UNDER REVIEW

RESCIENCE C

Replication

[Re] Deep Convolution Neural Network and Autoencoders-Based Unsupervised Feature Learning of EEG Signals

Bruno Aristimunha[1], Diogo Eduardo Lima Alves[1], Walter Hugo Lopez Pinaya[1, 2], Raphael Y. de Camargo[1]

[1]Center for Mathematics, Computation and Cognition (CMCC), Federal Univesity of ABC (UFABC), Rua Arcturus, 03. Jardim Antares, São Bernardo do Campo, CEP 09606-070, SP, Brazil. – [2]Department of Psychosis Studies, Institute of Psychiatry, Psychology & Neuroscience, King's College London, London, UK

Edited by
(Editor)

Reviewed by
(Reviewer 1)
(Reviewer 2)

Received
–

Published
–

DOI
–

This paper presents our efforts to implement, acquire similar results and improve the ones achieved by the authors of the original article. We follow the steps and models described in their article and the same public data sets of EEG Signals. Epilepsy affects more than 65 million people globally, and EEG Signals are critical to analyze and recognize epilepsy. Although the efforts in the last years, it is still challenging to extract useful information from these signals and select useful features from numerous them in a diagnostic application. We construct a deep convolution network and autoencoders-based model (AE-CDNN) in order to perform unsupervised feature learning. We use the AE-CDNN to extract the features of the available data sets, and then we use some common classifiers to classify the features. The results obtained demonstrate that the proposed AE-CDNN to perform the traditional feature extraction based classification techniques by achieving better accuracy of classification.

1 Introduction

Epilepsy is a chronic neurological disorder, and it is becoming one of the most common neurological diseases in the world. Approximately 1% of the world's population is affected by epilepsy representing more than 65 million people affected. This disorder is characterized by the occurrence of spontaneous convulsions due to the abnormal synchronous firing of the cortical neurons. This physical reaction can generate many problems for patients, including physical harm caused by the loss of consciousness, shame and discrimination.

Frequent seizures are dangerous conditions because, at the moment of disruption of the body can occur falls, fractures, burns, car accidents, and other serious physical injuries. Epilepsy can be defined as a permanent predisposition in the brain to cause epileptic seizures.

A person is diagnosed with epilepsy if they have two unprovoked seizures (or one unprovoked seizure with the likelihood of more) that were not caused by some known and reversible medical condition like alcohol withdrawal or extremely low blood sugar. Even when correctly diagnosed and treated, the epileptic patient still suffers side effects and sporadic seizures. The epileptic seizures can cause even irreversible damage to the brain, and then we can visualize the importance of analyzing epilepsy to improve the life quality and the medical treatments for these patients.

Correspondence should be addressed to Bruno Aristimunha (b.aristimunha@gmail.com)
The authors have declared that no competing interests exists.
Code is available at https://github.com/bruAristimunha/ReScience-submission.

To confirm the diagnostic, epileptologists should usually visually inspect the long-term electroencephalograms (EEG) of the scalp. EEG is a measure of the voltage fluctuation generated by the ion current of neurons in the brain, which reflects the activity of the brain's bio-electricity and may contain many physiological and disease information.
After the discovery that during a patient's seizure the brain activity changes, the EEG has become the most common epilepsy diagnostic tool. Many studies have been made, and the general problem consists in acquiring methods to classify the patients' EEG signals efficiently.
However, this costly task still presents several challenges for automatic crisis detection, among them: The scarce number of public data sets; The lack of standardization in seizure classification methodologies; The lack of standardization of data preprocessing; The cost of a specialist to label time intervals; The unbalance of the time series given the rare occurrence of the event; The difficulty of reproducing the works in the literature.
With this problem in hand, this paper reproduces the results obtained in [1], with public data labeled and preprocessed. In addition, we get new results by combining the proposal classifiers into a classifier by set voting, and we add new metrics.
The remainder of this paper is organized as follows: Section 2 presents a few works related to the classification of epileptic seizures in EGG. Section 3 introduce the methodological proposal employed, and their differences with the work of [1]. Section 4 lists the experimental validation process using epilepsy datasets. Section 5 presents the corresponding results and analyzes our approach. Finally, conclusions were summarized in Section 6.

2 Related Work

Several papers use automated methods for detecting seizures. We can extract several discriminative characteristics of the signals, among them, we mention the autocorrelation, probability of synchronization, functional connectivity network properties, EEG morphology and the reconstructed powers of the time series.
The oscillatory characteristics present in the time series of patients with epilepsy were extensively studied by temporal frequency analysis for classification [2, 3, 4, 5]. Using this technique, we mention the Discrete Wavelet Transform (DWT), which, despite requiring hand-designed parameters, it is the most used [6].
There is no standard rule for manually label seizures in databases, which makes it difficult to compare the results of these methods. Besides the few papers that use the same sets, few are looking for the same task.
In [7], it is used High Order Spectra (HOS) and spectrum-based energy resources for the automated detection of epilepsy. The proposed method yields good results when using Gaussian Mixture (GMM) for classification (93.11% and 88.78%). In [8], we have that the authors extracted the entropy of the permutation of the signals, and employed these in an SVM for classification. The result of this methodology, acquired 93.55% for our first database, in task A vs E.
Some researchers have applied Deep Belief Networks (DBNs) to the detection of seizures [9]. In the line of deep learning algorithms, Convolutional Neural Networks (CNNs) attract growing interest in the literature. In [10], they propose a CNN that learns based on the spectral information of each channel and a LSTM network with a single layer to classify the channels of the objects.
[11] propose an approach unusual when decoding each window of possible interval as an *EEG word* from the *EEG* dictionary. They explore temporal knowledge by learning context information from EEG fragments (Context-EEG). The authors obtained a 22.93% error rate in the control classification vs epileptic crisis using the second dataset present in [1].
[12] obtained 100% sensitivity with a simpler methodology than our. For each channel, it builds a autoencoder and through the error of reconstruction is classified. The dataset

was self and with no access available. [6] propose a pyramidal model of one dimension for convolution (P-1D-CNN). The method obtains 99.1 ± 0.9 in our first dataset.

