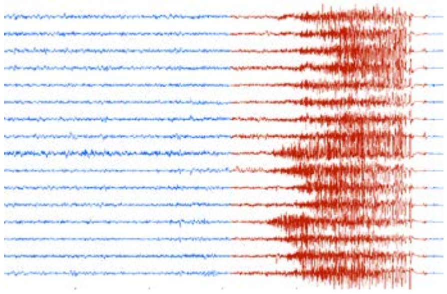

Figure 2 Seizure example.

2.1. Temporal Features Obtained

Taking a look to Figure 2, we made the initial assumption that the correlation among channels decreases during the seizure. The features obtained were:

1) Correlation between the 16 channels:

$$\rho_{ij} = \frac{c_{ij}}{\sigma_i \sigma_j} = \frac{E\{(x_i(t) - \mu_i)(x_j(t) - \mu_j)\}}{\sigma_i \sigma_j}$$

Since the correlation matrix is symmetric, the features obtained are $N(N+1)/2 = 136$.

2) Eigenvalues of the covariance matrix:

$$\bar{\lambda}_i \bar{v}_i = \bar{\bar{\Sigma}}_{ij} \bar{v}_i$$

One feature per channel is obtained, 16 features in total.

2.2. Frequency Features Obtained

During an epileptic seizure, the speed of the EEG signals increase, although the total energy of the signal may remain unchanged, the distribution of the energy along the frequencies may vary. The Power spectral density along different frequency ranges was obtained according the the standard denomination for brain waves:

- δ (delta): 0.1 - 4 Hz
- θ (theta) 4 - 8 Hz
- α (alpha) 8 - 12 Hz
- β (beta) 12 - 30 Hz,
- Low-γ (gamma) 30 - 70 Hz
- High- γ (gamma) 70 - 180 Hz

One feature per channel is obtained at each frecuency range. In total $\mathrm{N} \cdot 6 = 96$ features. Finally, there are 240 features in total per sample vector. To reduce the dimensionality of the problem, feature selection techniques have been used. First, non-informative features are discharged using a Random Tree, then RBE crossvalidation is used to obtain the most relevant features. These are the final features used for classification.

3. Boosting

Boosting is an ensemble technique used for creating a strong classifier as a linear combination of weak classifiers. Given a set of T weak classifiers $\mathrm{h_t}(\cdot)$, $\mathrm{t} = 1,2 \ldots \mathrm{T}$, the output of the whole system is obtained as:

$$H(\bar{x}_i) = sign(f(\bar{x}_i)) = sign\left(\sum_{t=1}^{T} \alpha_t \, h_t(\bar{x}_i)\right)$$

These weak classifiers $\mathrm{h_t}(\cdot)$ are trained in a sequential manner, each weak learner focuses on a different region of the input space, this is achieved using a weight vector $\bar{\mathrm{D}}$ during the training phase. This $\bar{\mathrm{D}}$ is a discrete distribution vector over the training samples space that tells the weak learner how much importance it has to give to every training sample. Samples that are poorly classified will have big weight values so the weak learner will focus on them.

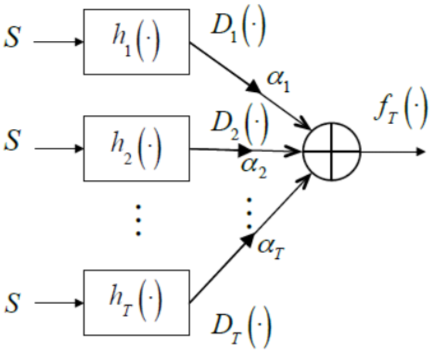

Figure 3 Structure of a Boosting Ensemble.

The emphasis $\bar{\mathrm{D}}$ is updated at every weak learner $\mathrm{h_t}(\cdot)$ so that the next weak learner $\mathrm{h_{t+1}}(\cdot)$ focuses in the samples that have been poorly classified previously. $\bar{\mathrm{D}}$ is only used for training, affecting to the cost function of the weak learner, not its actual output.

