Artifact Detection in EEG using Machine Learning

Elena Nedelcu[1], Raluca Portase[1], Ramona Tolas[1], Raul Muresan[2], Mihaela Dinsoreanu[1], and Rodica Potolea[1]

[1]Computer Science Department of Technical University, Cluj-Napoca
[2]Romanian Institute of Science and Technology, Coneural, Cluj-Napoca

***Abstract*—The electroencephalography (EEG) data records vast amounts of human cerebral activity yet is still reviewed primarily by human readers. Most of the times, the data is contaminated with non-cerebral originated signals, called artifacts, which could be very difficult to visually detect and, undiscovered, could damage the neural information analysis. The purpose of our work is to detect the artifacts by identifying the most relevant features, both in the temporal and frequency domains, and train various supervised learning algorithms: Decision Trees, SVM and KNN, in order to distinguish between clean and contaminated signals. The performance of our method exceeds the ones achieved in literature with an accuracy of detection of 98.78%, 98.30% for precision and 98.40% for recall, for the best settings we found.**

***Index Terms*—Artifact detection, EEG, learning from imbalanced data, feature extraction and selection**

I. Introduction

The electroencephalogram (EEG) is a technique used for recording the cerebral activity of the brain. It is frequently used in analyzing neurological pathologies, as well as in brain interaction researches. Besides cerebral activity, EEG records other electrical signals which are generically called artifacts. The analysis of the exam can be erroneously interpreted given the contamination with artifacts, this being the main reason for the artifacts removal process.

Several studies concerning the identification of artifacts have been conducted using algebraic methods [1], Independent Component Analysis [2], Principal Component Analysis [3] or different techniques of machine learning [4], [5], [6], [7].

Two of the most common artifact types are the ocular and muscular artifacts, which are interfering with most of the electrodes. The concern of the paper is the identification and elimination of these types of artifacts. The focus of our methods is to reduce the number of artifacts which were wrongly classified as brain signals. In this respect, we analyze the performance of decision trees, k-nearest neighbors and support vector machine based approaches.

This paper is organized as follows: In Section 2 we briefly describe the ocular and muscular artifacts, data acquisition and segmentation of the EEG data, as well as the proposed features for classification. The methods used are presented in Section 3, followed by the report of the results in Section 4. Section 5 contains the concluding remarks of our work.

[1]978-1-5386-3368-7/17/$31.00 ©2017 IEEE

II. Theoretical background

A. Ocular and muscular artifacts

The ocular and muscular artifacts are included in the physiological category of artifacts, which are signals produced also by the human body parts, but others than the brain.

The signals coming from the eye movements prevail in the frontal and prefrontal regions of the brain. According to [8], in the prefrontal lobe, a descending peak in the negative region of amplitude means an eye opening event, while an ascending peak in the positive region of amplitude marks an eye closing event. The amplitude of this signals is a lot higher comparing with the one of the brains output. According to [9], the blink artifacts last between 200-400 ms and can produce an electric magnitude up to 10 times bigger than the cerebral signals. Moreover, they are characterized by a low frequency, under 4 Hz.

On the other hand, muscular artifacts add an excessive amount of high-frequency signal components to normal EEG signals in the time domain, and overwhelm the EEG power spectrum in the Beta and Gamma bands in the frequency domain. Their characteristics are not actually specified, because they vary too much from a subject to another. The uncontrollable nature of muscle artifacts increases the EEG variability, and thus reduces the EEGs power to detect the functional states of the brain.

B. Data acquisition

The EEG signal was acquired with a headset according to the 10-20 system standard [10] with 128 channels. Each channel records the cerebral activity for a region with an electrode. The sampling frequency is 512 Hz. Each sample taken was stored into a binary file, the result of the data acquisition being 128 binary files containing recordings from each channel. Our signal analysis is based on recordings from a single subject collected in the experimental setting detailed in [11].

