Non Spontaneous Saccadic Movements Identification in Clinical Electrooculography Using Machine Learning

Roberto Becerra-García[1(✉)], Rodolfo García-Bermúdez[1,3], Gonzalo Joya-Caparrós[2], Abel Fernández-Higuera[1], Camilo Velázquez-Rodríguez[1], Michel Velázquez-Mariño[1], Franger Cuevas-Beltrán[1], Francisco García-Lagos[2], and Roberto Rodríguez-Labrada[4]

[1] Universidad de Holguín, Grupo de Procesamiento de Datos Biomédicos (GPDB), Holguín, Cuba
{idertator,afernandezh,cvelazquezr,mvelazquez}@facinf.uho.edu.cu, fcuevas@facii.uho.edu.cu, rodolfo.garcia@live.uleam.edu.ec

[2] Departamento de Tecnología Electrónica, Universidad de Málaga, Málaga, Campus de Excelencia Internacional Andalucía Tech, Málaga, España
gjoya@uma.es, lagos@dte.uma.es

[3] Universidad Laica Eloy Alfaro de Manabí, Facultad de Informática, Manta, Ecuador

[4] Centro para la Investigación y Rehabilitación de las Ataxias Hereditarias, Holguín, Cuba
roberto@ataxia.hlg.sld.cu

Abstract. In this paper we evaluate the use of the machine learning algorithms Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Classification and Regression Trees (CART) and Naive Bayes (NB) to identify non spontaneous saccades in clinical electrooculography tests. Our approach tries to solve problems like the use of manually established thresholds present in classical methods like identification by velocity threshold (I-VT) or identification by dispersion threshold (I-DT). We propose a modification to an adaptive threshold estimation algorithm for detecting signal impulses without the need of any user input. Also, a set of features were selected to take advantage of intrinsic characteristics of clinical electrooculography tests. The models were evaluated with signals recorded to subjects affected by Spinocerebellar Ataxia type 2 (SCA2). Results obtained by the algorithm show accuracies over 97%, recalls over 97% and precisions over 91% for the four models evaluated.

Keywords: Saccade identification · Clinical electrooculography · Classification

1 Introduction

The alteration of eye movements is one of the symptoms of many neurological diseases like Parkinsons syndrome, spinocerebellar ataxias or the Niemann-Pick

I. Rojas et al. (Eds.): IWANN 2015, Part II, LNCS 9095, pp. 56–68, 2015.
DOI: 10.1007/978-3-319-19222-2_5

disease [4]. Specifically in the Spinocerebellar Ataxia type 2 (SCA2) this alteration is an important clinical marker present in more than 90% of patients [29].

There are several kind of eye movements such as saccades, fixations and pursuits. Among them, saccades are critical to follow and evaluate subjects with SCA2. For instance, SCA2 patients have significantly slower saccades and with larger latencies than healthy subjects [29]. The analysis of this kind of movement is very often used in the researches conducted by the medical community, hence its importance.

A technique to measure eye movements called electrooculography consists in capturing the electrical potential of the eyes to calculate its magnitude and direction. This technique is widely used in electrophysiologic tests [16]. The resulting signals of this recording process are named electrooculograms [6].

There exists several methods and algorithms for identifying saccades in electrooculograms, the vast majority of them based on kinetic thresholds [11,14,26,31], using suppervised learning [6,28], unsupervised learning [20] or other novel approachs [18,22] like particle filters [8]. These methods were designed to work in a not constrained scheme having advantages in a lot of scenarios. They 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 posible, not distinguishing which of them are spontaneous and which not.

In a previous work [2], we proposed a method that identifies saccadic movements using a sample-to-sample approach. This method allows us to discriminate whether a sample belong to a saccadic movement or not. Now, in this work we have the task of identifying which of these movements are stimuli related using a feature-based approach.

Here we set out to evaluate the use of machine learning algorithms taking into account the strengths of clinical tests of electrooculography to solve the proposed task. Our approach have to use only horizontal movement signals and stimulus signals, and do not require the use of thresholds or any other user input. To do so, a new set of features were selected to train the models taking into account characteristics of valid saccadic movements.

