Accepted Manuscript

Data mining process for identification of non-spontaneous saccadic movements in clinical electrooculography

R.A. Becerra-García, R.V. García-Bermúdez, G. Joya-Caparrós,
A. Fernández-Higuera, C. Velázquez-Rodríguez,
M. Velázquez-Mariño, F.R. Cuevas-Beltrán, F. García-Lagos,
R. Rodráguez-Labrada

Please cite this article as: R.A. Becerra-García, R.V. García-Bermúdez, G. Joya-Caparrós, A. Fernández-Higuera, C. Velázquez-Rodríguez, M. Velázquez-Mariño, F.R. Cuevas-Beltrán, F. García-Lagos, R. Rodráguez-Labrada, Data mining process for identification of non-spontaneous saccadic movements in clinical electrooculography, *Neurocomputing* (2017), doi: 10.1016/j.neucom.2016.10.077
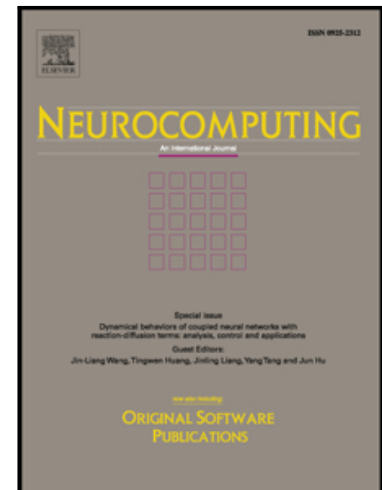
Data mining process for identification of non-spontaneous saccadic movements in clinical electrooculography

R. A. Becerra-García[1,**], R. V. García-Bermúdez[4,*], G. Joya-Caparrós[2,*], A. Fernández-Higuera[1], C. Velázquez-Rodríguez[1], M. Velázquez-Mariño[1], F. R. Cuevas-Beltrán[1], F. García-Lagos[2], R. Rodríguez-Labrada[3]

Abstract

In this paper we evaluate the use of the machine learning algorithms Support Vector Machines (SVM), K-Nearest Neighbors (KNN) and Classification and Regression Trees (CART) to identify non-spontaneous saccades in clinical electrooculography tests. We propose a modification to an adaptive threshold estimation algorithm for detecting signal impulses without the need for any manually pre-established parameters. Data mining tasks such as feature selection and model tuning were performed, obtaining very efficient models using only 3 attributes: amplitude deviation, absolute response latency and relative latency. The models were evaluated with signals recorded from subjects affected by Spinocerebellar Ataxia type 2 (SCA2). Results obtained by the algorithm show accuracies over 98%, recalls over 98% and precisions over 95% for the three models evaluated.

Keywords: saccade identification, clinical electrooculography, support vector machines, k-nearest neighbors, classification and regression trees

1. Introduction

The alteration of eye movements is a symptom of many neurological diseases such as Parkinsons syndrome, spinocerebellar ataxias or Niemann-Pick disease [1]. Specifically, in the Spinocerebellar Ataxia type 2 (SCA2) this alteration is an important clinical marker present in more than 90% of patients [2].

There are several kinds of eye movements such as saccades (rapid movement between fixation points), fixations and pursuits. Saccades are critical to evaluate subjects with SCA2. For instance, SCA2 patients have significantly slower saccades and with larger latencies than healthy subjects [2]. The analysis of this kind of movement is very often used in research conducted by the medical community, hence its importance.

Non-spontaneous saccades are those performed following a visual stimulus. These saccades are very useful in clinical studies because they allow control characteristics of the analyzed movements like amplitude. Spontaneous saccades are considered those that do not follow the visual stimuli, occurring at anytime with any amplitude. Studies

*Corresponding author

**Principal corresponding author

Email addresses: `idertator@facinf.uho.edu.cu` (R. A. Becerra-García), `rodgarberm@gmail.com` (R. V. García-Bermúdez), `gjoya@uma.es` (G. Joya-Caparrós), `afernandezh@facinf.uho.edu.cu` (A. Fernández-Higuera), `cvelazquezr@facinf.uho.edu.cu` (C. Velázquez-Rodríguez), `mvelazquez@facinf.uho.edu.cu` (M. Velázquez-Mariño), `fcuevas@facii.uho.edu.cu` (F. R. Cuevas-Beltrán), `fgl@uma.es` (F. García-Lagos), `roberto@ataxia.hlg.sld.cu` (R. Rodríguez-Labrada)

[1]Grupo de Procesamiento de Datos Biomédicos (GPDB), Universidad de Holguín, Cuba

[2]Departamento de Tecnología Electrónica (DTE), Universidad de Málaga, España

[3]Centro para la Investigación y Rehabilitación de las Ataxias Hereditarias (CIRAH), Holguín, Cuba

[4]Universidad Técnica de Manabí, Ecuador

conducted by medical researchers use the descriptive statistics of saccade features such as latencies and amplitudes [3]. The inclusion of spontaneous saccades introduces noise on these statistics. For this reason, spontaneous saccades are usually removed from this kind of analysis.

Video-oculography and electrooculography are the most common techniques used today to measure eye movement. Video-oculography uses infrared cameras to track the center of the pupil [4] with resolutions up to 0.3° [5]. Electrooculography consists in capturing the electrical potential of the eyes to calculate its magnitude and direction. This technique is widely used in electrophysiologic tests [6]. The resulting signals of this recording process are called electrooculograms [7].

Video-oculography signals recorded using high sample rate sensors have several advantages that outperform electrooculography in several scenarios. However, the data available for the authors were recorded using electrooculography, which is also adequate for saccadic analysis.

There are several methods and algorithms for identifying saccades in electrooculograms, the vast majority of them based on kinetic thresholds [8, 9, 10, 11], using supervised learning [12, 7], unsupervised learning [13] or other novel approaches [14, 15, 16] like particle filters [17]. Also, these algorithms can be classified into online [14, 7, 15, 16] or offline [17] methods.

The saccade identification algorithms are usually evaluated against data from healthy subjects where the differences between saccadic and non-saccadic movements are very evident. However, in electrooculography clinical tests these methods try to detect as many saccades as possible, without distinguishing which of them are spontaneous and which are not.