C. Segmentation of EEG signal

The signal received from the recording machine is 30-60 minutes long. The length of an artifact is on average 300 ms, so identifying a pattern with a length of 0.012 from the total length of the signal would not bring good results. Therefore a segmentation of the signal was necessary.

There are two fundamental methods for signal partitioning: segmentation with overlapping or non-overlapping segments. If the signal is split in non-overlapping segments, there is the

possibility for an artifact to be at one of the extremities of a window, and that artifact will not be detected. The solution for this problem is to use overlapping segments.

Two aspects are important in performing overlapping segmentation for a signal: the length of the segment and the step of moving the window of segmentation. Taking into consideration the length of an artifact and the fact that a segment has to be large enough to include an artifact, the length for a segment was defined to be 1 second. The length could be smaller if only the length of the artifact was important, but the size of the segment does influence the frequency accuracy when performing Fourier transform. This aspect is described in another chapter from this section. The shifting step of the window was set to 250 ms. The reason for this choice was the average length of an artifact (between 200-400 ms).

D. Features

We chose 16 features to reflect the relevant information of the signal in our classifications. All can be mapped to three categories, as follows: temporal domain features, which reflect the morphology of the signal, frequency domain features, which quantify the changes in the frequency spectrum of the EEG signal and entropy-based features, which describe the uncertainty in the variation of artifacts.

Time-domain features: The standard deviation is a measure that is used to quantify the amount of variation of a set of data values. A low standard deviation indicates that the data points tend to be close to the mean of the set, while a high standard deviation indicates that the data points are spread out over a wider range of values. For our experiments, we measured the absolute standard deviation of the EEG signal.

Skewness is a measure of the asymmetry of the probability distribution of a real-valued random variable. A negative skew indicates that the tail on the left side of the probability density function (PDF) is longer than that of the right side, with the bulk of the values lying to the right of the mean. Conversely, a positive skew indicates that the tail on the right side is longer than that of the left and the bulk of the values lie to the left of the mean [12]. For a segment x_j and mean μ_j, the skewness is defined as:

$$Skewness(x_j) = \frac{\frac{1}{n_s}\sum_{i=1}^{n_s}(x_j(i)-\mu_j)^3}{(\frac{1}{n_s}\sum_{i=1}^{n_s}(x_j(i)-\mu_j)^2)^{\frac{3}{2}}}$$

The root mean square (RMS) amplitude, or quadratic mean, is a statistical measure of the magnitude of a time varying quantity. The RMS amplitude expresses the mean of the absolute amplitude of an epoch x_j. As EEG artifacts are often high-energy, high-amplitude signals, the RMS amplitude aims to capture this trait [12].

Kurtosis, often referred to as the fourth central moment, is a measure of the peakedness of a probability density function. The kurtosis has a higher value when one of the following events happens: eyes opening, eyes closing, blink. For a segment x_j, the kurtosis is defined as:

$$Kurtosis(x_j) = \frac{\frac{1}{n_s}\sum_{i=1}^{n_s}(x_j(i)-\mu_j)^4}{(\frac{1}{n_s}\sum_{i=1}^{n_s}(x_j(i)-\mu_j)^2)^2}$$

The population mean or expected value refers to one measure of the central tendency of a probability distribution.

The median is the value separating the higher half of a data sample, a population, or a probability distribution, from the lower half.

Entropy-based features: Shannon entropy is a measure in information theory for estimating the uncertainty of an outcome.

Frequency-domain features: As mentioned above, the frequency bands are an important indicator of the type of signal represented in the EEG segment. For including these characteristics in the classification, Discrete Fourier transform was used. This transformation maps a signal of N points into two vectors, each of M+1 points, where M is $\frac{N}{2}$. The two vectors, real and imaginary coefficients, represent the quantity of influence that the corresponding basis function ($c_k[i] = cos(\frac{2\pi ki}{N})$, $s_k[i] = sin(\frac{2\pi ki}{N})$) has in the original signal. There are different methods for finding the coefficient vectors (i.e. algebraic-based, correlation-based). The method used in this study was FFT (Fast Fourier Transform). After calculating the FFT, every frequency has a coefficient of influence for the signal. An average was computed for each frequency band and this value is one of the features that characterize a segment.