To identify the ocurrence of saccadic movements we use an impulse detection method based on velocity thresholds. These thresholds are calculated adaptively with a modified version of the method proposed in [18]. Our algorithm uses a classification model to solve the presented task, so we evaluate four of them: Support Vector Machines (SVM) [7], K-Nearest Neighbors (KNN) [27], Classification and Regression Trees (CART) [5] and Naive Bayes (NB) [25]. The performance of the classification models were measured, obtaining very good results ($> 97\%$ accuracy) in all of them.

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

2 Material and Methods

To test the selected algorithms an experiment was designed. The first step was detect potential impulses and annotating them to build a labeled dataset. Then, each classification method is evaluated with stratified k-fold cross validation. Finally, we compare the performance of the models using nonparametric statistical tests to select the fittest.

Clinical tests of electrooculography are setup as follows. Subjects with their head fixed are seated in front to a monitor at a previously known distance. Then, they are commanded to follow a visual stimuli which appears and disappears from one side to the other in the monitor. Capturing eye movements in these conditions using electrooculography allows to researchers the identification of which saccades respond to stimulus and which ones are spontaneous. Also allows 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 bandwith of 0.02 to 70 Hz. Records of 12 sick subjects with SCA2 were used to build a dataset with features extracted from signal impulses. Each one of the records have at least tests of 10°, 20° and 30° of visual stimulation. Typically saccadic tests have at least one horizontal channel and one stimulus signal (Fig. 1).

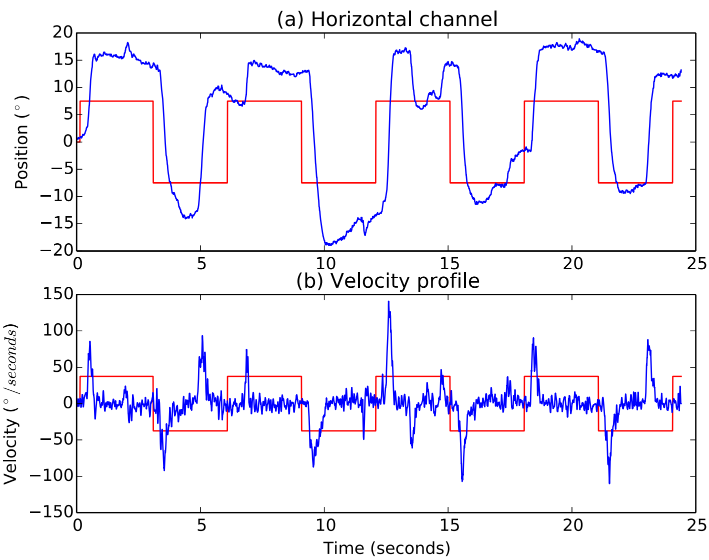

Fig. 1. Typical electrooculography signal with 30° stimulus angle of a subject suffering SCA2. Red signals are the scaled stimuli signals. Blue signals are the horizontal channel (a) and its velocity profile (b) respectively.

IPython notebooks [23] were used in conjunction with the Python language scientific facilities: NumPy [19], SciPy [12], Pandas [17], Matplotlib [10] and Scikit-Learn [21] 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.

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 non desired spectral components produced by equipment malfunction, poor analog filtering or biological artifacts. Differentiation allows to obtain the velocity profile used later by the algorithm.

Median filter (Equation 1) has proven to be very robust in eliminating high frequency signal noise while preserving sharp edges. An study carried out in [13] demonstrated that this kind of filters is appropiate for eye movements signals. To eliminate non desired noise present in the signals used in the experiment, we use a median filter with a window size of 9 samples (approximately 45 milliseconds) obtaining very good results. This is accomplished using the **medfilt** function of SciPy.

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

Due to the discrete nature of these signals, numerical differentiation is employed to calculate the velocity profiles. According [3], Lanczos differentiators (Equation 2) with 11 points ($N = 11$) have good performance for signals with the same characteristics as the ones used in this experiment.