In a previous study [18], we proposed a method that identifies saccadic movements using a sliding window approach in the angular movement signal. This method allows us to discriminate whether a sample belongs to a saccadic movement or not. Sometimes only a subset of these detected saccades have real value for clinical analysis, mainly due to their relation with the visual stimuli. Saccades unrelated to the stimuli are denominated

spontaneous saccades and may not be useful to researchers in some scenarios.

The goal of this work is to identify non-spontaneous saccadic eye movements for their use in clinical analysis using a feature-based approach. To accomplish the goal a 3-step process was implemented: signal preprocessing, impulse detection and impulse classification. In the preprocessing step, the signals are filtered and differentiated to remove noise and to obtain the velocity profile. To detect the impulses a modification to the threshold estimation algorithm introduced by [14] was employed. Finally, the impulses are binary classified into non-spontaneous saccadic movement or not using a machine learning model.

For the last step, three machine learning algorithms were considered based on the results obtained in [19]: Support Vector Machines (SVM) [20], K-Nearest Neighbors (KNN) [21] and Classification and Regression Trees (CART) [22]. To improve their performance two data mining tasks were implemented: feature selection and model tuning. Feature selection allow us to reduce the number of features needed to feed the models to 3. Also, the tuning task finds an optimal set of parameter values for the trained models. The performance of the tuned models was measured and obtained over 98% accuracy and recall, and over 95% precision in all of them.

The work presented here is an extension of [19]. Here the focus is to improve the performance of the learning algorithms using data mining techniques like feature selection and model tuning. The Naive Bayes classifier was removed from the current analysis due the inferior performance showed in [19].

The rest of this paper is organized as follows: In Section 2 we describe the designed experiments, the available data and the employed methods. Section 3 is devoted to analyze and comment on the results. Finally, Section 4 summarizes the main conclusions and future lines of work.

2

2. Material and Methods

To test the selected algorithms an experiment was designed. The first step was to detect potential impulses and annotate them to build a labeled dataset. Then, the best features were selected from among the ten considered. Once we have the best set of features, we proceed to tune the parameters of the selected models to find the fittest ones for the task at hand, using 10-fold cross-validation. Finally, we analyze and compare the performance of the models against new examples using metric accuracy, recall and precision.

Clinical tests of electrooculography are setup as follows. Subjects, with their heads fixed, are seated in front of a monitor at a set distance. Then, they are asked to follow two static points which appear and disappear alternately on each side of the monitor. The distance between the two points is used to control the angle of the stimuli from the subject's position. Capturing eye movements under these conditions, using electrooculography, allows researchers to identify which saccades correspond to the stimulus and which ones are spontaneous and also to calculate important features of these movements like latency, duration, amplitude, deviation and maximal velocity.

The electrooculograms were recorded using the OtoScreen electronystamography device at a sampling rate of 200 Hz with a bandwidth of 0.02 to 70 Hz. Records of 12 patients with SCA2 were used to build a dataset with features extracted from signal impulses. Each one of the records contains a minimum of three tests including 10°, 20° and 30° of visual stimulation. Typical saccadic tests have at least one horizontal channel and one stimulus signal (Figure 1).

Jupyter notebooks [23] were used in conjunction with the Python language scientific facilities: NumPy [24], SciPy [25], Pandas [26], Matplotlib [27] and Scikit-Learn [28] for running the experiments. The intention behind using Python powered technologies is that the resulting algorithm (including trained models) will be used at NSEog, a processing platform developed by the authors.
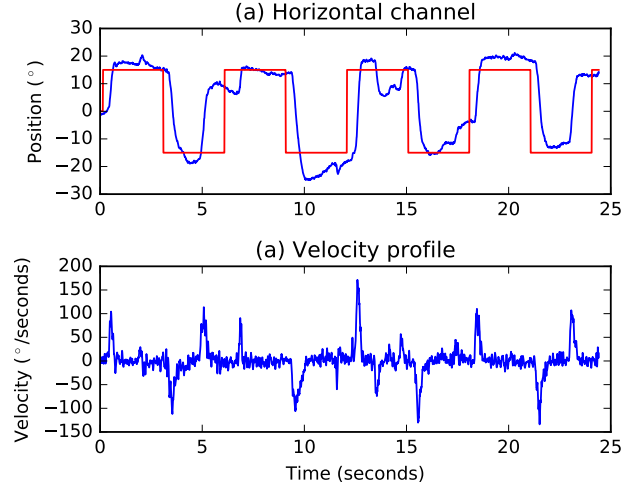

Figure 1: Typical electrooculography signal with 30° stimulus angle of a subject suffering SCA2. Red signals are the scaled stimuli signals. Blue signals are the horizontal channel (a) and its velocity profile (b) respectively.

2.1. Signals preprocessing

Before the identification of potentially saccadic impulses, two common tasks need to be performed: denoising and differentiation. Noise removal is a very important matter in order to eliminate undesired spectral components produced by equipment malfunction, poor analog filtering or biological artifacts. Differentiation allows the velocity profile used later by the algorithm to be obtained.

Median filter (Equation 1) has proven to be very robust in eliminating high frequency signal noise while preserving sharp edges. A study carried out in [29] demonstrated that this kind of filter is appropriate for eye movements signals. To eliminate undesired noise present in the signals used in the experiment, we use a median filter with a window size of 9 samples (approximately 45 milliseconds). This is accomplished using the **medfilt** function of SciPy.

$$y_i = median\{x_j | j = i - k, \ldots, i + k\} \qquad (1)$$

Due to the discrete nature of these signals, numerical differentiation is employed to calculate the velocity profiles. In accordance with previous experiments conducted by the authors, Lanczos differentiators (Equation 2) with 11 points have

good performance for signals with the same characteristics as the ones used in this experiment.

$$f'(x^*) \approx \frac{3}{h} \sum_{k=1}^{m} k \frac{f_k - f_{-k}}{m(m+1)(2m+1)} \quad (2)$$

where $m = \frac{N-1}{2}$ and h is the step (time between samples). The formula denoted in Equation 2 with $N = 11$ is used to calculate the velocity profiles.

2.2. Detection of impulses

Saccadic movements are represented as impulses in a velocity graph (Figure 1b). Typically, these movements can be easily detected by their contrast in magnitude or shape with other movements such as microsaccades and fixations respectively. However, for the same stimulus angle the range of values of true saccadic impulses vary from subject to subject. This situation depends greatly on the degree of illness present in the subject [30].