The main waveforms were considered in the following ranges: Alpha spectrum 8-12 Hz, Beta low spectrum 12-20 Hz, Beta high spectrum 20-30 Hz, Gamma low spectrum 30-60 Hz, Gamma high spectrum 60-120 Hz, Delta spectrum 0-4 Hz, Theta spectrum 4-7 Hz.

The following two features describe the relation between the recording channels in the same region, so also the influence of the artifacts on multi-channel segments could be analyzed. For both of them, the correlation increases proportionally with the distance from 0, so the absolute value was considered.

Pearson correlation coefficient is a measure of the linear correlation between two variables X and Y. It has a value between +1 and -1, where 1 is total positive linear correlation, 0 is no linear correlation, and -1 is total negative linear correlation.

Cross correlation is a measure of the similarity of two signals as a function of the displacement of one relative to the other, and its maximum peak reflects the point in which the analyzed signals are the most correlated.

III. Methods

An important aspect in performing classification is the presence or absence of the labels in the training set. The received data was not labeled, so there were two options available: performing unsupervised learning or labeling the data.

Unsupervised learning was tried on the EEG data. We performed k-means based clustering. This algorithm is efficient if the number of clusters is known (3 clusters: brain signal, ocular, muscular). The clustering algorithm classified wrong 28.65% of the segments. This percentage is not a satisfying

result, so the data was manually labeled and supervised learning was used.

We used different types of supervised learning algorithms and each of them is presented in the following sections. The objective of our algorithms configurations used for the experiments is to minimize the false negative rate, without affecting too much the accuracy of the detection.

Data labeling: The labeling was made using a visualization tool and analyzing the EEG signal from the recording channels step by step. The existing artifacts were identified by morphological characteristics and tagged. The process was slow and difficult, but necessary because of the need of big reliable data.

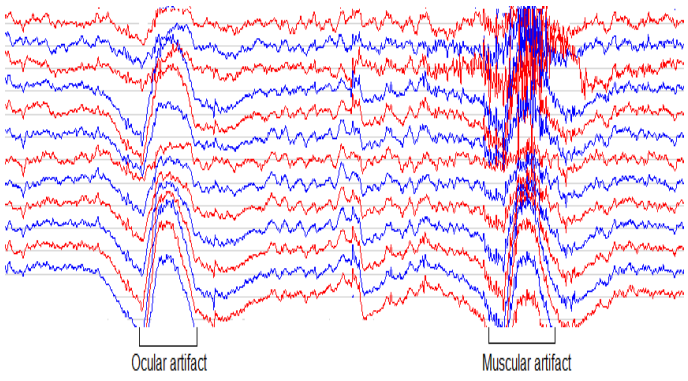

Fig. 1. Visual representation of the signal of the EEG signal

The Figure 1 presents part of the signal of the original unlabeled data from 14 different channels from the frontal lobe. We highlighted parts of the signal which are artifacts, the rest being clean brain data.

After the labels acquisition and signal segmentation, we generated images of each segment with the associated label, which were validated by a domain expert.

Feature selection: To increase the classification results and to enhance generalization by reducing over-fitting, a feature selection process was performed.

For this process, we used the information gain as a metric for ranking the features, and the whole training set was considered for creating the predictive model.

A. SVM

Support Vector Machine (SVM) [13] is a supervised technique highly used in classification due to its generalization performance and fast computing results. The main advantages of using SVM are that it has a regularization parameter for avoiding over-fitting, it uses the kernel trick that can be engineered for the problem in case, it is an approximation to a bound on the test error rate and offers results in small computing time. More than this, because it has the capacity to offer a good generalization independent of the input space dimensions, SVM is considered a good candidate for analyzing biomedical data like multichannel recordings of EEG.