The general implementation of boosting follows the next definition. Given a training set $\{\bar{\mathrm{x}}_\mathrm{i}, \mathrm{y_i}\}^\mathrm{N}$ where $\bar{\mathrm{x}}_\mathrm{i} \in \bar{\mathrm{X}}$ and $\mathrm{y_i} \in \{-1,1\}$. Initialize the weight distribution uniformly $D_0(i\,) = 1/M$. For every weak learner $t = 1, \ldots, T$:

1) Train Weak learner using distribution $\bar{D}_t$
2) Get output of weak learner for every sample in the training set $h_t: \bar{X} \rightarrow R$.
3) Choose the linear combination constant α_t for this learner.
4) Update the Samples Distribution using the following equation:

$$D_{t+1}(i) = \frac{D_t(i) e^{-\alpha_t \, h_t(\bar{x}_i) y_i}}{Z_t}$$

Where $\mathrm{Z_t}$ is a normalization factor.

Boosting's goal is to minimize the exponential loss function, which is minimized when the margin is also minimized

$$E = \sum_{i=1}^{M} e^{-y_i f(\bar{x}_i)} = \sum_{i=1}^{M} e^{-y_i \sum_{t=1}^{T} \alpha_t \, h_t(\bar{x}_i)}$$

Due to the Update rule of the weight vector $\bar{D}$, the training error probability is upper bounded by:

$$P_e \leq \prod_{t=1}^{T} Z_t$$

So the main objective of every weak learner should be minimizing its own particular and independent $\mathrm{Z_t}$. Note that $\mathrm{Z_t}$ depends on the weight vector $\mathrm{\bar{D}_t}$ and on the output of the current weak learner $h_t(\bar{x}_i)$

$$Z_t = \sum_{i=1}^{M} D_t(i) e^{-\alpha_t\, h_t(\bar{x}_i) y_i}$$

There are many implementations of boosting, this paper uses 2 approaches: RealAdaBoost and GentleBoost.

3.1. RealAdaBoost

This implementation chooses the de-emphasis $\mathrm{\alpha_t}$ that minimizes an approximation of the normalization constant Z_t at every iteration. No matter what the output of the weak learner is $h_t(\bar{x}_i)$.

$$\alpha_t = \frac{1}{2} ln\left(\frac{1+r}{1-r}\right)$$

Being r the expected margin over the distribution $\bar{D}$.

$$r = E\{margin\}_{\bar{D}_t} = E\{\, h_t(\bar{x}_i) y_i\}_{\bar{D}_t} = \sum_{i=1}^{M} D_t(i)\, h_t(\bar{x}_i) y_i$$

The upper bound approximation of $\mathrm{Z_t}$ used to derive this value of $\mathrm{\alpha_t}$ imposes that $\mathrm{h_t(\bar{x}_i) \in \{-1,1\}}$ but it finds a good $\mathrm{\alpha_t}$ no matter the weak learner $\mathrm{h_t(\bar{x}_i)}$.

3.2. GentleBoost

Instead of trying to minimize $\mathrm{Z_t}$ for a given $\mathrm{h_t(\bar{x}_i)}$ using $\mathrm{\alpha_t}$, its weak learners directly try to minimize the Taylor approximation of $\mathrm{Z_t}$:

$$Z_t = \sum_{i=1}^{M} D_t(i) e^{-\alpha_t\, h_t(\bar{x}_i) y_i} \propto \sum_{i=1}^{M} D_t(i)\, (h_t(\bar{x}_i) - y_i)^2$$

The cost function of the weak learner must be:

$$C = \sum_{i=1}^{M} D_t(i)\, (h_t(\bar{x}_i) - y_i)^2$$

The advantages of this method is that we don't have to calculate $\mathrm{\alpha_t}$, it is implicit in the weak learner $\mathrm{h_t(\bar{x}_i)}$, also, $\mathrm{h_t(\bar{x}_i)}$ can have any value, it is not bounded by $\mathrm{h_t(\bar{x}_i) \in \{-1,1\}}$. On the other hand, this technique does not ensure a good linear combination of the weak learners; a bad weak learner will hurt the overall combination. Weak learners are forced to use the MSE cost function.