One of the critical steps of the algorithm is the detection of velocity impulses which can potentially be saccades. For that matter, a threshold is needed to know when the velocity reaches a certain value that can be considered as a saccade candidate. Due to the inter-subject variability explained before, this threshold should not be fixed a priori. Also it should be large enough to ignore in most cases other movements like microsaccades and fixations, and not too large to miss valid saccadic movements.

To detect impulses, we developed the algorithm described in Algorithm 1, which is a modification to the method introduced by Nyström and Holmqvist in [14]. The algorithm uses the absolute values of the velocities samples (V) to calculate the approximation of the initial threshold (last). This initial threshold is calculated by adding λ times the standard deviation of the velocities to its mean. Then, it adjusts the threshold iteratively, with the same formula using only selected samples (S) of velocities below the previous threshold. The stop condition happens when the difference between the current threshold and the last one is less or equal to one degree per second. The value of the resulting threshold is represented graphically by the red line in Figure 2.
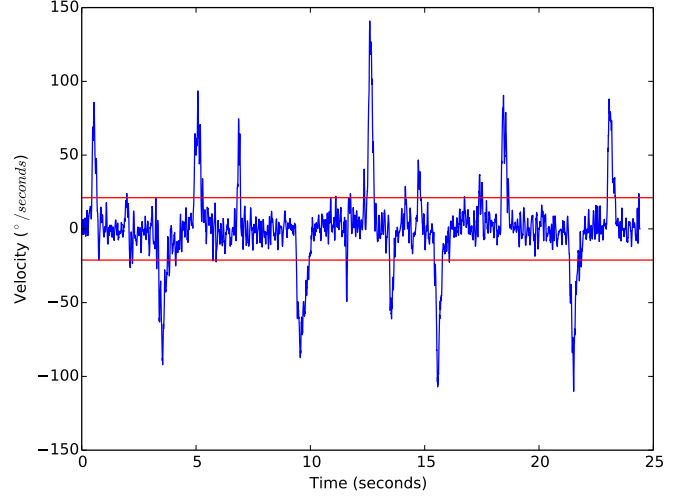

Figure 2: Threshold estimated in a 30° stimulus angle test of a subject with SCA2.

Algorithm 1: Modified version of Nyström and Holmqvist [14] threshold estimation algorithm

Input	: velocity profile (Array of degree/seconds samples)
Input	: λ (Safety margin)
Output	: Threshold estimation

begin

    V ⟵ Abs(velocity profile);

    last ⟵ Mean(V) + λ * Std(V);

    current ⟵ 0;

    while Abs(last- current) > 1 **do**

        S ⟵ samples from V below last threshold;

        current ⟵ last;

        last ⟵ = Mean(S) + λ * Std(S);

    return last;

4

The original algorithm requires the initial threshold as input. This adds subjectivity to the main process, because to obtain good detection results this value must be variable and set by the user. The noise levels present on the signals and the degree of affectation of the subject have great influence on this issue. The proposed modification consists in calculating the initial threshold in an adaptive way using all velocity samples, thereby eliminating the subjectivity of the original approach. Using the new approach, signals recorded from subjects with SCA2 in different stages seems to be adequate to the task at hand.

The safety margin ($\lambda = 6$) employed by [14] ignores too many valid saccadic movements in lower angle tests for subjects with SCA2. A value of $\lambda = 3$ seems to be adequate for most cases at the expense of the detection of more non valid impulses. Despite a penalty in runtime performance, the final accuracy of the method should not decrease significantly. Due to the amplitude of these new impulses the classification model should avoid them.

Finally, we detect the impulses individually by finding a group of samples grouped together that exceeds the calculated threshold. The principle behind this algorithm is looping through the signal to find velocities above the threshold. When we encounter one of these points, we move to the left and to the right until the velocity is zero or drops below zero. This approach allows further refinement of the starting and ending points of the saccade because the impulses usually obtain more samples from beyond the real saccade limits. If the length of a detected impulse is not greater than 10 samples, then it is discarded to avoid very small invalid movements. A typical output of this method is represented in Figure 3.

2.3. Data mining process

Because Python technologies are being used, Scikit-Learn was selected as the machine learning library. The main policy of our algorithm selection was to choose methods with different working principles. We therefore evaluated three different classification algorithms: Support Vector Machines, K-Nearest Neighbors, and Classification
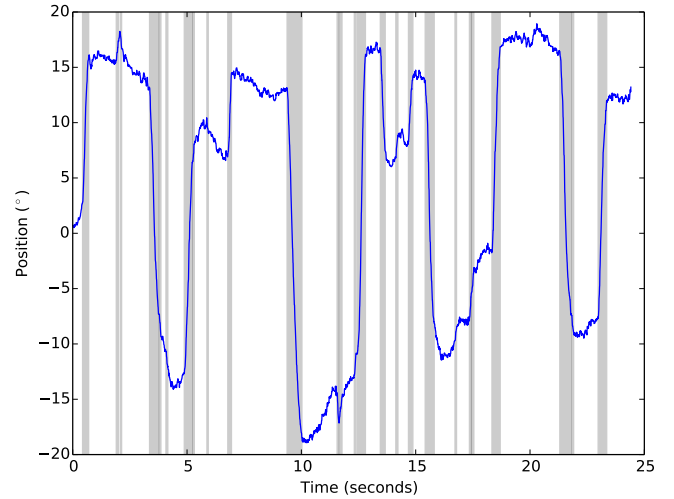
Figure 3: Identified impulses in the same signal used in Figure 2

and Regression Trees. Beside these algorithms, in the previous work [19] extended here, we included Naive Bayes classifier, but due to its poor performance we chose to remove it from this work.

2.3.1. Feature selection

Once we have the saccadic impulse candidates, we need to know if they are saccades and if they are related to the stimulus. Therefore, the strategy behind our approach uses features used by human intuition to solve this task. To take advantage of the characteristics of the clinical tests, a set of 9 features was preselected. These features are of spatial, temporal or kinetic nature. The final and definitive feature set will be obtained from among the most relevant of these ones, using feature selection techniques.