B. Decision Trees

A decision tree is a graphical representation that makes use of branching methodology to exemplify all possible outcomes of a decision, based on certain conditions.

For performance reasons, a REP (Reduced Error Pruning) Decision Tree algorithm was chosen [14]. Starting at the leaves, each node is replaced with its most popular class. If the prediction accuracy is not affected then the change is kept. This algorithm has the advantage of simplicity and speed. The maximum depth of the tree is not limited and the best results were obtained with a number of 3 folds for pruning.

The most important reasons for choosing the Decision Tree algorithm are the fact that it creates a model that is easy to interpret and analyze and the fact that it automatically selects the most relevant features, allowing us to experiment with multiple combinations of them.

C. KNN

KNN (K nearest neighbor) is an algorithm that is considered to be of type "lazy" because it doesn't perform any specific training phase except of reading the training set in memory. The algorithm constructs a model by reading the entire training set and for each test point it calculates K points that are closest to the test point. These points are called the K nearest neighbors. From the computed set, with a voting algorithm, the label that is assigned to the majority of the neighbors is chosen. This label is assigned to the test point.

The advantages of KNN are: fast training, easy to learn and implement, but there is a significant disadvantage, the algorithm is a memory consumer [15].

For computing the set of neighbors, different metrics can be used. Some of the most popular metrics are presented below:

$$d_{Euclidian} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$d_{Chebyshev} = max(|x_i - y_i|)$$

$$d_{Manhattan} = \sum_{i=1}^{n}|x_i - y_i|$$

$$d_{Minkovski} = (\sum_{i=1}^{n}|x_i - y_i|^q)^{1/q}$$

Another critical aspect which influences the result of the algorithm is choosing the value for k. A small value for k means that the noise highly influences the results and a large value of k is computationally expensive. According to [16] a common choice for k is $\sqrt{n}$ where n is the dimension of the training set. Experiments with different values of k (starting with k = 1) and different metrics for calculating the distance between two neighbors were made. The results of these experiments and their interpretation are presented in Section *Results and discussion*.

IV. Results and discussion

A. Experiments setup

1) Experiment 1: Table I presents the number of observations from the training set and testing set used for this experiment. An observation is represented in this experiment by 16 features of a EEG segment: mean, median, RMS, standard deviation, delta spectrum, alpha spectrum, beta spectrum, theta spectrum, gamma low spectrum, beta high spectrum, gamma high spectrum, skewness, kurtosis, entropy, Pearson coefficient, correlation. These features were introduced in the theoretical section of this paper.

Type of data	Dataset size
Train Set Size	33217
Clean Labeled Instances	28498
Ocular Labeled Instances	1513
Muscle Labeled Instances	3206
Test Set Size	14535
Clean Labeled Instances	13096
Ocular Labeled Instances	442
Muscle Labeled Instances	997

TABLE I
CONFIGURATION USED FOR THE EXPERIMENTS NO. 1 AND NO. 2

2) Experiment 2: This experiment uses the same dataset as Experiment 1 presented in Table I. The difference consists of the features used for describing an observation. In this experiment we use only 9 features: standard deviation, RMS, delta spectrum, theta spectrum, gamma low spectrum, gamma high spectrum, entropy, Pearson coefficient, correlation.

3) Experiment 3: The third experiment is focused on the unbalanced characteristic of the dataset used in the first two experiments. We aim to construct a balanced dataset for training. The resulting training set distribution is presented in Table II, where the column labeled as *Original* contains the original distribution of data and *Unders* presents the distribution obtained after random undersampling of the majority class where we keep $\frac{1}{6}$ of the data. The last column has the dataset size achieved after oversampling of the artifacts instances with SMOTE and undersampling of the brain signal while keeping a ratio of $\frac{1}{4}$ brain data. For testing, the dataset remained the same

	Training set distribution		
Signal type	Original	Unders	Unders + SMOTE
Dataset dimension	33217	9394	18784
Brain signal	28498	4675	9431
Ocular artifacts	3206	3206	6243
Muscular artifacts	1513	1513	3110

TABLE II
CONFIGURATION FOR THE EXPERIMENT NO. 3

as for Experiment 1 and Experiment 2, which can be observed in Table I. In this experiment an observation is represented by 16 features, the same features that were used in Experiment 1.

