

Epileptic Seizure Detection by Means of Genetically Programmed Artificial Features

Hiram Firpi
Michigan State University
2120 Engineering Building
East Lansing, MI 48824 USA
1 (517) 355 3796
hfirpi@ieee.org

Erik Goodman
Michigan State University
2120 Engineering Building
East Lansing, MI 48824 USA
1 (517) 355 6453
goodman@egr.msu.edu

Javier Echaz
BioQuantix Corp.
1000 Abernathy Rd. #1250
Atlanta, GA 30328 USA
1 (770) 399 9930
echaz@ieee.org

ABSTRACT

In this paper, we describe a general-purpose, systematic algorithm, consisting of a genetic programming module and a k -nearest neighbor classifier to automatically create *artificial features*—features that are computer-crafted and may not have a known physical meaning—directly from the reconstructed state-space trajectories of the EEG signals that reveal patterns indicative of epileptic seizure onset. The algorithm was evaluated in three patients and validation experiments were carried out using 267.6 hours of EEG recordings. The results with the artificial features compare favorably with previous benchmark work that used a handcrafted feature.

Categories & Subject Descriptors

I.5.2 [Computing Methodologies]: Pattern Recognition: Design Methodology – *feature evaluation and selection, pattern analysis*

General Terms: Algorithms, Design

Keywords: Epilepsy, seizure detection, feature extraction, state-space reconstruction, genetic programming

1. INTRODUCTION

Since the invention of the electroencephalograph, great progress has been made in studying many brain disorders. One of the most puzzling disorders is epilepsy, a neurological condition that makes people susceptible to brief electrical disturbances in the brain, thus producing a change in sensation, awareness, and/or behavior. Epilepsy is characterized by recurrent seizures. It affects up to one percent of the population of the world or sixty million people, and 25% cannot be fully controlled by current medical or surgical treatment.

Many approaches have been proposed to extract information from EEG signals that can be used to develop algorithms to predict or detect epileptic seizures [1], [3], [4], [8], [10], [13]. In this work, in particular, we are interested in the problem of seizure onset detection. Although, easier than the prediction problem, detection of epileptic seizures is still far from trivial. There are many

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'05, June 25–29, 2005, Washington, DC, USA.
Copyright 2005 ACM 1-59593-010-8/05/0006...\$5.00.

anomalies that occur naturally in EEG signals that cause detectors to fire and declare that a seizure is occurring when actually it is not. False starts, delta trains, and spike-and-wave discharges all lead to false alarms and thus medicating the patient unnecessarily. Nonetheless, seizures must be detected as soon as possible so the medication can be delivered immediately to control the seizure without further consequences. Therefore, there is a tradeoff between the number of false alarms and the number of false negatives (when the detector says an attack is occurring when it is not).

Some approaches that apply digital signal processing or filter theory have acceptable performance. However, to extract the relevant information that can facilitate such prediction or detection, features are calculated using conventional techniques and methodologies that are time-consuming, trial-and-error processes requiring a great deal of effort from researchers. All of these conventional techniques rely on knowledge of a feature formula or algorithm that may have been obtained from intuition, tradition, the physics of the problem, analogies to problems in other fields, etc. There is no guarantee that any of these conventional features extracts maximally relevant information from the raw data. The present work seeks to develop an algorithm that systematically and automatically can find or generate artificial features or patterns starting from raw data—in this case, EEG signals. Artificial features are defined as features that are computer-designed, are learned, inductive, optimized, and are designed based on given data (*e.g.*, EEG data). We will see that unlike most approaches, this work is not based solely on tuning a few parameters of fixed terms to find the “best” detector for a patient, but instead uses an algorithm that also changes the structure of the equations, giving enough flexibility to design a highly tuned feature that is sensitive to the characteristics or patterns of a given patient.

In Section 2, we explain the methodology used to reconstruct the state-space trajectory from the EEG signal. We also explain the components that constitute the genetic programming artificial features (GPAF) algorithm. In Section 3, we present the results of the experiments on data from three patients. Sections 4 and 5 present discussions and conclusions.