We have selected four spatial features (Figure 4b): **angle** ($10°, 20°, 30°$), **amplitude** ($°$), **deviation** ($0..2$) and **end normalized position** ($0..1$). **Angle** represents the amplitude of the stimulus and can take only three values. This feature was selected to specify the small differences between the saccadic movements resulting from using different angles of stimulation. **Amplitude** feature is the difference (in degrees) between the end position value and the start position value of the impulse. Used commonly by the medical community, the **deviation** is the ratio **amplitude** over
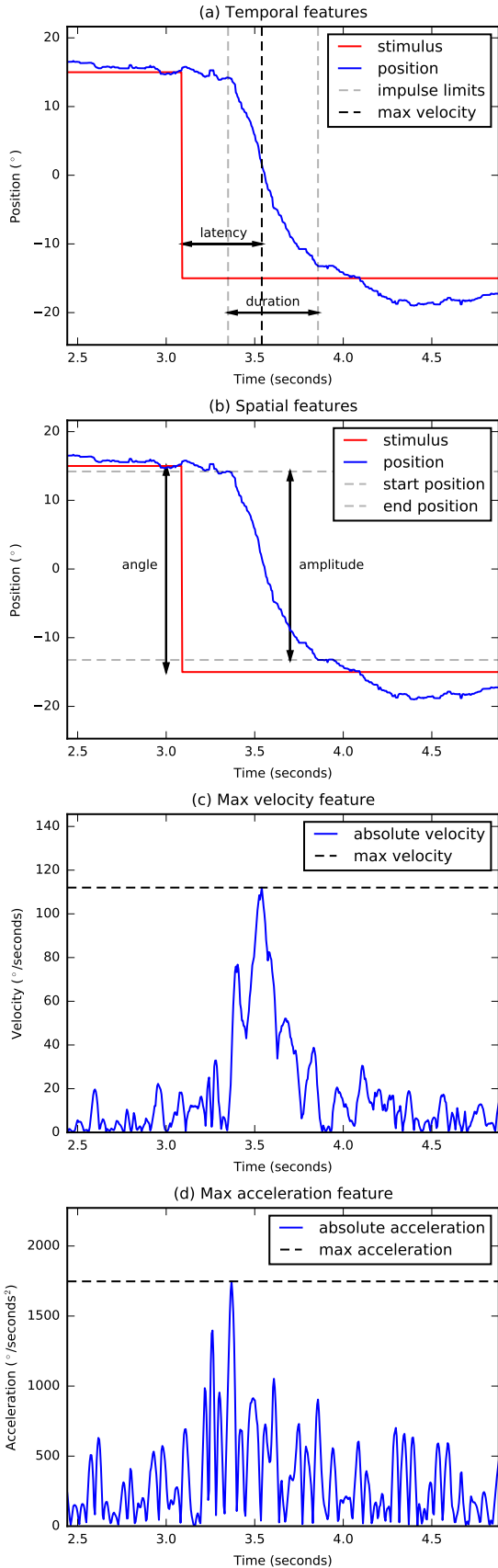
5

Figure 4: Graphical representation of selected features

angle. The **end normalized position** takes values between 0 (left side) and 1 (right side), representing on which side of the stimulus the impulse ends.

It is important to note, we have removed the **direction** feature employed in [19]. It is evident that movements against the stimulus direction are not valid, so we choose to set these kind of impulses as not valid. This variation leads to a simpler and better model, adding only one line of code in the identification algorithm.

We have **absolute latency** (ms) and **normalized latency** (0..1) as temporal features (Figure 4a). **Absolute latency** is the time between the start of the stimulus transition and the maximal velocity point of the impulse in milliseconds. **Normalized latency** is a version of absolute latency with values between 0 and 1. The value 0 means that the maximal velocity occurs at the moment of stimulus transition, and the value 1 means that maximal velocity occurs at a moment before the next stimulus transition.

Finally, we selected three kinetic features: **maximum velocity** ($°/s$, Figure 4c), **maximum acceleration** ($°/s^2$, Figure 4d) and **maximum jerk** ($°/s^3$). These features were calculated using the first, second and third derivatives respectively of the horizontal channel signal. The method employed to calculate the numerical derivatives was the one specified in Equation 2.

Using the features previously selected, a dataset of signal impulses was created. To build this dataset, a human specialist, aided by the NSEog, classified the detected impulses into valid and non-valid saccades. As a result, 1782 valid saccades and 3301 not valid impulses were obtained, resulting in 5083 instances as shown in the training row of Table 1. There was a notable decrease from the number of impulses (8606) used in [19], due to the removal of impulses against the direction of the stimulus.

Having the full dataset, the first step of our data mining process was to select how many and which of the features were the most relevant to our particular problem. To accomplish this task, we used the Recursive Feature Elimination method (RFE) implemented in Scikit-Learn library based

6

Table 1: Data distribution for training and validation stages

Stage	Valid	Nonvalid	Total
Training	1782	3301	5083
%	35.06	64.94	100.00
Validation	365	1461	1826
%	19.99	80.01	100.00
Total	2147	4762	6909
%	31.08	68.92	100.00

on [31]. This is a filter method that uses the weights assigned by a linear model trained with a specific subset of features and removes those with the smallest absolute weights from the subset. This procedure was repeated until the number of desired features was reached, similar to the one presented in [32].

We applied RFE with a Logistic Regression classifier to obtain the subsets with the best features with sizes from 1 to 9. Then, we trained the three selected models against the data from each obtained subset and evaluated them using the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve. We used this metric because we want to maximize the *true positive rate* (aka precision) while minimizing the *false positive rate*.

Figure 5 shows that all classifiers drop their performance, starting with a subset of four features. So, the best performance is achieved with 3 features, specifically with **latency**, **deviation** and **normalized latency**.

2.3.2. Model tuning

All the algorithms employed in this work use parameters that affect the performance of the classification task. The Scikit-Learn library provides default values for all parameters in the algorithms. These default values were selected to obtain acceptable results across a broad set of problems. So, it is very likely that we can improve the performance of these algorithms by selecting the right values of their parameters for our specific problem.