B. Classification setup

1) SVM configuration: SVM supports several types of kernel functions. In general, for a small number of features a linear kernel tends to perform very well. It was proven that the linear kernel is a degenerate version of RBF. Radial basis function kernel, also known as RBF, is a very popular kernel function used in classification problems and it is represented by the function

$$k(x_i, x_j) = exp(-\gamma||x_i - x_j||^2)$$

In our case, using this kernel function resulted in the highest gain of the accuracy.

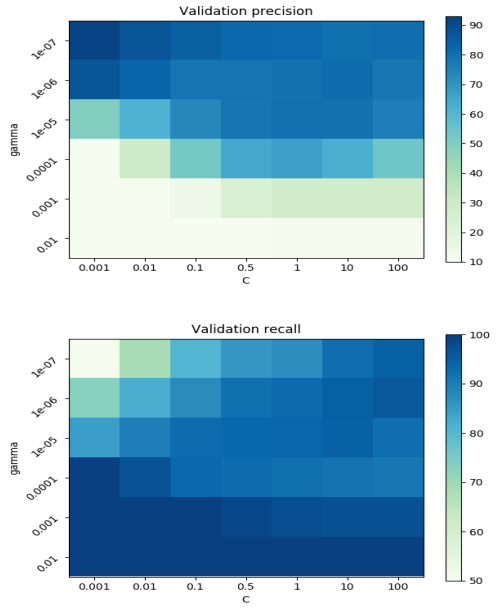

Fig. 2. Precision and recall results for RBF kernel of SVM

For choosing the parameters of the learning module we experimented with different values for γ, the parameter of the kernel function and c, the cost used for choosing the maximum-margin hyperplane. We registered the precision as well as the recall for each of the experiments, as shown in Fig. 2, where we used for $gamma$ axis a logarithmic scale. The figure consists of two intensity graphs, where the intensity of the color represents the value of the two metrics. It can be observed that the best setup with respect to the harmonic mean of precision and recall is given by $\gamma = 0.000001$ and $c = 10$, these being the parameters used in the following experiments.

2) KNN configuration: As mentioned in the *Methods* Section, different configurations were used for choosing the best of them in experimenting with KNN. The results of these experiments are presented next.

It was noticed that as k increases, the performance of the algorithm increases as well. For the constructed training set which has a dimension of 33200 points $\sqrt{n}$ is 181, but this value is too large for being chosen as k because the time to complete the classification is also important. Table III presents the most significant configurations for the algorithm. In this table a configuration is defined by the metric used for distance and the value of k.

The performance metrics considered for each configuration are: accuracy, average performance, average recall and another function based on recall that is going to be referred as binary recall in the next sections of this paper.

Accuracy is a ratio of correctly predicted observations to the total observations. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. As the experiment is based on multi-class labeling, positive and negative observations could not be defined, so an average precision is considered. This average is calculated by first considering the brain signal as positive and other observations (ocular and muscular artifacts) as negative, then considering segments with ocular artifact as positive observations and the others as negative and similar for muscular artifacts. In this manner we obtain 3 values for precision. These values are weighted with the number of instances from every class and a weighted average is obtained. The same algorithm is used for calculating the average recall. Recall is the ratio of correctly predicted positive observations to the all observations in one class. Maximizing the recall means to minimize false negatives from one class and this is important because we aim to predict that there is an artifact in a clean segment rather than to omit the existence of an artifact in an affected segment. However, we ultimately want to detect that there is an artifact in the signal, even if we don't detect the correct type of artifact. This is the reason why binary recall was introduced. This measure is different from recall because it uses the classification as a binary one: artifact or clean segment. All segments classified as ocular that are muscular (and muscular classified as ocular) together with the correctly classified ocular and muscular segments are considered correctly classified artifact segment. These segments are the true positive examples from the formula describing the recall (1). False negatives are the segments classified as clean EEG signal, but are ocular or muscular artifacts.