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

We implemented the rutine of a Lanczos 11 differentiator which have the following formula:

$$f'(x^*) \approx \frac{f_1 - f_{-1} + 2(f_2 - f_{-2}) + 3(f_3 - f_{-3}) + 4(f_4 - f_{-4}) + 5(f_5 - f_{-5})}{110h} \tag{3}$$

2.2 Detection of Impulses

Saccadic movements are represented as impulses in a velocity graph (Fig. 1b). Typically, this movements can be easily detected by its contrast in magnitude and shape with other movements such as fixations and microsaccades. However, for the same stimulus angle the range of values of true saccadic impulses vary from subject to subject. This situation is tied greatly on the degree of affectation present in the subject [24].

One of the critical parts of the algorithm is the detection of velocities impulses which can potentially be saccades. For that matter, a threshold is needed to know when the velocity has reached 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 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 Nystrm and Holmqvist in [18]. The algorithm uses the absolute values of the *velocities* samples to calculate the approximation of the initial threshold (*last threshold*). This initial threshold is calculated by adding σ times the standard deviation of the *velocities* to its mean. Then, iteratively it adjusts the *last threshold* with the same formula using only *selected samples* 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 than one degree. The value of the resulting threshold is represented graphically by the red line in Fig. 2.

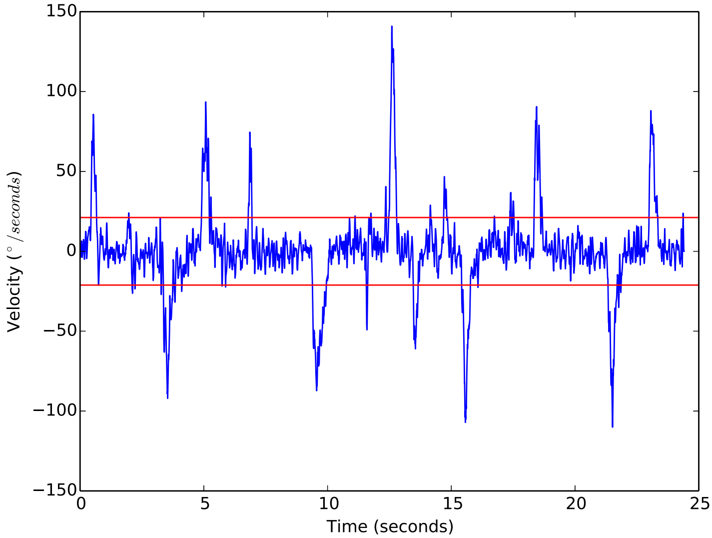

Fig. 2. Threshold estimated in a 30° stimulus angle test of a subject with SCA2

The original algorithm requires the initial threshold as input. This adds a little 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 calculate the initial threshold in a adaptive way using all velocity samples, so eliminating the subjectivity of the

Algorithm 1. Modified version of Nystrm and Holmqvist [18] threshold estimation algorithm

```
Input  : velocity profile (Array of degree/seconds samples)
Input  : σ (Safety margin)
Output: Threshold estimation
begin
    velocities ⟵ Abs(velocity profile);
    last threshold ⟵ Mean(velocities) + σ * Std(velocities);
    current threshold ⟵ 0;
    while Abs(last threshold- current threshold) > 1 do
        selected samples ⟵ samples from velocities below last threshold;
        current threshold ⟵ last threshold;
        last threshold ⟵ = Mean(selected samples) + σ * Std(selected samples);
    return last threshold;
```

original approach. Using the new approach on signals recorded to subjects with SCA2 in different stages seems to be adecuate to the task at hand.

The safety margin ($\sigma = 6$) employed by [18] ignores too many valid saccadic movements in lower angle tests for subjects with SCA2. A value of $\sigma = 3$ seems to be adequate for most cases at the expense of the detection of more non valid impulses. Even when has a penalty in runtime performance, the final accuracy of the method should not decrease significantly. Due the amplitude of this 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 with one of these points, we move to the left and to the right until the velocity is zero or cross it. This approach allows further refinement of the saccade start and ending points because the impulses usually get more samples 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 Fig. 3.