Before the search process, the dataset parameters were standardized by removing the mean and scaled to unit variance. This technique is critical in obtaining good results in the training of
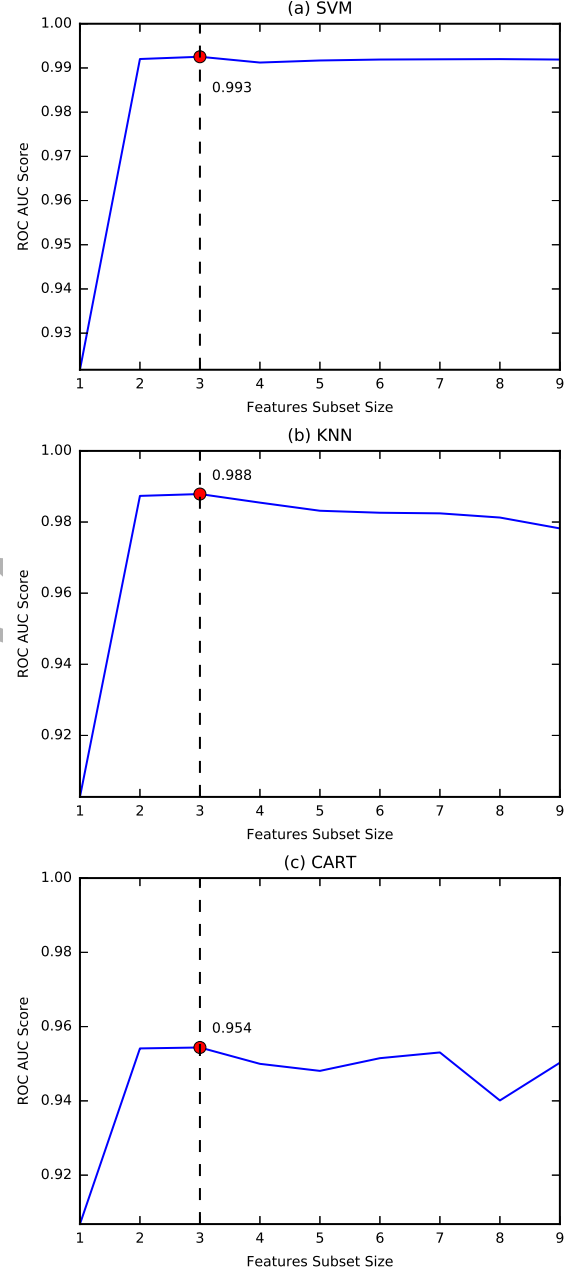

Figure 5: ROC AUC score of the three selected classifiers for each features subsets size. The red dots indicate the best score achieved for each classifier.

the Radial Basis Function (RBF) kernel version of Support Vector Machines. These scales were saved along with the model for further use by the algorithm.

The process of selecting the right values of these parameters is commonly known in the field as *model tuning*. The first step of this process is to determine which of the parameters needs tuning, thus a basic understanding of them is required. Then, we try different combinations of the parameter values to obtain the one that yields the best performance. This last task is done using a technique known as Grid Search [33], which basically evaluates all possible parameter combinations of a given set of values. This technique is very often used to tune the parameters of SVMs [34, 33]. However, they can also be used to tune any kind of models.

Support Vector Machines (SVMs) are a set of supervised learning methods which are very effective in high dimensional spaces [20]. The most important parameter of these algorithms is C [35], which is basically the tolerance of samples outside the boundaries of the decision hyperplane and regulates how close the obtained model is to the training data [36]. There are also very versatile classifiers, supporting several kernel functions. Scikit-Learn implements four of these kernels: linear $\langle x, x' \rangle$, polynomial $(\gamma \langle x, x' \rangle + r)^d$, RBF[5] $e^{-\gamma |x - x'|^2}$ and sigmoid $tanh(\gamma \langle x, x' \rangle + r)$. As shown in the kernel expressions, γ is a tuning parameter and d is the degree of the polynomial in this specific kernel. However, the polynomial kernel uses a great amount of runtime resources that the authors did not have, so we chose not to include them in this experiment.

K-Nearest Neighbors is a type of instance-based learning that can be used both for supervised or unsupervised learning. Instead of creating a generalizing function, it stores all the data inside the model using different data structures like Ball Trees or KD Trees. The principle behind the algorithm is to find a number of training samples nearest to that analyzed and predict the label

---

Table 2: Parameter values evaluated using GridSearch

Algorithm	Parameter	Values
SVM	C	$10^{-2}, 10^{-1}, \ldots, 10^5, 10^6$
	kernel	linear, sigmoid, rbf
	γ	$10^{-5}, 10^{-4}, \ldots, 10^2, 10^3$
KNN	number of neighbors (nn)	$1 \ldots 100$
CART	split criteria (sc)	gini, entropy
	max depth (md)	$1 \ldots 10$

from it [21]. So, in this case the parameter to tune is the *number of neighbors* to evaluate. The data structure used is determined automatically by the Scikit-Learn implementation according to the properties of training data.

Decision trees are nonparametric, supervised learning techniques. This algorithm requires little preprocessing and its runtime performance is good enough to handle real time tasks. This method splits the data trying to infer decision rules which can be used to classify instances. Scikit-Learn uses an optimized version of the Classification and Regression Tree (CART) [22]. There are several parameters that can be tuned. Although parameters such as *min samples split*, *min samples leaf*, *min weight fraction leaf* and *max leaf nodes* are really influential in deep complex trees, that was not our case. So we determine in our case that only the *split criteria* and the *max depth* parameters need tuning. Another important feature of these kinds of algorithms is that if the resulting tree is simple enough, it can be easily interpreted by humans and represented in a basic set of rules.

Table 2 shows the parameter values used in the tuning search process. In the case of C and γ we use a logarithmic range like the one used in [33].

The algorithms were trained with the same dataset used for feature selection, with 5083 impulses (see Table 1 for the distribution). As validation scheme we use a stratified 10-fold cross validation to internally evaluate the algorithms. The Area Under the Curve of the Receiver Operating Characteristic (ROC AUC) was employed as evaluation metric. As mentioned, this metric allows us to maximize true positives while minimizing false positives, reducing the impact of the classification error in clinical applications.