4. Artificial Neural Networks

An Artificial Neural Network is a machine learning system inspired by the architecture of the mammals' brain. They are composed by a mesh of interconnected artificial neurons (perceptron), simple systems that map an input vector $\bar{x}$ to an output o. An ANN is characterizes over 3 main properties: the Neuron Model, the Architecture, and the Learning Algorithm.

The Neuron Model is the model that defines each of the neurons of the ANN. The most used model is a system that outputs a transformed linear combination of its input:

$$o = f_A(Z) = f_A(\bar{X} \cdot \bar{W} + b) = f_A\left(\sum_{i=1}^{N} (w_i \cdot x_i) + b\right)$$

The linear combination of the input Z is called the activation value of the neuron; each neuron also has a input bias that does not depend on any other input. The transformation function $\mathrm{f_A}$ is called the activation function; it is usually a sigmoid function. The image below show a graphical representation of the system:

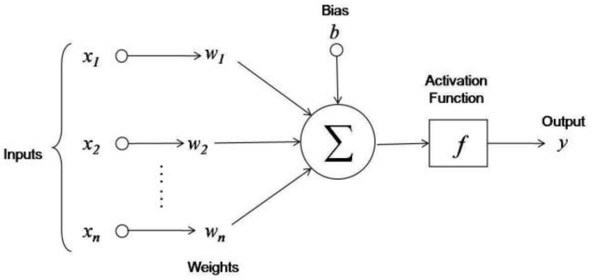

Figure 4 Neuron Model

The Architecture of the ANN defines the number of neurons it has and the interconnections among them. Usually, neurons are clustered into groups called layers; all the neurons in a layer have the same neuron model and interconnection geometry. We can differentiate 3 main kinds of layers; the input layer, in which the input of its neurons is the input of the system, the output layer which neurons have the output of the system as output and the hidden layer, which inputs and output are only connected to other neurons.

In this paper a Single Layer FeedForward Network is used, these layers only have one hidden layer and there are no loops in the neurons interconnections. The image below shows an example of this structure

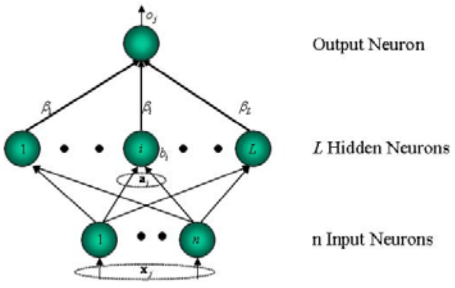

Figure 5 Single Layer FeedForward Network.

The SLFN architecture has the parameters $\{\bar{\bar{W}}, \bar{b}, \bar{\beta}\}$. Given a set of input samples $\bar{\bar{X}}$, the output o_{ij} of every hidden neuron to every input sample can be expressed into a matrix H, called the hidden output matrix:

$$H(\bar{\bar{X}}, \bar{\bar{W}}, \bar{b}) = \begin{bmatrix} o_{1,1} & \cdots & o_{1,L} \\ \cdots & \cdots & \cdots \\ o_{N,1} & \cdots & o_{N,L} \end{bmatrix}_{NxL}$$