3 Methodology Proposal

In this section, we describe implementation details, since the core here is the reproducible aspect of our reference article. We introduce the idea and implementation of autoencoder/feature learning and our version of the model in [1], explaining the differences we have made to the original model.
In our study, we keep the autoencoder and feature learning as proposed. However, in the classification, we no longer consider classifiers individually; these methods now make up a large ensemble learning classifier, which decides by majority vote the object class.

3.1 Implementation Details

We decided to reproduce the implementation described in the article using Keras [13] and backend in TensorFlow [14]. Our repository includes the list of all the required libraries employed in acquiring the datasets and running the model (the original and the proposed one). According to the methodology proposed in [15], we store all the checkpoints for the trained models, for reproduction purposes. Besides that, the training logs can be visualized using TensorBoard tool.
Given the lack of information about implementation in the original paper, some assumptions or cuts are made:

- The number of epoch in the AutoEnconder is assumed to be 5000;
- The number of samples per batch size is assumed to be 256;
- A column of the first database is removed, there is disagreement in the literature on the total instances, 4097 or 4096. In the specific database we use there is 4097. The removed attribute is at the endpoint of each object;
- In the second dataset, considering the information gap, we use the channel reported by the author to train the AutoEncoder;
- The loss function presented in equation 12 of the [1] was implemented and we also compared the result obtained with *MAPE*;
- The value of the seeds selected in all classifiers, data splitting and elsewhere was 42;
- The train-validation ratio was $80\% - 20\%$ to AutoEnconder, in classifiers we use cross-validation with 5 or 10-fold;
- Given a sizing problem, we resized the values using the MinMax method, before the classification process.
- The classifier presented in the final subsection (NN2) was not reproduced for lack of information;

The experiments were performed using a CPU with Intel Core i7-5930K with 3.50 GHz and two GPUs: Nvidia Quadro K5200 and GeForce GTX 970. Some experiments were also run using Nvidia Titan X.

3.2 AutoEncoders

The autoencoder implemented is a specific case of neural network structure. It is formed by three layers, the input layer, output layer and a hidden layer. The training is done to set the weights of hidden layer to force the input layer and output layer to be as close to each other as possible. Our features are extracted from the hidden layer, which reduces the dimension of data.
Therefore we have a encoding process and a decoding process, and we obtain the hidden layer h by applying the encoding function:

$$h = encoder(x) = g(W * x + b), \tag{1}$$

where W is the weight matrix between input layer and hidden layer. In the decoding function the hidden layer h is the input and $y = decoder(h)$ as output, the function is defined as follows:

$$y = decoder(x) = g(W' * x + b'), \tag{2}$$

where W' is the weight matrix between hidden layer and output layer. Since we want input and output to be as close as possible, we have the object function for the model training process:

$$\min \sum |y^{(i)} - x^{(i)}|, \tag{3}$$

where $y^{(i)}$ is the output signal and $x^{(i)}$ is the input signal.

3.3 Feature Learning Model

In this subsection, we will omit equations and minor details (for complete information, see [16, 12]. Since we have the dimension reduced by autoencoder we focus on the next challenge: how to obtain effective features from EEG signals. The AE-CDNN implemented follows the steps:

- Encoder: sample input, convolution layer, down-sampling layer, reshape operation, full connection layer, and the feature coding.
- Decoder: feature coding as input, full connection layer, reshape operation, deconvolution layer, up-sampling layer and the reconstruction samples.

Basically, the convolution layer acts as our feature extractor. It performs many successive convolution calculations of the input data and the expectation is to maintain the main components of the input data. The pooling layer is a down-sampling method which reduces data dimension. It uses windows to slide and extract the feature maps. These intervals do not overlap each other, and with then we obtain the pooled feature maps. The feature sizes tested were $m \in \{2, 4, 8, 16, 32, 64, 128, 256\}$[1].
The convolution and pooling operations can be iterated multiple times. Reshape operation uses the pooled feature maps to construct a one-dimension vector and a full-connection layer to transform this one-dimension vector.
Considering x as the input and y as the output, now we need to re-transform the one-dimension vector which will generate the y output, recall we want to minimize the difference between x and y and we have the following equation to calculate loss Mean Absolute Error:

$$\text{Loss MAE} = \frac{1}{N} \sum_{i=1}^{N} |x^{(i)} - y^{(i)}|.$$

[1] Size $m = 256$ has not been tested in [1].

In addition, given the possible interpretations in the original text, we have also used/implemented two loss functions, namely Mean Absolute Percentage Error - MAPE and Mean Absolute Average Error - MAAE [2], that are contained below:

$$\text{Loss MAPE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|x^{(i)} - y^{(i)}|}{x^{(i)}}.$$

The difference between the loss functions is only in the fact that one takes in the denominator the value per $x^{(i)}$ and the other takes the average $\bar{x}^{(i)}$.

$$\text{Loss MAAE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|x^{(i)} - y^{(i)}|}{\bar{x}^{(i)}}.$$

3.4 Classification

Since we have extracted the features with reduced dimension, we use supervised learning models on these features in order to classify the EEG signals. We evaluate each classifier and then we compare the results obtained with each one. The classical classifiers used are: K-Nearest Neighbors (K-NN), Support-Vector Machine - Linear Kernel and Radial Basis Kernel (SVM1, SVM2), Decision Tree (DT), Random Forest (RT), Multilayer Neural Network (MLP), Adaptive Boosting Algorithm (ADB) and Gaussian Naive Bayesian (GNB).
The proposed modification combines these classifiers and creates a single classifier that decides by voting. In short, the classifiers were combined by ensemble learning, and the result of the classification became the classification most voted by the classifiers.