2. METHODOLOGY

We attempt to capture a “pocket” of deterministic dynamics of EEG signals by means of delay-embedding in a stream of sliding windows. First, we reconstruct the state-space trajectory of the EEG signals using the standard delay-embedding scheme [12].

Later, this reconstructed trajectory is input into a genetic programming algorithm, which attempts to find the pattern(s) giving the best discrimination between baseline data (nonseizure, see Figure 1) and ictal (during seizure, see Figure 2) data in the sense of a minimum-error-risk objective function. A universal classifier then performs the categorization task. Figure 3 is a diagram depicting all components of the algorithm.

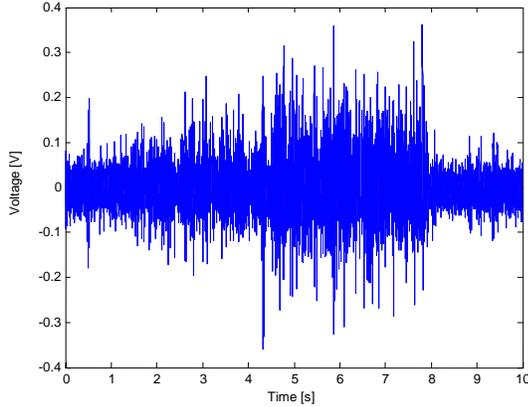


Figure 1. Illustration of a ten-minute duration baseline signal.

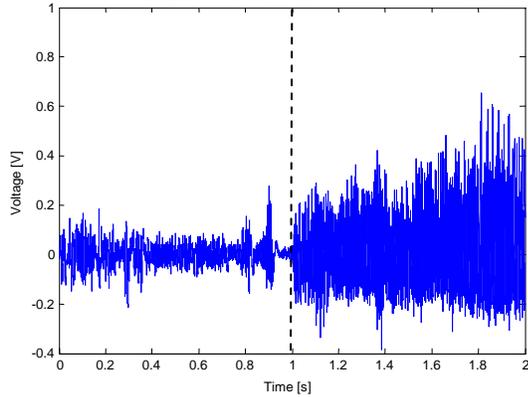


Figure 2. Illustration of a two-minute ictal signal. The unequivocal electroencephalographic onset (UEO) is located at one minute (dashed line).

2.1 EEG Data

The anonymized EEG data used for these experiments were obtained from a tripartite database from Georgia Institute of Technology, Emory University, and University of Pennsylvania. The EEG data were recorded from epileptic patients undergoing a pre-surgical evaluation. Patients were simultaneously videotaped during their hospital stays, which varied in duration from 4 to 11 days. The EEG signals were recorded at 200Hz, with 12-bit resolution.

In this work, we deal with *mesial temporal lobe epilepsy* (MTLE). MTLE is strongly associated with complex partial seizures, the most common type of seizure, being present in 40% of all epilepsy cases reported [2].

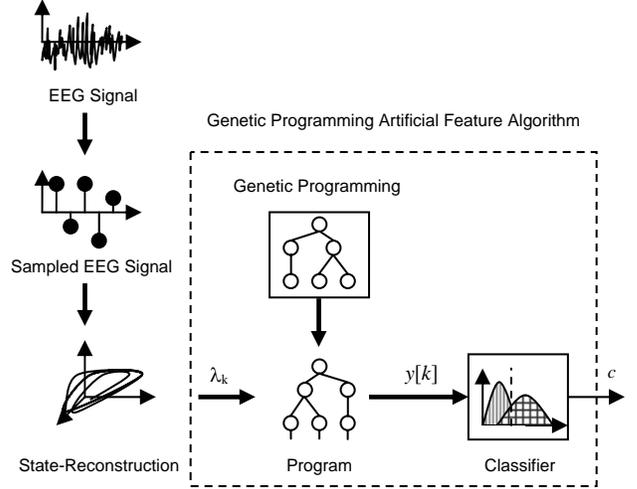


Figure 3. State-space reconstruction and the components of the artificial feature algorithm.