2.3 Model Evaluation

Once we have the saccadic impulses candidates, we need to know if they are saccades and if they are related to the stimulus. For this reason, the strategy behind our approach uses human intuitive features to solve this task. To take advantage of the characteristics of the clinical tests, the following set of features was carefully selected:

Angle: Amplitude of the stimulus, it can take 3 values: 10, 20 or 30.

Absolute Latency: Time between the start of the stimulus transition and the maximal velocity point of the impulse in milliseconds (ms).

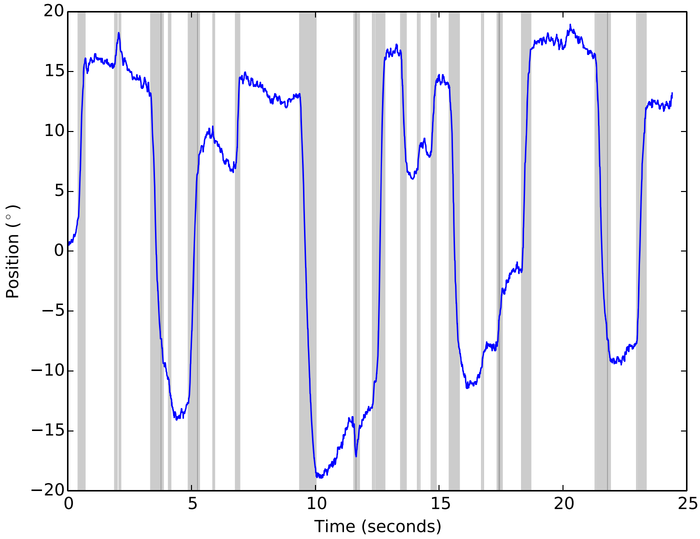

Fig. 3. Identified impulses in the same signal used in Figure 2

Normalized Latency: Normalized version of absolute latency with values between 0 and 1. The value 0 means that the maximal velocity is in the start of the fixation window, and the value 1 means that the maximal velocity is at the end of the fixation window.

Amplitude: Difference between the maximum position value and the minimum position value in the impulse.

Deviation: Difference between amplitude and the angle of the stimulus.

Maximum Velocity: Maximum velocity achieved during the impulse in $^\circ/s$.

Maximum Acceleration: Maximum acceleration achieved during the impulse in $^\circ/s^2$.

Maximum Jerk: Maximum jerk achieved during the impulse in $^\circ/s^3$.

Direction: Take the value 1 if the movement follows the direction of the stimulus or -1 in other case.

End Relative Position: Values between 0 and 1, representing in which side of the stimulus the impulse ends. The value 0 represents the left side and the value 1 the right side.

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 in valid and non valid saccades (Figure 4). As results, 1797 valid saccades and 6809 not valid impulses were obtained, resulting in 8606 instances.

Because we are using Python technologies, Scikit-Learn was selected as machine learning library, hence we are constrained to a restricted set of models