4 Experimental Methodology

In this paper, as in our reference paper [1], we use unsupervised learning method in EEG signals in order to obtain useful features. This process is needed because the original data is high-dimensional. By using the auto-encoder, we can extract features with reduced dimension.

4.1 Bonn University EEG database

We can use different approaches to detect epileptic crisis. Then, to acquire a comparative measure, we verify our outputs using the method described in 3 and the original one showed in [1]. This database is public and was published by [17]. The study groups were the control, inter-ictal and ictal distributed into five sets (denotated A-E). Each containing 100 records of 23.6 seconds duration and frequency of 173.6 Hz on a single channel, with 12-bit resolution. Each data segment has 4097 samples. These recordings underwent a pre-processing in which the signals had a band filter between 0.53 to 40 Hz. There was also the removal of artifacts such as muscle movements or flicker movements.
Using labels A, B, C, D and E for the subsets, we have that A and B contain records of 5 healthy volunteers. Set A corresponds to open-eye activity and subset B to closed-eye activity. The subsets C and D have signals during the absence (interictal epileptiform activity) of 5 epileptic patients. And E records the signals during epileptic patients' seizure (ictal intervals). According to [18], this dataset is a compilation of recordings under different conditions.

[2] The formula presented in the original article by [1] differs from the MAPE formula, despite having similar intuitions. Thus, we chose to implement this loss equation, and we have not found its use elsewhere.

4.2 Children's Hospital of Boston EEG database

The second database, also public, contains the EEG signals from a Childrenѝ Hospital of Boston [16]. It was recorded by measuring the brain's electrical activity to obtain EEG signals by connecting multiple electrodes to the patients' scalp. The data incorporates the EEG signals of 23 children with refractory epilepsy.
This database, built in partnership with the Massachusetts Institute of Technology (MIT), has 5 men and 18 women between 3 and 22 years. The frequency range was 256 Hz with 16 resolution bits. Most patients contain 23 channels and some with 24 channels. In contrast to the first set of data, we have multiple channels here, then we need to select channels. The selection followed the methodology used in [19], which analyzes the variance of each patient, and after that, chooses the channel of greater variance to represent that individual. The channel reported by the authors was FT9-FT10.
In the data of the first ten patients, 200 windows of the same size of the control set were chosen from the epileptic patients we choose 200, with size of 4096, in the same way of the control group.

4.3 Performance Measures

According to [20], most of the state-of-the-art systems for epilepsy use the metrics defined below. The adaptation of these metrics for evaluating our system contributes to fair comparison with state-of-the-art systems. The definitions of these metrics are given in Table 1.

Acurracy	Precision	Specificity	Sensitivity	F-Measure
$\frac{TP+TN}{TP+TN+FP+FN}$	$\frac{TP}{TP+FP}$	$\frac{TN}{TN+FP}$	$\frac{TP}{FN+TP}$	$\frac{2 \cdot Precision \cdot Sensitivity}{Sensitivity+Precision}$

Table 1. Metrics and Definition use in our paper. Only the Acurracy was considered in [1].

where False Negatives - FN is the number of epileptic cases, which are predicted as control, True Positives - TP is the number of epileptic cases, which are predicted as epileptic, True Negative - TN is the number of control case that is predicted as control and False Positives - FP is the number of control cases that are identified as epileptic by the system. In addition, there was also the AUC-ROC (Area Under The Curve - Receiver Operating Characteristic) defined as the cumulative distribution function of the true positive rate vs the false-negative rate denoted by a threshold.

5 Results and Discussion

In this section we analyze three analytical approaches, a first subsection an analysis of the variance present in the channels. The second being a sub-section with the reproduction of all possible tables and figures, with a discussion of the reasons for the differences. In the third sub-section, we present an extension of the results, evaluating other classification metrics, proposing a new classifier and varying the classifier parameters.

5.1 Checking the Variance

According to the original authors, the choice of the channel in the second dataset observed the variance present in the channels. For that, they followed the methodology: "1) calculate the variance of each channel in each sample, and select the channel with the maximum variance for each sample; 2) count these channels."
Thereby, we model the three scenarios for the interpretation of what is the sample defined by the author. In the first, we analyzed each recording file of the dataset as a sample, having an average length of 921600 referring to the recorded $3600s$. In these

files, we compute and list the electrode with more variance and discard the rest. We accumulate and count for all files. The results obtained can be seen in the Figure below:

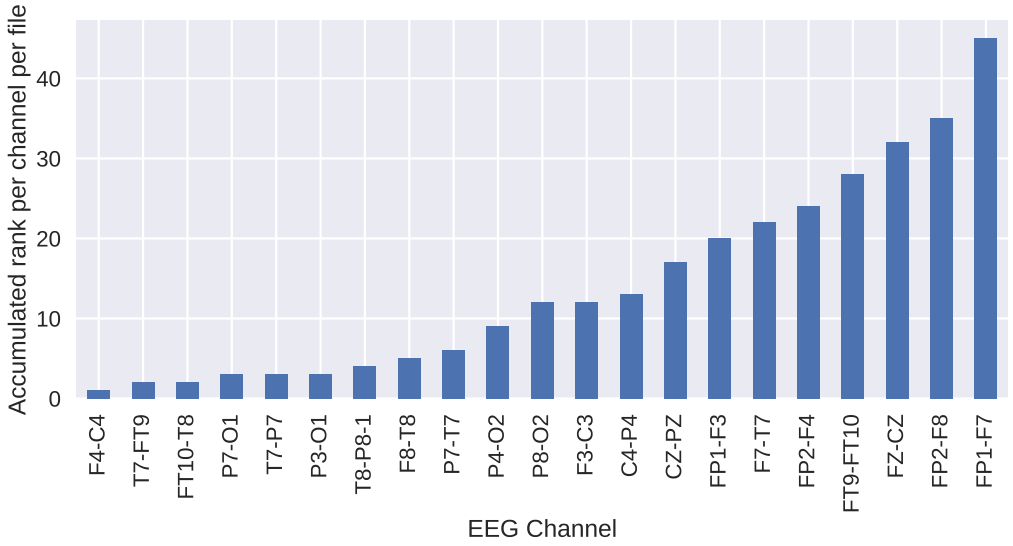

Figure 1. Accumulated variance per sample, considering a sample as each recording file.

In the second scenario, we understand that each sample is accumulated per person with all his recordings. So the variance was calculated in parallel in the files and combined for each person. For each person, we count the occurrence of the channel with more variance. As shown in Figure 2.

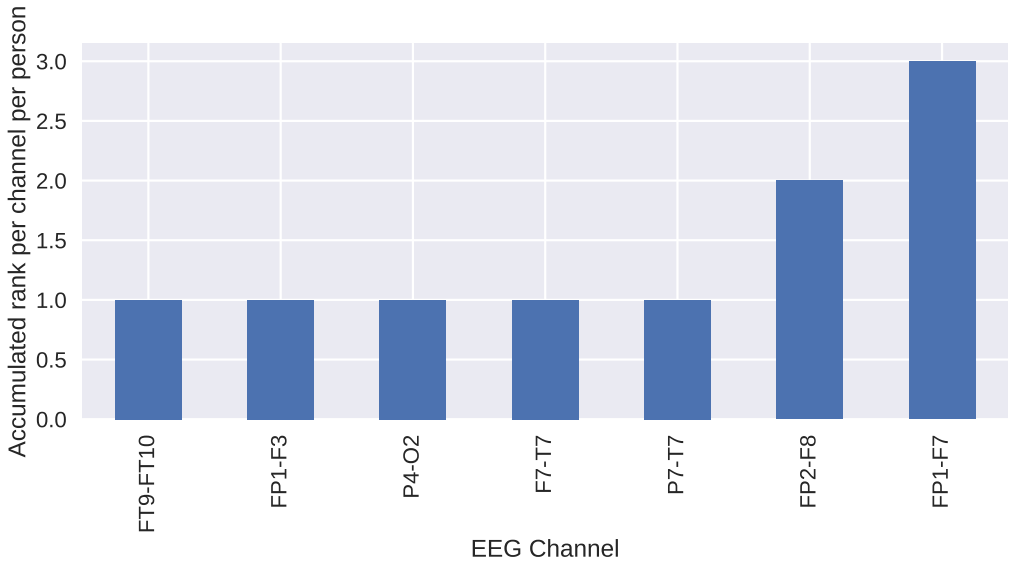

Figure 2. Accumulated variance per sample, considering a sample as being all the recordings of each person.

Finally, as a final scenario, we calculate the cumulative variance across all people and all records, thus, we did not perform a sampling process. In other words, we put all the files together and calculate the variance as if it were a single record. For this, we compute the variance, number of points and average per channel in each file and accumulate through the cumulative variance calculation algorithm. The result of this analysis approach can

be seen in Figure 3.

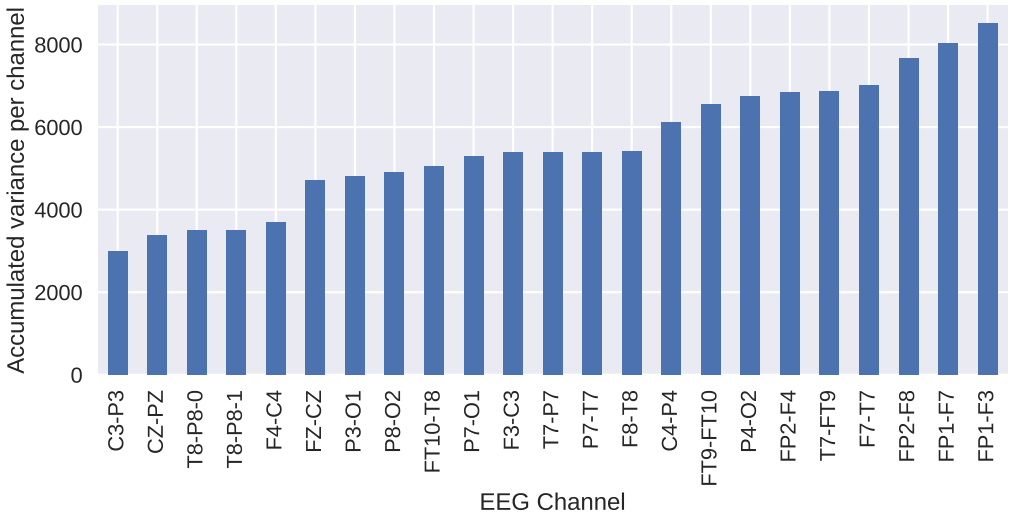

Figure 3. Total accumulated variance in the first ten people of the dataset, as granular as possible.

The results obtained in the first, second and three scenario were not consistent with those reported by the author. Thus, not in any of the scenarios analyzed did we obtain the same results in the variance as the original authors. There is still another possible scenario, however, not reproducible, being the possibility that the authors randomly sampled the dataset and in this they verified the variance.
However, given the associated uncertainty, we decided to repeat the choice of the channel $FT9 - FT10$, although this is not the one with the most variance in the modeled scenarios.