2.2 State reconstruction via delay-embedding

In the past, various authors have labeled the dynamics governing the brain as chaotic, and thus have applied nonlinear dynamic tools to analyze EEG data [6], [9]. Chaos theory [11] states that within a chaotic system, that is, one displaying apparently disordered random-like data, an underlying order exists. Because of this, one of the most relevant properties is that precise long-term prediction is impossible; however, prediction in the short term and with an error allowance is possible. Such a property allows us to reconstruct the state-space trajectory of an attractor of the system (in this case, the model of the brain that generates the EEG signals). We can reconstruct the deterministic component of the state trajectory of the EEG signals by taking previous samples of the observable output and creating an artificial state vector with n_e elements, which we denote λ_k (input to the program in Figure 3), the embedding vector. This process creates a diffeomorphism (a function that is a smoothly distorted copy of the original trajectory and preserves dynamic and geometric qualities of the trajectory of the EEG system). In other words, we reconstruct the set of dynamic variables that governs the dynamics of the system that in this work is the model of the brain that produces the EEG signals.

Given such state-space trajectory reconstruction, our approach is to input these pseudo-state vectors (*i.e.*, λ_k), evolving in time, to the genetic programming (GP) module, and by means of the algorithm, to find a transformation, usually nonlinear, that achieves the maximum separability between baseline and ictal data. In other words, the GP algorithm combines the inputs (states) in a (non)linear way and outputs a function that is meant to separate the baseline and ictal classes such that the performance of the classifier is either better than or at least equivalent to categorizing the classes with no transformation or with a benchmark algorithm previously defined. The following expression mathematically describes the result:

$$y_i[k] = \phi_i(\lambda), \quad (1)$$

where ϕ_i is a transformation function (artificial feature or program) designed by the GP algorithm, i is the artificial feature index (it is equal to 1 if the GP module is designing just one artificial feature), and

$$\lambda = \{x(n), x(n-\tau), x(n-2\tau), \dots, x(n-(n_e-1)\tau)\} \quad (2)$$

is the set of delayed samples (inputs) that the GP will use in any combination (with replacement) to construct the artificial features. The parameter τ is the delay time, which will be determined using the autocorrelation function, and n_e is the embedding dimension. Therefore, the artificial feature is a function defined as $\phi_i: \mathbb{R}^{n_e} \rightarrow \mathbb{R}$. Here we heuristically set the embedding dimension to 6.

As for any conventional feature, such as Fourier transform coefficients, signal energy, etc., in the genetic programming artificial features (GPAF) algorithm, we process the data as if viewed through a sliding observation window. The observation window is defined by two parameters: length of the window, denoted L , and displacement, denoted D , alternatively describable as the percentage of overlap of adjacent windows. The length L defines the number of points that will be evaluated (or analyzed) at any one time. D is defined as the number of new points that will be used in the next evaluation, or $D = L - O$, where O is the overlap with the previous window. To reduce variability in the feature time series, we use a summation in each window position. Recalling (2) and placing the summation operator, the resultant equation is stated in (3).

$$y_i[k] = \sum_{n=1+D(k-1)}^{D(k-1)+L} \phi_i(x(n), x(n-\tau), \dots, x(n-(n_e-1)\tau)) \quad (3)$$

The $y_i[k]$ is the GP artificial feature of the EEG signal $x(t)$, termed here the *artificial feature time series*, the subscript i denotes the number of the feature (*i.e.*, we can have a single artificial feature or a vector of such features), n is the index that controls the displacement inside the sliding window, and k is the discrete time unit of this time-series (or the index that indicate the next sliding-window position). The sliding window observes all L points in window-position k , whereas index n is delimited by $1 + (k-1)(L-D) \leq n \leq k(L-D)+1$ (the superscript and subscript on the summation symbol). In this work, we selected n to be increased one at a time. Figure 4 shows an illustration, which present a sliding window being slid through an EEG segment.