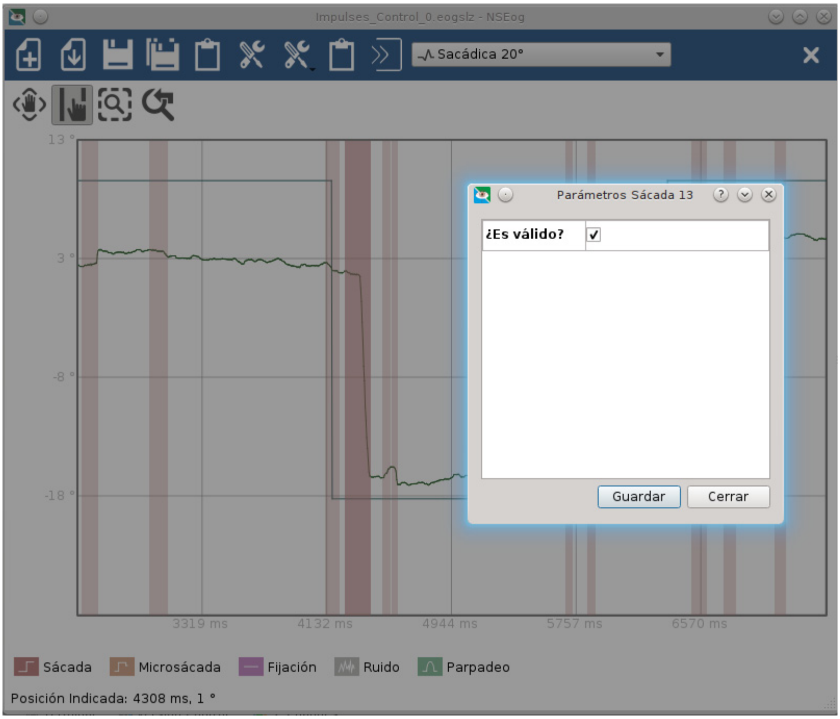

Fig. 4. Impulses annotation with the NSEog platform

implemented in it. The main policy of model selection was family representation, meaning that we try to choose methods with different working principles. So we evaluate four different models: Support Vector Machines, K-Nearest Neighbors, CART decision trees and the Gaussian version of Naive Bayes.

Support Vector Machines (SVMs) are a set of supervised learning methods very effective in high dimensional spaces [7]. There are also very versatile supporting a set of kernel functions. Scikit-Learn implements four kernel functions: linear $\langle x, x'\rangle$, polynomial $(\gamma\langle x, x'\rangle + r)^d$, rbf $e^{-\gamma|x-x'|^2}$ and sigmoid $tanh(\gamma\langle x, x'\rangle + r)$. Results from preliminary experiments showed that for the proposed task, the rbf kernel function have the best performance compared with the others. Further study are necessary to fine tune the parameter γ of this kernel.

K-Nearest Neighbors is a type of instance-based learning which can be used for supervised or unsupervised learning. Instead of creating a generalizing function, it stores all the data inside the models 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 the analized point and predict the label from it [27]. To train our model we tried several numbers of neighbors starting from 2, giving the best results when this value is equal to 3. The data structure used is determined automatically by the Scikit-Learn implementation using optimization techniques.

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 split the data trying to infere decision rules which can be used to clasify instances. Scikit-Learn uses an optimized version of the CART tree that support classification and regression [5]. The implementation used here do not require any parameter by default.

Naive Bayes classifiers are supervised methods based on Bayes theorem which assume independence between every pair of features [25]. We used a gaussian version of this classifier implemented in Scikit-Learn. Like for decision trees, the default implementation of this statistical classifier do not require any parameters.

As validation scheme we use an stratified 10-fold cross validation to evaluate internally the models. The metrics employed to measure the performance were accuracy (Equation 4), recall (Equation 5) and precision (Equation 6) [30]. The accuracy gives a general quality measure of the performance of the models, while the recall and the precision allow to know how well the model predict or miss predict valid saccadic movements. In the following equations, TP (true positives), TN (true negatives), FP (false positives) and FN (false negatives) are the items from the confusion matrix used to compute involved metrics.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

The whole dataset was adjusted by removing the mean and scaled to unit variance. This technique is critical to obtain good results in the training of the RBF kernel version of Support Vector Machines. These scales was saved along with the model for further use by the algorithm.

To compare the real performance of the models, the Friedman's nonparametric statistical test was used as recommended in [9]. In this step we use records not used in the training phase. Each metric were analyzed by separate and the statistical calculations were performed using the Keel tool [1].

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 swap the model if needed. Due the use of the proposed impulse detection algorithm, the need for parameters managed by the user is eliminated.

3 Results

The evaluated models were trained with 8606 impulses, 1797 valid saccades and 6809 invalid ones. Using 10-fold cross validation the internal performance of the trained process was measured with the metrics accuracy, recall and precision. Table 1 shows results above .97 of accuracy, .94 of recall and .90 of precision in all cases.