Every row represents the output of the hidden neurons for a given input sample, o_{ij} is the output of the $j-th$ hidden neuron for the $i - th$ input sample. The total output of the system $\bar{O}$ can be obtained as:

$$\bar{\bar{H}}_{N\text{xL}}\,\bar{\beta}_{\text{L}x1} = \bar{O}_{Nx1}$$

The Learning Algorithm is the algorithm that tunes the parameters of the model $\{\bar{\bar{W}}, \bar{b}, \bar{\beta}\}$ so that the output of the system $\bar{O}_{Nx1}$ is the best possible estimation of the target output $\bar{T}_{Nx1}$. The learning algorithm is defined by a cost function that usually depends on the difference between $\bar{O}_{Nx1}$ and $\bar{T}_{Nx1}$ and an algorithm that tunes the ANN parameters in order to decrease the cost function. The error metric used for most algorithms is the Mean Square Error whose cost function is:

$$E_{MSE} = \sum_{i=1}^{M} \varepsilon_i^2 = \sum_{i=1}^{M} (o_i - t_i)^2$$

We can write this error in a vector form as:

$$E_{MSE} = \bar{\varepsilon}^t \cdot \bar{\varepsilon} = (\bar{O} - \bar{T})^t (\bar{O} - \bar{T})$$

4.1. Extreme Learning Machine Algorithm

The ELM algorithm is a fast learning algorithm for SLFNs, basically it chooses the input weight parameters $\{\bar{\bar{W}}, \bar{b}\}$ at random and then computes the output weights vector $\{\bar{\beta}\}$ analytically as the Least Square Solution of the system:

$$\bar{\bar{H}}\bar{\beta} = \bar{T} \quad \text{with} \quad \bar{\bar{H}} = f_A(\bar{\bar{X}} \cdot \bar{\bar{W}})$$

We have to find a $\hat{\beta}$ so that:

$$\|H\hat{\beta} - T\| = \min_{\beta} \|H\beta - T\|$$

Every neuron can be seen as a hyperplane and its output, the projection of the samples over it. So this algorithm can be interpreted as a generation of random orthogonal hyperplanes, transformed by a sigmoid function and linearly combined by $\bar{\beta}$ as the least square solution of the system with respect to the desired output $\bar{T}$. The analytical equation for $\bar{\beta}$ is:

$$\hat{\beta} = H^{\dagger} T \quad \text{where} \quad H^{\dagger} = (\bar{\bar{H}}^t \bar{\bar{H}})^{-1} \bar{\bar{H}}$$

The derivation of the analytical equation of $\bar{\beta}$ is as follows: Given a training set $\{\bar{x}_i, t_i\}^N$ where $\bar{x}_i \in X^d$ and $t_i \in \{-1,1\}$. We have $\bar{T}$ the vector of desired outputs of the training set, $\bar{O}$ the vector of outputs of the system and $\bar{\varepsilon}$, the MSE error between the previous two.

$$\bar{\varepsilon} = \begin{bmatrix} \varepsilon_0 \\ \vdots \\ \varepsilon_M \end{bmatrix} \quad \bar{O} = \begin{bmatrix} o_0 \\ \vdots \\ o_M \end{bmatrix} \quad \bar{T} = \begin{bmatrix} t_0 \\ \vdots \\ t_M \end{bmatrix}$$

Since $\bar{O} = \bar{\bar{H}}\bar{W}_o$ we have the square error to be:

$$E_{MSE} = (\bar{\bar{H}}\bar{W}_o - \bar{T})^t (\bar{\bar{H}}\bar{W}_o - \bar{T})$$

$$E_{MSE} = (\bar{\bar{H}}\bar{W}_o)^t (\bar{\bar{H}}\bar{W}_o) - 2 \cdot (\bar{\bar{H}}\bar{W}_o)^t \bar{T} + \bar{T}^t \bar{T}$$

This error is a convex function of $\bar{W}_o$ so we take derivative with respect to $\bar{W}_o$ and equal to 0:

$$\frac{\partial E_{MSE}}{\partial \bar{W}_o} = (\bar{\bar{H}})^t (\bar{\bar{H}}\bar{W}_o) - 2 \cdot (\bar{\bar{H}})^t \bar{T} = 0$$