In other words, equation (3) is proportional to the average of the GPAF-processed points observed through the sliding window, where the argument of the summation $\phi(\cdot)$ (*i.e.*, the GPAF function) is found by the GP algorithm. That is, at each window position k , L points will be reduced, by means of (3), to a single point at the output. Thus, if we have a signal that contains P_{Tot} points and we set displacement to D , the number of output points after the entire signal is processed by the GP artificial feature is

$$\kappa = \frac{P_{Tot}}{D} - \frac{L}{D} + 1, \quad (4)$$

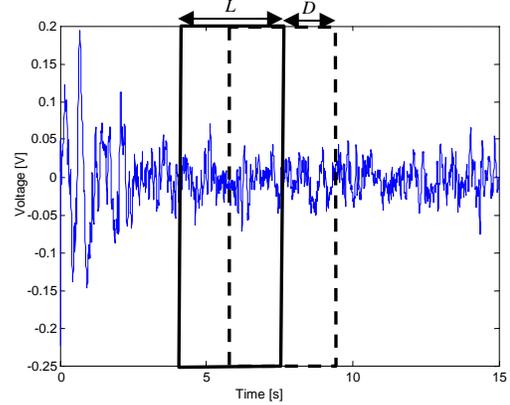


Figure 4. The sliding window slides (solid rectangle) over an EEG signal. The dashed rectangle indicates the position where the window will be displaced. The sliding window position is controlled by k whereas the shifting inside the sliding window is controlled by the index n . L is the length of the window and D is displacement.

where the term $-L/D + 1$ is an adjustment for the beginning and end of the signal, where the window of length L is not complete and, thus, some points are not calculated. Thus, the discrete index of the artificial time series k goes from 1 to κ .

2.3 Genetic programming

Genetic programming (GP) [7], established formally by Koza, is closely related to genetic algorithms (GA), although it has a few critical differences. In GP, the length of the chromosome is variable and the representation is typically formulated using trees instead of the strings used in GA. Another important aspect of GP, unlike GA, in which chromosomes often directly encode the solution to the problem, is that the tree typically provides a program that is used to solve the problem, or instructions for how to construct a solution. In GP, there are two types of nodes: functions, which have some number of arguments that they operate on, and terminals.

On the other hand, similarly to GA, GP evolves programs by means of the genetic operators selection, crossover, reproduction, and mutation. Table 1 shows the function set or building blocks used, which the GP algorithm will use to construct the programs (artificial features), in this work. The maximum tree depth was set to 10 and the population size to 1000. The initial population was generated using the ramped half-and-half method. The GP algorithm used a crossover operator with a rate of 0.9, fitness proportional selection, and a breeding operator with a rate of 0.1. The GP algorithm was extended to allow it to evaluate not just one tree per individual, but multiple trees per individual (*i.e.*, a forest), allowing us to generate a prespecified number of features for a patient simultaneously. All the trees of the individual were evaluated at the same time; however, there was no crossover among trees (features) of the same individual— crossover was done only among homologous trees.

Table 1. Function set for the GP algorithm

+	-	÷	×
cos	sin	log ₂	log ₁₀
ln	√	() ²	abs

2.4 Objective Function

The error risk is the metric to be optimized [5], which assigns risk factors $r > 0$ to the errors such that their relative costs can be accounted for as

$$R_E = P_{FN} P(S_T) r_{FN} + P_{FP} P(NS_T) r_{FP}, \quad (5)$$

where P_{FN} is the probability of false negatives (*i.e.*, when the algorithm misses a seizure), P_{FP} is the probability of false positives, and $P(S_T)$ and $P(NS_T)$ are the prior probabilities of the respective classes (seizure and baseline). Additionally, r_{FN} is a risk factor associated with missing seizures, and r_{FP} is a risk factor associated with declaring false positives. The relative sizes of these risk factors should be chosen as if assuming that the priors are equal $P(S_T) = P(NS_T)$. From experience, we select $r_{FN} = 0.75$ and $r_{FP} = 0.25$, yielding

$$R_E = (0.75)P_{FN} + (0.25)P_{FP}. \quad (6)$$

2.5 Classifier

Although any classifier in the GPAF algorithm may be used, in this work we select the classifier with one aspect in mind: to prescribe a general-purpose algorithm that automatically creates artificial features from raw data (or conventional features if those are the starting-point data). Therefore, we selected the k -nearest neighbor classifier (k -NN) as the classifier component for the GPAF algorithm (last stage in Figure 3). This classifier is nonparametric, nonlinear, and capable of producing multiple thresholds or complicated decision boundaries, making it suitable for n -dimensional, multi-modal problems. In addition, the training process is relatively easy, simply having the classifier store all the training data.