5.2 Reproduction of the values reported by the original author

The results obtained in our reproduction experiment, for the first dataset, are presented in Accuracy Tables 2 and 3 and the differences between results can be seen in Figures 4 and 5. We employed the same methodology as the one used in the original paper, performing a 5-fold cross-validation for each classifier, and we show the mean values. For each table reproduced, we also present the original result and the difference between them.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian??_nb	average
2	0.702	0.600	0.600	0.712	0.700	0.600	0.710	0.562	0.64825
4	0.674	0.600	0.600	0.700	0.678	0.600	0.758	0.562	0.64650
8	0.480	0.600	0.600	0.600	0.600	0.600	0.600	0.400	0.56000
16	0.802	0.600	0.600	0.762	0.812	0.634	0.784	0.636	0.70375
32	0.794	0.600	0.606	0.758	0.830	0.666	0.790	0.638	0.71025
64	0.772	0.714	0.754	0.682	0.786	0.792	0.808	0.618	0.74075
128	0.800	0.662	0.726	0.756	0.776	0.742	0.780	0.608	0.73125

Table 2. Accuracy values obtained in reproduction, AE-CDNN-MAE for Dataset 1.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	average
2	0.722	0.6	0.600	0.718	0.690	0.600	0.750	0.560	0.65500
4	0.480	0.6	0.600	0.600	0.600	0.600	0.600	0.400	0.56000
8	0.776	0.6	0.600	0.708	0.784	0.600	0.744	0.574	0.67325
16	0.804	0.6	0.600	0.678	0.788	0.596	0.760	0.610	0.67950
32	0.740	0.6	0.600	0.734	0.744	0.600	0.754	0.586	0.66975
64	0.800	0.6	0.600	0.774	0.820	0.658	0.772	0.670	0.71175
128	0.784	0.6	0.624	0.738	0.818	0.700	0.812	0.640	0.71450

Table 3. Accuracy values obtained in reproduction, AE-CDNN-MAAE for Dataset 1.

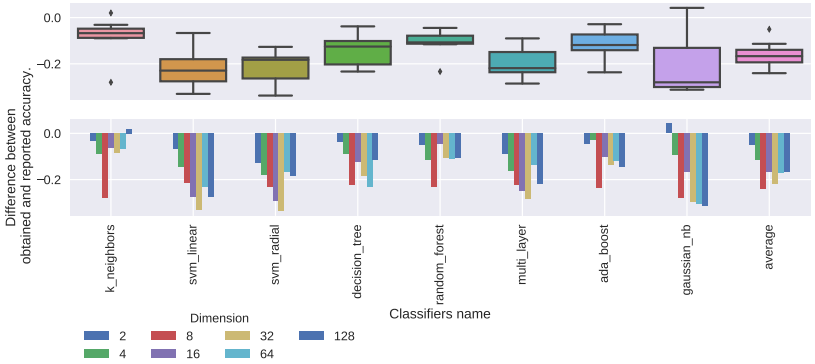

Figure 4. Classification Accuracy Results of AE-CDNN-MAE for Dataset 1 [1], Reproduced Original and Difference.

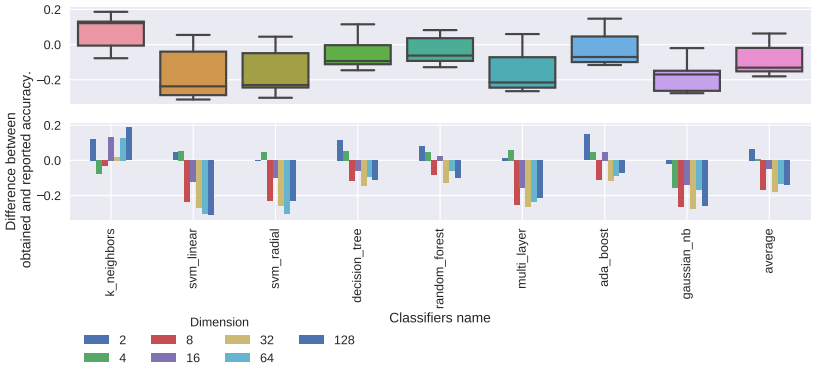

Figure 5. Classification Accuracy Results of AE-CDNN-MAAE for Dataset 1 [1], Reproduced Original and Difference.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetuer tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.
Tables 4 and 5 Figures 6 and 7

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	average
2	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.500000
4	0.7125	0.5000	0.5525	0.7300	0.6575	0.4725	0.7075	0.7525	0.635625
8	0.8025	0.5325	0.6325	0.8075	0.7925	0.7050	0.8000	0.7025	0.721875
16	0.8150	0.6500	0.7575	0.8175	0.8350	0.7725	0.8175	0.7800	0.780625
32	0.7400	0.8575	0.8625	0.8225	0.8550	0.8725	0.8475	0.8675	0.840625
64	0.7200	0.8325	0.8450	0.7625	0.8350	0.8525	0.8150	0.8650	0.815937
128	0.6925	0.8300	0.8300	0.7450	0.8025	0.8275	0.7850	0.8225	0.791875

Table 4. Accuracy values obtained in reproduction, AE-CDNN-MAE for Dataset 2.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	average
2	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.500000
4	0.7200	0.5000	0.5575	0.6825	0.6725	0.5500	0.7000	0.7525	0.641875
8	0.8400	0.6975	0.8025	0.8150	0.8450	0.7825	0.8550	0.8375	0.809375
16	0.7900	0.7475	0.8250	0.7725	0.8250	0.8200	0.7950	0.8525	0.803438
32	0.8125	0.7450	0.8125	0.8100	0.8625	0.7950	0.8225	0.8675	0.815937
64	0.7125	0.8150	0.8475	0.7550	0.8225	0.8300	0.8250	0.8525	0.807500
128	0.6950	0.8400	0.8375	0.7700	0.8325	0.8275	0.8100	0.8500	0.807813

Table 5. Accuracy values obtained in reproduction, AE-CDNN-MAAE for Dataset 2.