The tuning stage was executed in the Jupyter

---

[5]Used by default in Scikit-Learn and the most popular one

8

Table 3: Parameter tuning results

Algorithm	Parameter	Value	Score	Size	Time
SVM	C	10000	0.9941	17 Kb	168.7 min
	kernel	rbf			
	γ	0.01			
KNN	nn	79	0.9926	348 Kb	4.9 s
CART	sc	entropy	0.9892	2 Kb	0.8 s
	md	3			

Table 4: Comparison of validation results against [19]

	ROC AUC	Accuracy	Precision	Recall
SVM	0.991410	0.987952	0.954907	0.986301
SVM [19]	-	0.980248	0.910039	0.991477
CART	0.992080	0.988499	0.959893	0.983562
CART [19]	-	0.970766	0.904502	0.941761
KNN	0.992797	0.987404	0.950000	0.989041
KNN [19]	-	0.979457	0.909686	0.987216

Notebook environment in a MacBook Pro of Early 2015 with an Intel Core i5 at 2.7 GHz and 8 Gb of RAM. Table 3 shows the results obtained by the Grid Search procedure. The **Score** column refers to the ROC AUC performance metric value, **Size** is the amount of storage used by the model and **Time** is how long it takes the procedure to finish. As we can see the performance score obtained is very high and very similar in all three cases. In the next section we analyze this performance with data not used for training the algorithms.

3. Results

To carry out a performance analysis of the obtained models, a new dataset was prepared with 1826 impulses, 365 valid saccades and 1461 nonvalid ones as shown in the validation row of Table 1. Note that the ratio between positive and negative samples is very different from the one used in the training stage. However, this situation has no influence on the resulting performance as shown in the first column of Table 4.

In previous works the metrics employed to validate the algorithms were *accuracy*, *precision* and *recall* [19]. For reasons of comparison, Table 4 shows the performance of the tuned models using these metrics in addition to ROC AUC. These results are slightly better than those achieved in [19] for all cases.

To analyze the influence of the classifier predictions in the clinical parameters used by med-

ical researchers a boxplot is presented in Figure 6. Only the parameter values of manual or predicted, valid saccadic movements were included in each subplot. For the parameters **amplitude** and **max velocity**, the values are almost the same for those predicted by the algorithms and the ones manually labeled by the human expert.

In the case of **absolute latency**, the CART and SVC (Support Vector Classifier) ignores some outlier values which are probably due to human errors in the annotation process. In contrast, KNN prediction adds some outlier values that were marked as false positives. Also, for the **deviation** parameter, KNN adds one outlier value marked as a false positive.

As shown in Table 4, all the classifiers have very similar performance. However, the false positives predicted by the KNN classifier can introduce errors to clinical analysis which are generally based on descriptive statistics like *mean* or *standard deviation*. The SVC have slightly less performance than the CART tree and have a high cost on training runtime.

In contrast with KNN and SVC, the output of the CART algorithm is easily interpretable by humans. In this experiment, the optimal depth is 3, so the resulting model is efficient and simple. It is also noteworthy that the parameter tuning has a significant impact on this specific algorithm, in which the gain of ROC AUC performance is about 4 percentile points (see the difference between the max value in Figure 5c and the value of Table 4). For all these reasons the recommended algorithm is CART over SVC and KNN.

As explained in [19], literature about the task proposed in this work is scarce and no methods to specifically solve it were found. This work outperforms that presented by [19] in most cases using less input parameters. Also, the data mining process yield an improved CART model, which seems to be the most suitable for the proposed task.

4. Conclusions

This work describes a procedure to identify spontaneous saccades from a set of detected impulses in electrooculography signals. To detect
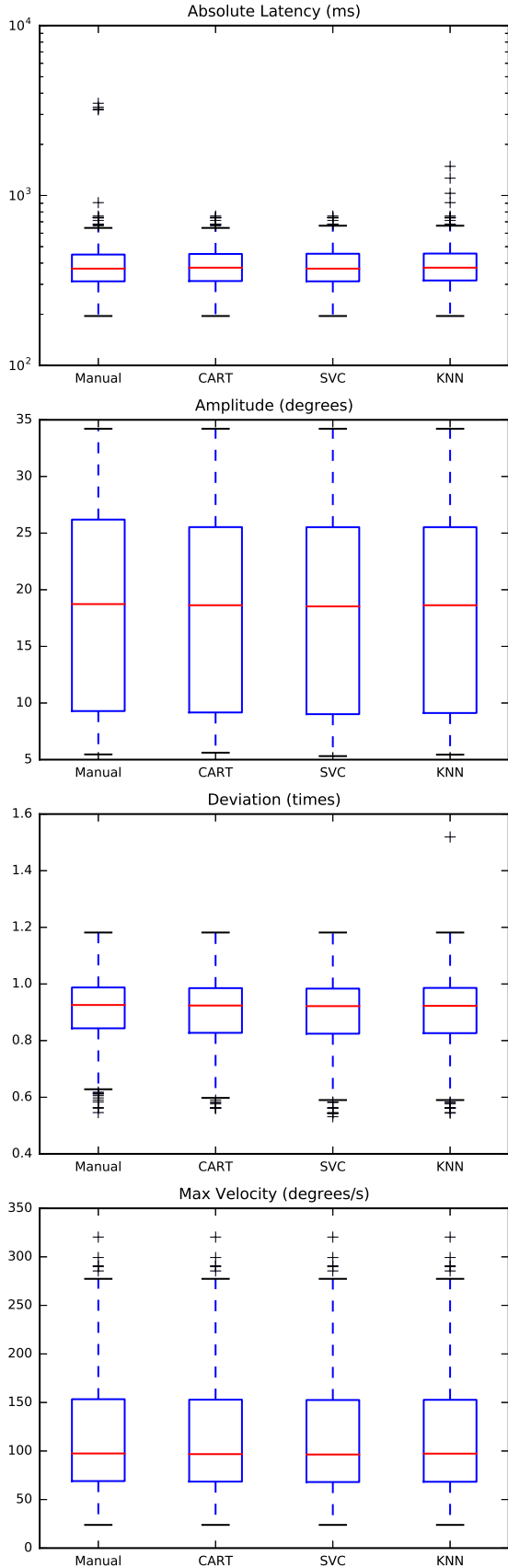
Figure 6: Comparison of properties of manual labeled saccades against predicted ones.

the impulses a modification to the algorithm proposed in [14] was presented, which consists in adaptively calculating the initial thresholds. This new algorithm avoids the need for thresholds or any other user input and works very well for noisy signals like the ones recorded from subjects with SCA2, which is a difficult task.

For the classification task three machine learning paradigms are compared: Support Vector Machines, K-Nearest Neighbors, and Classification and Regression Trees. The procedure has been applied to a database of eye movements recorded from subjects suffering spinocerebellar ataxias.