Holdout method was used in the classification phase in order to train and test the GPAF algorithm. The experiments were conducted using a point basis, where a feature point counts as one example. However, statistics reported on a point basis are not easy to interpret, therefore results for testing data will be reported using a block basis (where a whole signal epoch counts as one example). A threshold will be set for this transformation. If the point basis classification result is equal to or exceeds the threshold, the epoch will be labeled ictal; otherwise, it will be categorized as baseline (nonseizure).

2.6 Decision Integration Window

The experiments were conducted using a point basis, where each point in an artificial feature epoch counts as one example, *i.e.*, each point κ is an example for the classifier. However, if, for instance, an implantable device (with GPAF features integrated) classified each t seconds over an incoming EEG signal, the device could not be allowed to drug the patient based on the decision that the classifier makes each t seconds. The device needs a longer,

fixed-length window, so that it can observe the past evolution of the point-basis classification during a defined period to decide whether or not a patient is suffering a seizure. We denote the length of this window as L_{DIW} . This parameter controls a tradeoff between the number of false positives and the detection delays of seizures (thus, it also controls the number of seizures that are detected; however, because most seizures are detected sooner or later given the sudden and large changes on the EEG signal’s amplitude, what is also of great interest is how early those seizures can be detected). This parameter can be fixed as a constant for all patients, that is, the same L_{DIW} value for all the patients. However, this parameter gives flexibility to our approach; therefore, to make the system more “personal” it is better to tune the parameter for each patient. Here we selected L_{DIW} by “educated guess” and trial-and-error, but the ability of the GP algorithm to automatically select the optimal parameter will be investigated in the future.

3. RESULTS

In the training phase of the experiments, we used baseline and ictal epochs of 2 minutes duration, thus containing 24,000 points each. For ictal epochs, we used 1 minute of data before the UEO and 1 minute of data after it (2 min). The training set for each patient was three baseline epochs and three ictal epochs. The length of the sliding window was set to $L = 200$ (1 s) and $D = 50$ (0.25 s). Therefore, the number of points after an epoch is processed by the GP equation κ was 477 points per epoch. As stated before, the embedding dimension was set to $n_e = 6$. Euclidean distance was selected as the metric and number of nearest neighbors for the k -NN classifier was set to $k = 5$, a value commonly used. In these experiments, we set the number of trees to 2; that is, the GPAF algorithm created two artificial features for each patient.

3.1 Patient A

Patient A was diagnosed as having all seizures coming from the right inferior temporal neocortex. This patient had 5 seizures during 46.19 hours of recording. The delay time τ was set to $\tau = 11$ (0.055 s), selected from the first zero-crossing of the autocorrelation plot. Thus, the terminal set (the pseudo state-space vector) for this patient is $\{x(n), x(n-11), x(n-22), x(n-33), x(n-44), x(n-55)\}$. Equation (7) shows the artificial features found by the GPAF algorithm. Table 2 shows the results obtained from the validation data, where FP denotes the number of false positives, FPH is the rate of false positives per hour, ADEO denotes the average delay of the detection time with respect to the unequivocal electrographic onset (in seconds), FN is the number of false negatives, and HP is the number of hours of EEG recordings processed for each patient. The length of the DIW was set to 3.75 s.

$$y_1[k] = \sum_{n=50k-49}^{50k+150} \log_{10} \left(x(n-11) - x(n-22) + \sqrt{x(n-11)} \right) \quad (7)$$

$$y_2[k] = \sum_{n=50k-49}^{50k+150} \log_2 \left(\frac{\log_2(x(n-11))}{x(n-33) + x(n-44)} \right)$$