$$R_b = \frac{TP}{TP + FN} \tag{1}$$

It can be observed that there is no configuration that has the best accuracy, the best performance and also the best recall. Different configurations are the best option for different metrics. For choosing a final configuration, an utility function was defined. This function is described in 2 where BR stands for binary recall, R for recall, P for precision and A for accuracy. There is also a constraint, presented in 3.

$$f_{utility} = \alpha * BR + \beta * R + \gamma * P + \theta * A \tag{2}$$

$$\alpha + \beta + \gamma + \theta = 1 \tag{3}$$

The weights for the metrics were defined based on the impact they should have in choosing the right configuration. The binary recall should be the most important and it weights as all the others together. The weights used for accuracy, precision and recall were assigned on the fact that recall and precision are equally important and accuracy has a smaller weight because the dataset is unbalanced, as mentioned above. According to this matter, α was defined 0.5, β and γ equal to 0.2 and θ 0.1.

All metrics were included in the utility function, but weighted because of the above mentioned issue (it is better to classify a clean segment as an affected one than to miss a artifact from a segment).

Config	Ac.	P	Recall	Recalc. rec	Utility fct
Manhatten-10	97.36	97.2	97.4	99.66	98.52
Manhatten-6	97.46	97.3	97.5	99.64	98.48
Euclidian-25	97.14	97.0	97.1	99.61	98.33
Cebasev-25	96.60	96.4	96.6	99.48	98.00
Manhatten-25	97.44	97.3	97.4	99.66	98.51
Minkovski-25	97.14	97.0	97.1	99.61	98.33
Euclidian-51	96.97	96.8	9.7	99.709	98.31
Cebasev-51	96.24	96.0	96.2	99.557	97.84
Manhatten-51	97.19	97.1	97.2	99.702	98.43
Minkovski-51	96.97	96.8	97.0	99.70	98.31

TABLE III
RESULTS OF RUNNING KNN ALGORITHM WITH DIFFERENT CONFIGURATIONS

Small values for k were not presented in the table because they give smaller performances. Starting from 6 neighbors k was increased in order to see if the performance increases as well. It can be observed from the table that increasing k from 10 to 25 and then 51 doesn't bring any improvement in the utility function.

The configuration that maximizes the utility function is the Manhattan metric with 10 neighbors for the distance measure.

C. Feature selection results

After the selection process, the features were ranked by information gain. The results are presented in Fig 3.

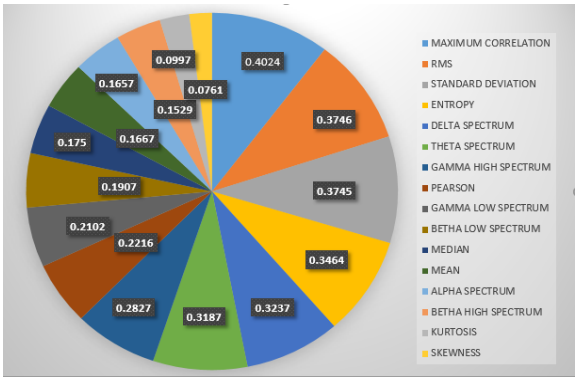

Fig. 3. Scores for each of the feature, by information gain

The best ranked features, which scored higher than 0.2, as can be seen, are: the maximum correlation coefficient, RMS, standard deviation, entropy, delta spectrum, theta spectrum, gamma high spectrum, Pearson coefficient and gamma low spectrum. For better results, an experiment with only those nine features was also performed, and the variation of the evaluation metrics could be seen in the following subsection.

D. Experiments results

Table IV presents the results of the methods used for classifying EEG segments. All the algorithms used give close values for our performance metrics, this being an evidence that the results are accurate.