Resolving the equation we obtain:

$$(\bar{\bar{H}}^t \bar{\bar{H}}) \bar{W}_o = \bar{\bar{H}}^t \bar{T}$$

$$\bar{W}_o = (\bar{\bar{H}}^t \bar{\bar{H}})^{-1} \bar{\bar{H}} \cdot \bar{T} = \bar{\bar{H}}^{+} \bar{T}$$

With $\bar{\bar{H}}^{\dagger}$ the Moore-Penrouse inverse of $\bar{\bar{H}}$.

4.2. ELM for boosting.

Since we are using boosting, we have to modify the ELM algorithm in order to include the Sample Distribution D_t so that the error cost function is:

$$E_{MSE} = \sum_{i=1}^{M} D_i \cdot \varepsilon_i^2 = \bar{D} \circ (\bar{\varepsilon}^t \cdot \bar{\varepsilon})$$

Where the symbol ∘ means element-wise multiplication. Since we don't know the matrix properties of this operation, we express $\bar{D}$ as a diagonal matrix Λ that has D as diagonal:

$$\bar{\bar{\Lambda}} = \begin{pmatrix} D_0 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & D_M \end{pmatrix} \quad \bar{D} = diag(\bar{\bar{\Lambda}})$$

Therefore we can express the error as:

$$E_{MSE} = \bar{\varepsilon}^t \cdot \bar{\bar{\Lambda}} \cdot \bar{\varepsilon}$$

Moreover, the LS solution for $\bar{W}_o$ is derived as follows:

$$E_{MSE} = (\bar{\bar{H}}\bar{W}_o - \bar{T})^t \, \bar{\bar{\Lambda}} \, (\bar{\bar{H}}\bar{W}_o - \bar{T})$$

$$E_{MSE} = (\bar{\bar{H}}\bar{W}_o)^t \bar{\bar{\Lambda}} (\bar{\bar{H}}\bar{W}_o) - 2 \cdot (\bar{\bar{H}}\bar{W}_o)^t \bar{\bar{\Lambda}} \bar{T} + \bar{T}^t \bar{\bar{\Lambda}} \bar{T}$$

This error is a convex function of $\bar{W}_o$ so we take derivative with respect to $\bar{W}_o$ and equal to 0:

$$\frac{\partial E_{MSE}}{\partial \bar{W}_o} = (\bar{\bar{H}})^t \bar{\bar{\Lambda}} (\bar{\bar{H}}\bar{W}_o) - 2 \cdot (\bar{\bar{H}})^t \bar{\bar{\Lambda}} \bar{T} = 0$$

Resolving we obtain:

$$(\bar{\bar{H}}^t \bar{\bar{\Lambda}} \bar{\bar{H}}) \bar{W}_o = \bar{\bar{H}}^t \bar{\bar{\Lambda}} \bar{T}$$

$$\bar{W}_o = (\bar{\bar{H}}^t \bar{\bar{\Lambda}} \bar{\bar{H}})^{-1} \bar{\bar{H}} \bar{\bar{\Lambda}} \cdot \bar{T} = \bar{\bar{H}}^{+} \cdot \bar{T}$$

As we can see, the equation $\bar{W}_o$ for the Boosting ELM is quite similar to the normal ELM but it needs to be derived properly.

5. Experimental Setup

The system proposed performs seizure detection over the dataset given in the second laboratory session of [1], obtained from the Kaggle competition [2]. The dataset is composed of 1280 sample vectors containing 400 variables. These are temporal samples from the EEG electrodes obtained at 400 Hz rate, so every sequence represents one second. The dataset is unbalanced with ration 1:2.6 and divided into two sets, 800 sample vectors for training and 480 for testing.

The system has two parameters to validate, the number of hidden neurons of the weak learner Nh and the number of weak learners T. A stratified K-fold was used for the tuning of these parameters.