3.2 Patient B

Patient B was diagnosed with multifocal seizures. This patient had 10 seizures during 65.7 hours of EEG recording. The delay time τ was set to $\tau = 19$ (0.095 s). Thus, the terminal set for this patient was $\{x(n), x(n-19), x(n-38), x(n-57), x(n-76), x(n-95)\}$. The length of the DIW was set to 3 s. Equation (8) shows the artificial feature found by the GPAF algorithm. Table 2 shows the results obtained from the validation runs.

$$y_1[k] = \sum_{n=50k-49}^{50k+150} \left(x(n) - \frac{\sqrt{x(n-76)}}{x(n-95)} \right)^2 \quad (8)$$

$$y_2[k] = \sum_{n=50k-49}^{50k+150} \left| \frac{(x(n-19) - x(n))^2}{\log_{10}(\log_{10}(x(n-38)))} \right|$$

3.3 Patient C

Patient C was diagnosed as having all seizures coming from the right inferior frontal region. This patient had 11 seizures in 156.4 hours of recording. The delay time τ was set to $\tau = 9$ (0.045 s). Thus, the terminal set (the pseudo state-space vector) for this patient was $\{x(n), x(n-9), x(n-18), x(n-27), x(n-36), x(n-45)\}$. The length of the decision integration window was set to 2.75 s. Equation (9) shows the artificial feature found by the GPAF algorithm. Table 2 shows the results obtained from the validation run.

$$y_1[k] = \sum_{n=50k-49}^{50k+150} |x(n-9) - x(n)| \quad (9)$$

$$y_2[k] = \sum_{n=50k-49}^{50k+150} |x(n)|$$

Table 2. GPAF detection results

Patient	FP	FPh	ADEO	FN	HP
A	1	0.022	1.15 s	0	46.2
B	0	0	1.93 s	0	65.7
C	0	0	2.1 s	0	156.4
Average	0.33	0.007	1.73 s	0	268.29

Table 3. Line length detection results

Patient	FP	FPh	ADEO	FN	HP
A	4	0.087	-3.44 s	0	46.2
B	0	0.000	0.32 s	0	65.7
C	0	0.000	2.73 s	0	156.4
Average	1.33	0.029	-0.13 s	0	268.29

4. DISCUSSION

To evaluate how the GPAF is doing, we cite the results of work reported by Esteller *et al.* [3]. The authors used a handcrafted feature called line length, stated in (10).

$$LL(k) = \frac{1}{K} \sum_{n=k-L+1}^k |x(n-1) - x(n)| \quad (10)$$

Using this feature, the authors processed the same data that we did. Table 3 shows the results obtained by using the line length equation. Despite the equation's simplicity, the results produced by the feature were good enough to be implemented in an implantable detection device. Comparing Tables 2 and 3, for patient A, the GPAF features performed better than the line length feature by having just one false positive and thus a lower false positive rate. The detection delay was a little higher than that for the line length feature.

With the second patient, the performances of both algorithms were very similar, with the line length feature doing a little better on the detection delay average. Finally, for the third patient, the GPAF equations produced a lower average detection delay. For this patient, we remark that the artificial feature $y_1[k]$ in (9) is the conventional feature line length in (10). The better performance is because of the additional feature and a more flexible tuning of the L_{DIW} parameter, allowing only one false positive.

5. CONCLUSION

It is noted from inspection of the artificial feature formulas for each patient that, although relatively simple, they are far from trivial and not likely to come from intuition or knowledge of the physics of the problem. The artificial features designed by the GPAF algorithm are "optimized" for each particular patient based only on their available raw EEG recordings. Although not by much, these features match or exceed in some aspects the performance of traditional conventional features. However, it is noted that further experiments are planned to further improve the artificial features and, it is expected, exceed the performance of conventional features. Nonetheless, these results make the GPAF algorithm a worthy subject of further research. Additionally, we plan to use more patients and also to design a general (cross-patient) model for detection.