Table 1. 10-fold cross validation results

Model	Acc.	Rec.	Pre.
SVM	0.9833	0.9750	0.9467
KNN	0.9796	0.9666	0.9376
CART	0.9769	0.9449	0.9445
NaiveBayes	0.9747	0.9817	0.9056

To perform a more objective evaluation, the algorithm was tested against records obtained from five new subjects not used in the training phase. A total of 3797 impulses were evaluated this time, 704 real saccadic impulses and 3093 not saccadic.

Table 2. External validation results by stimulus amplitude

	SVM			KNN			CART			NaiveBayes		
Angle	**Acc.**	**Rec.**	**Pre.**	**Acc.**	**Rec.**	**Pre.**	**Acc.**	**Rec.**	**Pre.**	**Acc.**	**Rec.**	**Pre.**
10	.9765	.9703	.9051	.9659	.9449	.8745	.9636	.9237	.8790	.9575	.9661	.8261
20	.9858	.9837	.9377	.9844	.9837	.9305	.9822	.9633	.9365	.9780	.9837	.8993
30	.9720	.9686	.9038	.9646	.9686	.8745	.9674	.9462	.9017	.9543	.9686	.8372
Mean	.9780	.9742	.9155	.9716	.9657	.8932	.9711	.9444	.9058	.9633	.9728	.8542
Std	.0070	.0082	.0192	.0111	.0195	.0323	.0098	.0198	.0290	.0128	.0095	.0394

Results obtained analysing the performance individually by stimulus angle seems to favor slightly the SVM model (Table 2). However, doing the same analysis using independent subject records shows a more erratic behaviour (Table 3). Because of this situation, the Friedman's nonparametric statistical test was employed to compare the performance of the four models. Each record was considered as an individual dataset and each of the three performance metrics was analyzed independently using the data in Table 3. Results obtained by this method show that there are no significant differences in the performance of these models for a significance level of $p = 0.10$.

Literature about the task proposed in this work is scarce and no methods to specifically solve it were found. However, similar works reported a recall of .89 for $10°$ recordings on healthy subjects [22] and .80 of recall on subjects with Obstructive Sleep Apnea Syndrome (OSAS) [15]. Other related research conducted by Tigges et al. shows an accuracy of .92 [28]. Taking into account that we are dealing with signals recorded to subjects which suffers a very severe neurological disorder, results shown in Table 2 and Table 3 are better than the others presented in the literature.

Table 3. External validation results by subject record

	SVM			KNN			CART			NaiveBayes		
Subject	**Acc.**	**Rec.**	**Pre.**	**Acc.**	**Rec.**	**Pre.**	**Acc.**	**Rec.**	**Pre.**	**Acc.**	**Rec.**	**Pre.**
1	.9881	.9877	.9699	.9881	.9877	.9699	.9796	.9693	.9576	.9881	.9755	.9815
2	.9862	.9935	.9107	.9724	.9610	.8506	.9845	.9870	.9048	.9535	.9935	.7427
3	.9871	.9754	.9444	.9794	.9590	.9141	.9704	.8852	.9231	.9717	.9836	.8571
4	.9799	.9420	.9559	.9784	.9348	.9556	.9741	.9130	.9545	.9756	.9420	.9353
5	.9410	.9685	.8039	.9392	.9843	.7911	.9358	.9528	.7961	.9375	.9685	.7935
Mean	.9765	.9734	.9170	.9715	.9654	.8962	.9689	.9415	.9072	.9653	.9726	.8620
Std	.0201	.0201	.0669	.0189	.0215	.0748	.0193	.0417	.0659	.0199	.0195	.0982

4 Conclusions

In this work we have described a procedure to indentify spontaneous saccades from a set of detected impulses in electrooculography signals. To detect the impulses we made a modification to the algorithm proposed in [18], which consists in adaptively calculate the initial thresholds. This new algorithm avoids the need of thresholds or any other user input and works very well for noisy signals like the ones recorded to subjects with SCA2, which is a difficult task.