Also, in order to see the importance of the different set of features, these were evaluated separately and in groups, the combination that performed the best was the one obtained after performing feature selection over the initial 240 features. The features selected correspond mainly to the power features; correlation features were very powerful but caused overfitting and therefore were excluded by the recursive backward elimination algorithms which use crossvalidation.

6. Results and Evaluation

In order to evaluate the performance of the proposed system, we use two measures, the hard probability of error and the area under the ROC curve (AUC), since the system outputs a soft prediction, we can obtain the AUC for different thresholds. The following results show the value of these performance measures for different sets of features:

1) Power features

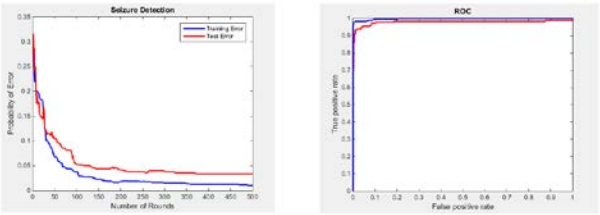

Figure 6 Results for the power features

These features scored a probability of error of 0.033 and an AUC of 0.9783 for testing.

2) Correlation features

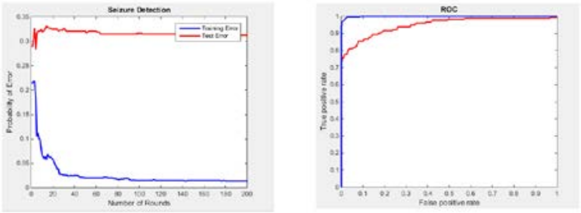

Figure 7 Results for the correlation features

These features scored a probability of error of 0.3 and an AUC of 0.9783 for testing. On the other hand the training error and ROC were perfect, which means the systems overfitted the training data.

3) Combined features

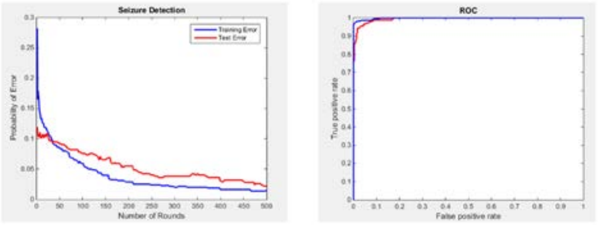

Figure 8 Results for the combined features

The combined and selected features scored a probability of error of 0.02 and an AUC of 0.9850 for testing which is an improvement over the previous features meaning that features have complementary information and that learning algorithms work better with less features.

7. Conclusions and Future Work

As the outstanding results show, the problem is highly separable, obtaining training errors of 0.02 using the by the ensemble ANN + ELM + Adaboost proposed, this system has better performance than the SVM, which scores a testing error of 0.05. The system is very fast and scalable; it takes very few seconds for both training and testing. Power features are the most important ones since they generalize pretty well; on the other hand, correlation features are very discriminative but cause overfitting. There are many future works to be done with this project concerning all of their stages:

- Use transfer learning to predict seizures in a different dog.
- Reduce de number of time samples of the initial features in order to predict as soon as possible the seizure.
- Implement feature extraction techniques.
- Try out other classifiers such as Random Forest and more ensemble methods.

8. References

[1] Lab Session 2 – Prediction of Epileptic Seizures from C4.278.8931-1 APPLICATIONS OF SIGNAL PROCESSING 14/15-S1.

[2] Kaggle. https://www.kaggle.com/c/seizure-prediction

[3] Epilepsy. http://www.who.int/mediacentre/factsheets/fs999/en/

[4] Extreme learning machine: Theory and applications. Guang-Bin Huang, Qin-Yu Zhu, Chee-Kheong Siew.

[5] Improved Boosting Algorithms Using Confidence-rated Predictions. Robert E. Schapire Yoran Singer.