To improve classification performance, some data mining tasks such as feature selection and model tuning were performed. The feature selection yields that the optimal subset of features includes three parameters: deviation, absolute latency and normalized latency. The use of the Grid Search technique in the model tuning stage, identifies a good configuration of parameter values for the classification algorithms. Specifically, for the CART algorithm, the selected configuration improved its performance significantly.

The evaluation of the performance was carried out using metrics such as ROC AUC, Accuracy, Recall and Precision. The three paradigms achieved a ROC AUC above 99%, an accuracy above 98%, a recall above 98% and a precision above or equal to 95% by external validation (using patterns not used for training). The analysis of saccadic attributes with clinical importance, the simplicity of the resulting model and the performance measured, result in the recommendation of the CART algorithm for the proposed task.

The resulting classification algorithm is very simple and flexible. It consists in the evaluation of the features calculated from impulses detected in the signal by the supervised model. This approach allows the parallelization of the algorithm and even the swapping of the model if necessary. Due the use of the proposed impulse detection algorithm, the need for parameter managed by the user is eliminated.

10

References

[1] C. Bonnet, J. Hanuška, J. Rusz, S. Rivaud-Péchoux, T. Sieger, V. Majerová, T. Serranová, B. Gaymard, E. Růžička, Horizontal and vertical eye movement metrics: What is important?, Clinical Neurophysiology 124 (11) (2013) 2216–2229. doi:10.1016/j.clinph.2013.05.002.

[2] L. Velázquez-Pérez, R. Rodríguez-Labrada, J. C. García-Rodríguez, L. E. Almaguer-Mederos, T. Cruz-Mariño, J. M. Laffita-Mesa, A Comprehensive Review of Spinocerebellar Ataxia Type 2 in Cuba, The Cerebellum 10 (2) (2011) 184–198. doi:10.1007/s12311-011-0265-2.

[3] L. Velázquez Pérez, G. Sánchez Cruz, N. Canales Ochoa, R. Rodríguez Labrada, J. Rodríguez Díaz, L. Almaguer Mederos, J. Laffita Mesa, Electrophysiological features in patients and presymptomatic relatives with spinocerebellar ataxia type 2, Journal of the Neurological Sciences 263 (1-2) (2007) 158–164. doi:10.1016/j.jns.2007.07.013.

[4] J. N. van der Geest, M. A. Frens, Recording eye movements with video-oculography and scleral search coils: a direct comparison of two methods, Journal of Neuroscience Methods 114 (2) (2002) 185–195. doi:10.1016/S0165-0270(01)00527-1.

[5] P. Blignaut, K. Holmqvist, M. Nyström, R. Dewhurst, Improving the accuracy of video-based eye-tracking in real-time through post-calibration regression, in: EyeTrack Australia Conference, 2012. doi:10.1007/978-3-319-02868-2.

[6] M. F. Marmor, E. Zrenner, Standard for clinical electro-oculography, Documenta Ophthalmologica 85 (2) (1993) 115–124. doi:10.1007/BF01371127.

[7] A. Bulling, J. Ward, H. Gellersen, G. Troster, Eye Movement Analysis for Activity Recognition Using Electrooculography, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (4) (2011) 741–753. doi:10.1109/TPAMI.2010.86.

[8] P. Inchingolo, M. Spanio, On the Identification and Analysis of Saccadic Eye Movements-A Quantitative Study of the Processing Procedures, IEEE Transactions on Biomedical Engineering 32 (9) (1985) 683–695. doi:10.1109/TBME.1985.325586.

[9] M. Juhola, V. Jäntti, I. Pyykkö, M. Magnusson, L. Schalén, M. Åkesson, Detection of saccadic eye movements using a non-recursive adaptive digital filter, Computer Methods and Programs in Biomedicine 21 (2) (1985) 81–88. doi:10.1016/0169-2607(85)90066-5.

[10] H. J. Wyatt, Detecting saccades with jerk, Vision Research 38 (14) (1998) 2147–2153. doi:10.1016/S0042-6989(97)00410-0.

[11] D. D. Salvucci, J. H. Goldberg, Identifying fixations and saccades in eye-tracking protocols, ETRA '00, ACM, New York, USA, 2000, pp. 71–78. doi:10.1145/355017.355028.

[12] P. Tigges, N. Kathmann, R. R. Engel, Identification of input variables for feature based artificial neural networks-saccade detection in EOG recordings, International Journal of Medical Informatics 45 (3) (1997) 175–184. doi:10.1016/S1386-5056(97)00042-7.

[13] J. Otero-Millan, J. L. A. Castro, S. L. Macknik, S. Martinez-Conde, Unsupervised clustering method to detect microsaccades, Journal of Vision 14 (2) (2014) 18.

[14] M. Nyström, K. Holmqvist, An adaptive algorithm for fixation, saccade, and glissade detection in eye-tracking data, Behavior Research Methods 42 (1) (2010) 188–204. doi:10.3758/BRM.42.1.188.

[15] K. Pettersson, S. Jagadeesan, K. Lukander, A. Henelius, E. Hæggström, K. Müller, Algorithm for automatic analysis of electro-oculographic data, Biomedical Engineering Online 12 (1) (2013) 110. doi:10.1186/1475-925X-12-110.

[16] M. Toivanen, K. Pettersson, K. Lukander, A probabilistic real-time algorithm for detecting blinks, saccades, and fixations from EOG data, Journal of Eye Movement Research 8 (2) (2015) 1–14.

[17] P. M. Daye, L. M. Optican, Saccade detection using a particle filter, Journal of Neuroscience Methods 235 (2014) 157–168.

[18] R. A. Becerra-García, G. Joya, R. V. G. Bermúdez, L. Velázquez, R. Rodríguez, C. Pino, Saccadic Points Classification Using Multilayer Perceptron and Random Forest Classifiers in EOG Recordings of Patients with Ataxia SCA2, in: I. Rojas, G. Joya, J. Cabestany (Eds.), Advances in Computational Intelligence, Springer Berlin Heidelberg, 2013, pp. 115–123.

[19] R. A. Becerra-García, R. García-Bermúdez, G. Joya-Caparrós, A. Fernández-Higuera, C. Velázquez-Rodríguez, M. Velázquez-Mariño, F. Cuevas-Beltrán, F. García-Lagos, R. Rodríguez-Labrada, Non Spontaneous Saccadic Movements Identification in Clini-

11

cal Electrooculography Using Machine Learning, in: I. Rojas, G. Joya, A. Catala (Eds.), Advances in Computational Intelligence, Vol. 9095 of Lecture Notes in Computer Science, Springer International Publishing, 2015, pp. 56–68. doi:10.1007/978-3-319-19222-2_5.