To clasify we used and compared four machine learning paradigms: Support Vector Machines, K-Nearest Neighbors, Classification and Regression Trees and Naive Bayes. The procedure has been applied to a database of eye movements recorded to subjects suffering spinocerebellar ataxias. The evaluation of the performance of the different paradigms were carried out using metrics such as Accuracy, Recall and Precision. The four used paradigms achieved an accuracy above 95%, a recall above 92% and a precision above 83% by external validation (using patterns not used for training). Specifically for Support Vector Machines the performance obtained was always above 97%, 96% and 90% for the three metrics respectively. These results exceed widely the reported by the literature in related works.

Acknowledgments. This work has been partially supported by the Universities of Holguín and Málaga through the joint project titled *"Mejora del equipamiento para la evaluación de la rehabilitación de enfermedades neurológicas de especial prevalencia en el oriente de Cuba"*. Also, we would like to thank the Agencia Española de Cooperación Internacional para el Desarrollo (AECID) and the Campus de Excelencia Internacional Andalucía Tech of University of Málaga because of the funding granted to this project. Finally, the feedback of the anonymous reviewers is gratefully acknowledged.

References

1. Alcalá-Fdez, J., Sánchez, L., García, S., del Jesús, M.J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, J., Rivas, V.M., et al.: KEEL: a software tool to assess evolutionary algorithms for data mining problems. Soft Computing **13**(3), 307–318 (2009). http://link.springer.com/article/10.1007/s00500-008-0323-y

2. Becerra, R., Joya, G., Bermúdez, R.V.G., Velázquez, L., Rodríguez, R., Pino, C.: Saccadic Points Classification Using Multilayer Perceptron and Random Forest Classifiers in EOG Recordings of Patients with Ataxia SCA2. In: Rojas, I., Joya, G., Cabestany, J. (eds.) IWANN 2013, Part II. LNCS, vol. 7903, pp. 115–123. Springer, Heidelberg (2013)
3. Becerra Garca, R.A.: Plataforma de procesamiento de electrooculogramas. Caso de estudio: pacientes con Ataxia Espinocerebelosa Tipo 2. Master en matemtica aplicada e informtica para la administracin, Universidad de Holgun, Holgun (2013)
4. Bonnet, C., Hanuka, J., Rusz, J., Rivaud-Pchoux, S., Sieger, T., Majerov, V., Serranov, T., Gaymard, B., Rika, E.: Horizontal and vertical eye movement metrics: What is important?. Clinical Neurophysiology **124**(11), 2216–2229 (2013). http://www.clinph-journal.com/article/S1388245713006378/abstract
5. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and regression trees. CRC Press (1984)
6. Bulling, A., Ward, J., Gellersen, H., Troster, G.: Eye Movement Analysis for Activity Recognition Using Electrooculography. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(4), 741–753 (2011)
7. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning **20**(3), 273–297 (1995). http://link.springer.com/article/10.1007/BF00994018
8. Daye, P.M., Optican, L.M.: Saccade detection using a particle filter. Journal of Neuroscience Methods **235**, 157–168 (2014)
9. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research **7**, 1–30 (2006). http://dl.acm.org/citation.cfm?id=1248548
10. Hunter, J.D.: Matplotlib: A 2D Graphics Environment. Computing in Science and Engineering **9**(3), 90–95 (2007)
11. Inchingolo, P., Spanio, M.: On the Identification and Analysis of Saccadic Eye Movements-A Quantitative Study of the Processing Procedures. IEEE Transactions on Biomedical Engineering BME **32**(9), 683–695 (1985)
12. Jones, E., Oliphant, T., Peterson, P.: SciPy: Open source scientific tools for Python (2001). http://www.scipy.org/, http://www.citeulike.org/group/2018/article/2644428
13. Juhola, M.: Median filtering is appropriate to signals of saccadic eye movements. Computers in Biology and Medicine **21**(1–2), 43–49 (1991). http://www.sciencedirect.com/science/article/pii/0010482591900347
14. Juhola, M., Jäntti, V., Pyykkö, I., Magnusson, M., Schalén, L., Åkesson, M.: Detection of saccadic eye movements using a non-recursive adaptive digital filter. Computer Methods and Programs in Biomedicine **21**(2), 81–88 (1985). http://www.sciencedirect.com/science/article/pii/0169260785900665
15. Magosso, E., Provini, F., Montagna, P., Ursino, M.: A wavelet based method for automatic detection of slow eye movements: A pilot study. Medical Engineering & Physics **28**(9), 860–875 (2006). http://www.sciencedirect.com/science/article/pii/S1350453306000154
16. Marmor, M.F., Zrenner, E.: Standard for clinical electro-oculography. Documenta Ophthalmologica **85**(2), 115–124 (1993). http://dx.doi.org/10.1007/BF01371127
17. McKinney, W.: Data Structures for Statistical Computing in Python, pp. 51–56 (2010). http://conference.scipy.org/proceedings/scipy2010/mckinney.html
18. Nystrm, M., Holmqvist, K.: An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. Behavior Research Methods **42**(1), 188–204 (2010). http://link.springer.com/article/10.3758/BRM.42.1.188