Depending on the desired result, one algorithm or another can be chosen for performing the classification. Decision trees and KNN maintain an almost constant percentage for all the metrics presented, while SVM performs better from the perspective of accuracy and true negative rate.

	Classifier					
Metrics	Decision tree %		knn %		svm %	
Features no	16	9	16	9	16	9
Accuracy	98.27	98.27	97.46	98.35	98.31	98.78
Recall	98.30	98.30	97.50	98.40	91.67	93.55
True Negative Rate	99.25	99.25	99.63	99.67	99.27	99.43
Precision	98.30	98.30	97.30	98.30	93.15	94.79

TABLE IV
COMPARISON OF THE CLASSIFIERS RESULTS AFTER THE EXPERIMENT NO. 1 AND NO. 2

The results of Experiment 1 are presented in Table IV. These are obtained using all the features described in the theoretical section, 33217 instances in the training set and 14535 instances in the testing set. A new configuration for the supervised learning algorithms was tried, by removing the lowest ranked features and maintaining the same number of instances for train and test. This is the Experiment 2. The Decision tree algorithm has the same results because feature selection was already performed in the initial experiment for this algorithm, but KNN and SVM brings an improvement to all of the used metrics, as it can be observed in Table IV.

Balanced training set results: Given the fact that we have a highly unbalanced dataset, with the ratios between ocular artifacts, muscular artifacts and brain signal $1 : 2 : 30$, the effects on the results after balancing the dataset were investigated. In Experiment 3 we used random undersampling to achieve the same number of brain signal instances as the number of artifacts. Moreover, we used the SMOTE Method [17] to create artificial data such that we doubled the number of instances of the minority class. We combined this with random undersampling and we tested with the same test configuration from Experiment 2. The resulting training set distribution is presented in Table II.

Generalizing the minority class can increase the precision of the classification. The effects of balancing the training set are showed in Table V. Comparing with the results of Experiment 2, it can be seen that we have worse results for the decision tree and KNN, while for SVM we increase the recall with 3%. It should be mentioned that with SVM a significant decrease in precision is obtained. Depending on the metrics of interest to be improved and the tool used for supervised learning, undersampling can and should be used, but the authors do not suggest the overhead of combining undersampling with oversampling methods for this problem.

Clasifier	Dataset balancing method	Precision	Recall
Decision tree	Undersampling	97.80	97.41
	Undersampling+SMOTE	96.90	96.13
KNN	Undersampling	97.20	97.10
	Undersampling+SMOTE	97.02	97.00
SVM	Undersampling	84.12	95.16
	Undersampling+SMOTE	84.97	96.07

TABLE V
RESULTS OBTAINED AFTER THE EXPERIMENT NO. 3

E. Comparison with related work

There are different approaches for detecting artifacts, one widely used is based on ICA (Independent Component Analysis). This method is based on decomposing the signal in independent functions and detecting the function that is the artifact. The method assumes that by removing that function and recomposing the signal without it, the real signal without the artifact will be generated. This method is not completely independent by the human expert, because the component that is generated by the artifact has to be chosen after the algorithm is performed. A major advantage of the currently proposed method is the fact that no involvement of the human expert has to be done when running the detection program. The features were obtained by performing the initial labeling and considering this a-priori information in the following phases.

Other work on the artifact detection problem is [18] where the author uses unsupervised learning techniques for classification (one method being clustering). By applying those methods, an average of 76% correctly classified instances by testing on different subjects are obtained. The best classification percentage is 91%. Our proposed methods based on supervised learning bring an important improvement of this percentage as shown in Table IV where it can be seen that the accuracy of all 3 techniques is greater than 98%.

V. CONCLUSION

This paper presents a machine-learning technique for detecting ocular and muscular artifacts in the EEG signal. The proposed method splits the signal in overlapping segments and applies an algorithm of classification for each segment for detecting the presence of artifacts. We also analyzed and proposed a set of relevant features for artifact detection. Different supervised learning algorithms were used for classifying the signal and the performance of each method was analyzed. The proposed methods do not require any human expert visualization or intervention, providing an automatic detection method of artifacts and they perform better than the unsupervised methods, to the best of our knowledge. The features selected to be used in classification are the key of obtaining high performance.