[20] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297. doi:10.1007/BF00994018.

[21] B. W. Silverman, M. C. Jones, An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation, International Statistical Review 57 (3) (1989) 233–238. doi:10.2307/1403796.

[22] L. Breiman, J. Friedman, C. J. Stone, R. A. Olshen, Classification and regression trees, CRC press, 1984.

[23] F. Pérez, B. E. Granger, IPython: A System for Interactive Scientific Computing, Computing in Science & Engineering 9 (3) (2007) 21–29. doi:10.1109/MCSE.2007.53.

[24] T. E. Oliphant, Python for scientific computing, Computing in Science & Engineering 9 (3) (2007) 10–20.

[25] E. Jones, T. Oliphant, P. Peterson, SciPy: Open source scientific tools for Python, http://www.scipy.org/.

[26] W. McKinney, Data Structures for Statistical Computing in Python, 2010, pp. 51–56.

[27] J. D. Hunter, Matplotlib: A 2D Graphics Environment, Computing in Science and Engineering 9 (3) (2007) 90–95.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, Édouard Duchesnay, Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[29] M. Juhola, Median filtering is appropriate to signals of saccadic eye movements, Computers in Biology and Medicine 21 (1–2) (1991) 43–49. doi:10.1016/0010-4825(91)90034-7.

[30] R. Rodríguez-Labrada, L. Velázquez-Pérez, Eye movement abnormalities in spinocerebellar ataxias, Spinocerebellar ataxias. Rijeka: Intech (2012) 59–76.

[31] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Machine Learning 1–3 (46) (2002) 389–422.

[32] Q. Liu, Z. Zhao, Y.-X. Li, Y. Li, Feature selection based on sensitivity analysis of fuzzy isodata, Neurocomputing 85 (2012) 29–37.

[33] C. wei Hsu, C. chung Chang, C. jen Lin, A Practical Guide to Support Vector Classification.

[34] D. Meyer, F. Leisch, K. Hornik, The support vector machine under test, Neurocomputing 55 (1) (2003) 169–186.

[35] K. Duan, S. S. Keerthi, A. N. Poo, Evaluation of simple performance measures for tuning svm hyper-parameters, Neurocomputing 51 (2003) 41–59.

[36] I. H. Witten, E. Frank, M. A. Hall, Data Mining. Practical Machine Learning Tools and Techniques, third edition Edition, Morgan Kaufmann, 2011.

Roberto Becerra García is a PhD student in engineering from the University of Malaga, Spain. Master in Applied Mathematics and Computer Science of Administration by the University of Holguin (Cuba). His research interests are artificial intelligence, programming techniques and signal processing. Develops his scientific work in biomedical signal processing, with emphasis on the application of computational techniques to eye movements records and neurological system. Is a member of the Cuban Association for Pattern Recognition and the Cuban Society of Mathematics and Computer Science.

Rodolfo García Bermúdez received the B.Sc. degree in Electronic Engineering from the Universidad Central de Las Villas (Cuba, 1985), and the M.S. degree in Mathematics and Computer Science for Management from the Universidad de Holguín "Oscar Lucero Moya" (Cuba, 2005); and the Ph.D. degree in Computer Science from the Universidad de Granada (Spain, 2010), focused in independent component analysis applied to ocular movements. Currently, he is pro-

fessor in the University of Manabí, Ecuador and his research interests are related to adaptive processing of biomedical signals.

Gonzalo Joya was born in Salobreña (Spain), in 1960. He received the B.S. degree in Physics from the University of Granada (Spain) and the Ph.D. degree from the University of Málaga (Spain). He is currently an Associate Professor (Titular) at the Department of Electronics Technology in the University of Málaga and a Member of the Programme Committee of several conferences, such as the International Work-Conference on Artificial Neural Networks (IWANN) and the European Symposium on Artificial Neural Networks (ESANN). His research interests include application of neural networks to optimization problems and control of autonomous systems.

Abel Fernández Higuera is a Computer Engineer graduated from the University of Holguin (Cuba). Teaches on topics related to web programming. Develops its scientific work in digital biomedical signal processing, with emphasis on electrooculografics records and Independent Component Analysis. Is a leader member of the develop of the project NSEog. Is a member of the Cuban Association for Pattern Recognition and

13

the Cuban Society of Mathematics and Computer Science.

Camilo Velázquez Rodríguez is a Computer Engineer graduated from the University of Holguin (Cuba). Teaches on topics related to artificial intelligence and signal processing. Develops its scientific work in the digital processing and mathematical modeling of biomedical signals, with emphasis on electrooculografics records. He is a member of the Cuban Association for Pattern Recognition and the Cuban Society of Mathematics and Computer Science.

Michel Velázquez Mariño is a PhD student in engineering from the University of Malaga, Spain. Master in Applied Mathematics and Computer Science of Administration by the University of Holguin (Cuba). His research interests are programming techniques, database systems and signal processing. Develops his scientific work in biomedical signal processing with application in disorders of movement coordination and cognitive disorders. Is a member of the Cuban Association for Pattern Recognition and the Cuban Society of Mathematics and Computer Science.

Franger Cuevas Beltrán is an Industrial Engineer graduated from the University of Holguin (Cuba). Professor in the Department of Industrial Engineering at the University of Holguin. Is a member of the discipline Logistics and Process Management. Collaborates in the development of NSWeb, computer system for the management of biomedical data.
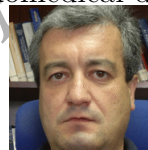
Francisco García Lagos graduated as a Computer Engineer and then received the PhD degree from the University of Málaga in 2003. He is currently an Associate Professor at the Department of Electronics Technology in the University of Málaga, Spain. He has published several journal and conference papers about his research interests, which include neural networks, genetic algorithms and fuzzy logic

Roberto Rodríguez Labrada received the B.Sc. degree in Microbiology from the Universidad de la Habana, and a Ph.D. degree in Health Sciences focused in ocular movements in ataxia patients. Currently, he is a researcher at the Center for the Research and Rehabilitation of Hereditary Ataxias, Holguín, Cuba.

14