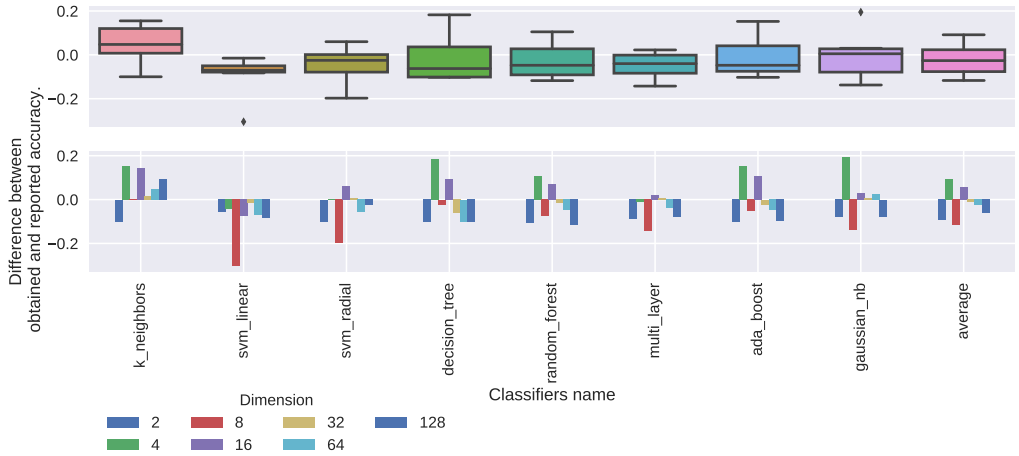

Figure 6. Classification Accuracy Results of AE-CDNN-MAE for Dataset 2 [1], Reproduced Original and Difference.

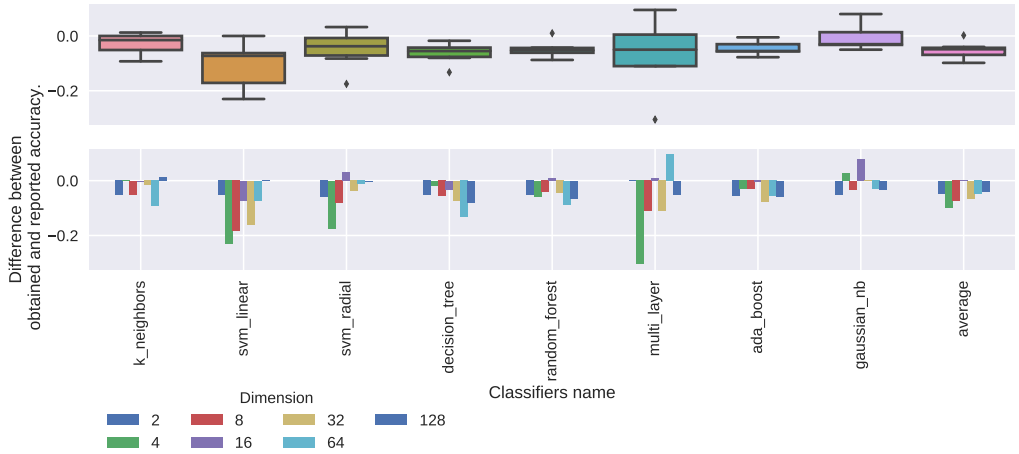

Figure 7. Classification Accuracy Results of AE-CDNN-MAAE for Dataset 2 [1], Reproduced Original and Difference.

Considering Dataset 2 and Tables we acquired similar results when compared with the results obtained by the original authors [1]. In general, the original paper acquired a maximum accuracy greater than those obtained by our reproduction implementation, but the average and unique values per dimension are close in most cases considering both functions AE-CDNN-MAE and AE-CDNN-MAAE. However, for Dataset 1 the accuracy values obtained in this paper are significantly lower than the ones obtained by the original paper considering both AE-CDNN-MAE and AE-CDNN-MAAE, as shown in Figure 8.

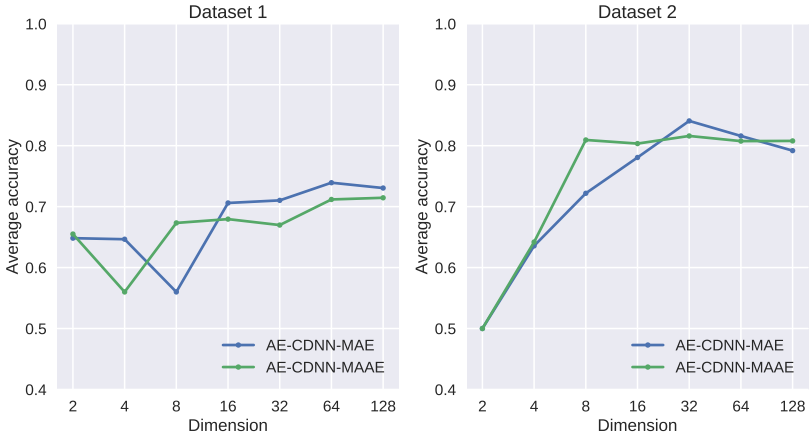

Figure 8. Average Accuracy Results of AE-CDNN-MAE and AE-CDNN-MAAE, with different dimension values in the two dataset.