ACKNOWLEDGMENT

The authors would like to thank the Experimental and Theoretical Neuroscience Lab at Coneural Cluj-Napoca for providing the data on which the study was performed. Work supported by CNCS-UEFISCDI, grant PN-III-P2-2.1-PED-2016 -0007, contract 199 PED/2017.

REFERENCES

[1] Z. Tiganj, M. Mboup, C. Pouzat, and B. Lotfi, "An algebraic method for eye blink artifacts detection in single channel EEG recordings," in *17TH INTERNATIONAL CONFERENCE ON BIOMAGNETISM ADVANCES IN BIOMAGNETISM – BIOMAG2010*, vol. 28, no. 6, Dubrovnik, Croatia, Mar. 2010, pp. 175–178. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00518627

[2] A. Delorme, T. Sejnowski, and S. Makeig, "Enhanced detection of artifacts in eeg data using higher-order statistics and independent component analysis," *Neuroimage*, vol. 34, no. 4, pp. 1443–1449, 2007.

[3] P. Bhuvaneswari and J. S. Kumar, "Characterization of principal component analysis for eye artifact detection in eeg signal."

[4] N. A. Chadwick, D. A. McMeekin, and T. Tan, "Classifying eye and head movement artifacts in eeg signals," in *5th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2011)*, May 2011, pp. 285–291.

[5] R. Singla, B. Chambayil, A. Khosla, and J. Santosh, "Comparison of svm and ann for classification of eye events in eeg," *Journal of Biomedical Science and Engineering*, vol. 4, no. 01, p. 62, 2011.

[6] S. Y. Shao, K. Q. Shen, C. J. Ong, X. P. Li, and E. P. V. Wilder-Smith, "Automatic identification and removal of artifacts in eeg using a probabilistic multi-class svm approach with error correction," in *2008 IEEE International Conference on Systems, Man and Cybernetics*, Oct 2008, pp. 1134–1139.

[7] M. A. Sovierzoski, F. I. M. Argoud, and F. M. de Azevedo, "Identifying eye blinks in eeg signal analysis," in *2008 International Conference on Information Technology and Applications in Biomedicine*, May 2008, pp. 406–409.

[8] R. J. Brijil Chambayil, Rajesh Singla, "Eeg eye blink classification using neural network."

[9] I. M. B. Nunez, "Eeg artifact detection."

[10] D. T. Valer Jurcak and I. Dan, "10/20, 10/10, and 10/5 systems revisited: Their validity as relative head-surface-based positioning systems," 2007.

[11] V. V. Moca, I. Ţincaş, L. Melloni, and R. C. Mureşan, "Visual exploration and object recognition by lattice deformation," *PloS one*, vol. 6, no. 7, p. e22831, 2011.

[12] S. H. . O'Regan, "Artefact detection and removal algorithms for eeg diagnostic systems. phd thesis, university college cork."

[13] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995. [Online]. Available: http://dx.doi.org/10.1023/A:1022627411411

[14] T. Elomaa and M. Kaariainen, "An analysis of reduced error pruning," *Journal of Artificial Intelligence Research*, vol. 15, pp. 163–187, 2001.

[15] N. Bhatia, "Survey of nearest neighbor techniques," 2010.

[16] D. G. Richard O. Duda, Peter E. Hart, *Pattern classification Second Edition*.

[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Int. Res.*, vol. 16, no. 1, pp. 321–357, Jun. 2002. [Online]. Available: http://dl.acm.org/citation.cfm?id=1622407.1622416

[18] S. B. Shaibal Barua and P. F. Mobyen Uddin Ahmed, "Classification of ocular artifacts in eeg signals using hierarchical clustering and case-based reasoning."