We need to highlight that there is tradeoff between the number of false positives and how early seizures are detected. If we desire a minimum of false positives, we can increase the value of L_{DIW} , but an increase in the average delay of detection will also result. If a low false positive rate and low average delay are desired, more training data must be fed to the GPAF algorithm in order to design a set of features that better detect seizures, rather than being sometimes triggered by artifacts or overfitting of data. Additionally, a term that weights more specifically the average delay could be also added to the cost function so the algorithm can seek to minimize average delay at the same time that is minimizing false positive and false negative rates.

Additionally, this work has shown—specifically in the results for patient C—that the GPAF algorithm is not only able to create artificial features without any obvious physical meaning, but also is able to find features with known physical meaning. Although

unintentionally, this work confirms the claim by Esteller *et al.* in the sense that the line length feature is a good and efficient feature for detection of epileptic seizure onsets.

6. REFERENCES

- [1] Chen, H., Zhong, S., and Yao D. Detection singularity value of character wave in epileptic EEG by wavelet. *IEEE International Conference on Communications, Circuits and Systems and West Sino Expositions 2002*, vol. 2, (29 Jun.-1 Jul., 2002) 1094-1097.
- [2] D'Alessandro, M. *The Utility of Intracranial EEG Feature and Channel Synergy for Evaluating the Spatial and Temporal Behavior of Seizure Precursors*. Ph.D. Dissertation, Georgia Institute of Technology, Atlanta, GA, 2001.
- [3] Esteller, R., Echauz, J., Tchong, Litt, B., and Pless, B. Line length: An efficient feature for seizure onset detection. *2001 Proceedings of the 23rd Annual EMBS International Conference*, (2001), 1707-1709.
- [4] Esteller, R., Vachtsevanos, G., Echauz, J., D'Alessandro, M., Bowen, C., Shor, R., and Litt, B. Fractal dimension detects seizures onset in mesial temporal lobe epilepsy. *Proceedings of the First Joint BMES/EMBS Conference: Serving Humanity, Advancing Technology*, (Atlanta, GA, Oct. 13-16, 1999), 442.
- [5] Firpi, H. *Genetically Found, Neurally Computed Artificial Features with Applications to Epileptic Seizure Detection and Prediction*. Master's Thesis, University of Puerto Rico-Mayagüez, Mayagüez, PR, 2001.
- [6] Jansen, B. H. Quantitative analysis of electroencephalograms: Is there chaos in the future?. *International Journal of Biomedical Computing*, vol. 27, pp. 95-123, 1991.
- [7] Koza, J. R. *Genetic Programming: On Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press, 1992.
- [8] Niederhoefer, C., Gollas, F., Chernihovskyi, A., Lehnertz, K., and Tetzlaff R. Detection of seizure precursors in the EEG with cellular neural networks. *Epilepsia*, vol. 45, suppl. 7 (abstract), (2004) 245.
- [9] Pritchard, W. S. and Duke, D. W. Measuring chaos in the brain: a tutorial review of nonlinear dynamical EEG analysis. *International Journal of Neuroscience* (preprint), Apr. 1992.
- [10] Szilágyi, L., Benyó, Z., and Szilágyi, S. M. A new method for epileptic waveform recognition using wavelets decomposition and artificial neural networks. *Proceedings of the Second Joint EMBS/BMES Conference*, (Houston, TX, Oct. 23-26, 2002), 2025-2026.
- [11] Sprott, J. C. *Chaos and Time-Series Analysis*, New York: Oxford University Press, 2003.
- [12] Takens, F. Detecting strange attractors in turbulence. *Dynamical Systems and Turbulence, Warwick 1980 Lecture Notes in Mathematics 898*. Berlin, Germany: Springer-Verlag, (1981), 336-381.
- [13] Tetzlaff, R., Niederhofer, C., and Fischer, P. Feature extraction in epilepsy using a cellular neural network based device – first results. *Proceedings of the International Symposium on Circuit and Systems 2003*, vol. 3, (May 25-28, 2003) III-850-III-853.