19. Oliphant, T.E.: Python for scientific computing. Computing in Science & Engineering **9**(3), 10–20 (2007). http://scitation.aip.org/content/aip/journal/cise/9/3/10.1109/MCSE.2007.58
20. Otero-Millan, J., Castro, J.L.A., Macknik, S.L., Martinez-Conde, S.: Unsupervised clustering method to detect microsaccades. Journal of Vision **14**(2), 18 (2014)
21. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011). http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html
22. Pettersson, K., Jagadeesan, S., Lukander, K., Henelius, A., Hæggström, E., Müller, K.: Algorithm for automatic analysis of electro-oculographic data. Biomedical Engineering Online **12**(1), 110 (2013)
23. Prez, F., Granger, B.E.: IPython: A System for Interactive Scientific Computing. Computing in Science & Engineering **9**(3), 21–29 (2007). http://scitation.aip.org/content/aip/journal/cise/9/3/10.1109/MCSE.2007.53
24. Rodrguez-Labrada, R., Velzquez-Prez, L.: Eye movement abnormalities in spinocerebellar ataxias. Spinocerebellar Ataxias, pp. 59–76. Intech, Rijeka (2012)
25. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 3rd edn. Prentice Hall, Upper Saddle River (December (2009)
26. Salvucci, D.D., Goldberg, J.H.: Identifying fixations and saccades in eye-tracking protocols. In: ETRA 2000, pp. 71–78. ACM, New York (2000). http://doi.acm.org/10.1145/355017.355028
27. Silverman, B.W., Jones, M.C.: An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation. International Statistical Review / Revue Internationale de Statistique **57**(3), 233–238 (1989). http://www.jstor.org/stable/1403796
28. Tigges, P., Kathmann, N., Engel, R.R.: Identification of input variables for feature based artificial neural networks-saccade detection in EOG recordings. International Journal of Medical Informatics **45**(3), 175–184 (1997). http://www.sciencedirect.com/science/article/pii/S1386505697000427
29. Velzquez-Prez, L., Rodrguez-Labrada, R., Garca-Rodrguez, J.C., Almaguer-Mederos, L.E., Cruz-Mario, T., Laffita-Mesa, J.M.: A comprehensive review of spinocerebellar ataxia type 2 in cuba. The Cerebellum **10**(2), 184–198 (2011). http://link.springer.com/article/10.1007/s12311-011-0265-2
30. Witten, I.H., Frank, E., Hall, M.A.: Data Mining. Practical Machine Learning Tools and Techniques, 3 edn. Morgan Kaufmann (2011)
31. Wyatt, H.J.: Detecting saccades with jerk. Vision Research **38**(14), 2147–2153 (1998). http://www.sciencedirect.com/science/article/pii/S0042698997004100