When analyzing the behavior of the different loss functions, we see that the MAAE function does not always obtain superior results than the MAE function. The same is observed in the original article, as well as a similar behavior, but the average accuracy obtained are significantly higher than those obtained by our reproduction. Similarly, when we analyze the loss function MAAE and MAPE, in Figure 9, we have that the behavior of both is not very divergent, being MAAE generating a higher accuracy in the first dataset. In the second dataset, MAPE has a more stable behavior and generates greater accuracy.

Figure 9. Average Accuracy Results of AE-CDNN-MAAE and AE-CDNN-MAPE in the two dataset.

When observing the values obtained in the classification, in the k-fold, we have that the accuracy values follow the proportion of the data, indicating the non-learning of the classification methods. We observed below the result for the accuracy inspection, for the cross validation, for $m = 2$. Analyzing the accuracy obtained by classifiers in Tables 6 and 7 we observe the values obtained by AE-CDNN-MAAE and AE-CDNN-MAE are close, however the function AE-CDNN-MAAE acquired smoothly better results and with less variation, in general.

5-fold	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb
1	0.66	0.6	0.6	0.71	0.61	0.6	0.68	0.51
2	0.70	0.6	0.6	0.65	0.72	0.6	0.70	0.66
3	0.68	0.6	0.6	0.73	0.75	0.6	0.73	0.54
4	0.74	0.6	0.6	0.76	0.74	0.6	0.74	0.56
5	0.73	0.6	0.6	0.71	0.68	0.6	0.70	0.54

Table 6. Accuracy in Classification, with loss MAE, on each fold cross-validation, for Dataset 1.

5-fold	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb
1	0.65	0.6	0.6	0.66	0.64	0.6	0.73	0.53
2	0.72	0.6	0.6	0.72	0.64	0.6	0.72	0.64
3	0.76	0.6	0.6	0.72	0.72	0.6	0.77	0.54
4	0.70	0.6	0.6	0.73	0.70	0.6	0.75	0.53
5	0.78	0.6	0.6	0.76	0.75	0.6	0.78	0.56

Table 7. Accuracy in Classification, with loss MAAE, on each fold cross-validation, for Dataset 1.

In the original paper we observe similar differences between the two functions, the results for AE-CDNN-MAAE are smoothly better for most classifiers, but considering **gaussian_nb**, for example, the function AE-CDNN-MAAE acquired much better results comparing with AE-CDNN-MAE. Although the results in original paper also have few variations for the classifiers **svm_linear**, **svm_radial** and **multi_layer** we had no variance in these classifers for function AE-CDNN-MAAE.
In the second dataset, in Tables 8 and 9, when analyzing by fold we have that results are worse than those reported by the authors. However, the results is consistent with the hypothesis during the process, there was no feature learning. Also given the balance of this second dataset, we have that all methods do not present a better result than the random chance.

5-fold	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb
1	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
2	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
3	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
4	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

Table 8. Accuracy in Classification, with loss MAE, on each fold cross-validation, for Dataset 2.

5-fold	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb
1	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
2	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
3	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
4	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

Table 9. Accuracy in Classification, with loss MAAE, on each fold cross-validation, for Dataset 2.

When analyzing the reduced values by class, specifically with $m = 4$ we have 3 of the 4 attributes are 0, in the best scenario, indicating that there was no learning in Auto Encoder to distinguish the behavior by class. This bad representation of latent space occurs regardless of the loss function.

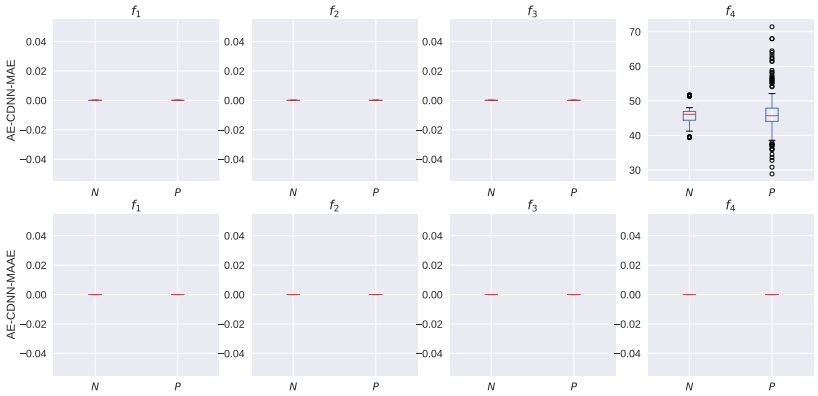

Figure 10. Feature Distribution of AE-CDNN-MAE and AE-CDNN-MAAE, with $m = 4$, in the first dataset.

When we analyze the behavior of the loss functions at the epoch, in Figure 11, in the first dataset, we have that these do not have a parallel with those reported by the original author. In addition, numerically in the second function MAAE the values also do not present an adequate dimension with that originally reported. Consequently, we also have an indication that in the MAAE function there was not an adequate generalization in the validation.

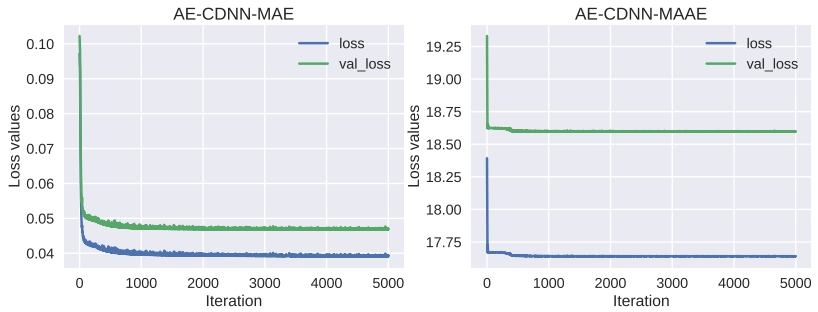

Figure 11. Change of loss function of AE-CDNN-MAE and AE-CDNN-MAAE, in the first dataset, with $m = 4$.

Even assuming that the author used the MAPE loss function, we still do not obtain an adequate result in loss at the epoch, as show the Figure 12.

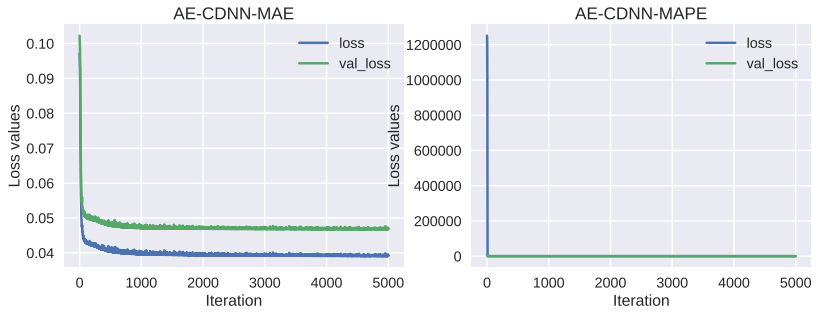

Figure 12. Change of loss function of AE-CDNN-MAE and AE-CDNN-MAPE, in the first dataset, with $m = 4$.

These differences also occur in the establishment in the baseline methods, indicating

that there is some cut in the training set that was not included in this modeling, given the lack of information in the article. In Figure 13 we observe similar average accuracy between AE-CDNN-MAE, AE-CDNN-MAAE, PCA and SRP for both datasets.

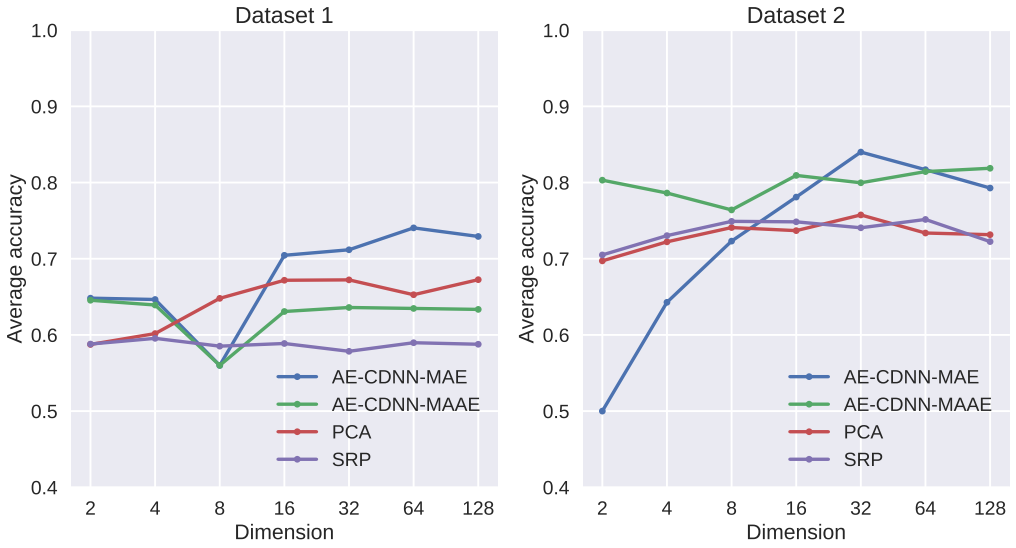

Figure 13. Comparison of accuracy for different loss functions (AE-CDNN-MAE, AE-CDNN-MAAE), and also with baseline (PCA, SRP).

The same is observed in the original article, as well as a similar behavior, but the average accuracy obtained for Dataset 1 are significantly higher than those obtained by our reproduction, as shown in the Table 10:

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	average
16	0.800	0.6	0.600	0.766	0.820	0.652	0.778	0.632	0.70600
32	0.802	0.6	0.602	0.756	0.832	0.674	0.802	0.638	0.71325

Table 10. Accuracy in Classification, in the first dataset with $CV = 10$.

When we analyze the result assuming a 10-fold, we have an increase in the accuracy values for the first dataset, however, still below that reported by the author.

5.3 Extension of the values reported by the original author

In Precision Tables 11, 12, 13 and 14, we realize that one of the most precise and specific method was the Gaussian Naive Bayesian; however, when analyzing the behavior in the Sensitivity metric, we do not have a satisfactory result. This indicates that the method pinpoints true negatives rather than true positives. If treated from a medical field, this result is worrying. The cases that the method indicates are true positives; however, this method misses many cases.

We also analyze that we cannot consider Support Vector Machine (Linear and Radial) or Multi-Layer results with the lowest m. The result in specificity indicates that the method behaves unwanted, possibly indicating all values as true positives. This rule burdens the medical system because further detection of the seizure requires further investigation for a complete diagnosis of the disease.

By our method, we note that the three metrics indicate that a progression in the number of features generates an improvement in seizure detection. Similar behavior is observed in Multi-Layer, only for Accuracy and Sensitivity, in this case, a beneficial behavior for the application. No trend was observed in the other methods. The average of the methods does not exceed our Ensemble method in almost any scenario.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	ensemble	average
2	0.757324	0.600000	0.600000	0.741488	0.761257	0.600000	0.750363	0.730169	0.719986	0.695621
4	0.737413	0.600000	0.600000	0.757120	0.748638	0.600000	0.799839	0.733892	0.736059	0.701440
8	0.240000	0.600000	0.600000	0.600000	0.600000	0.600000	0.600000	0.000000	0.600000	0.493333
16	0.886412	0.600000	0.600000	0.856665	0.875097	0.660251	0.830206	0.992857	0.857434	0.795436
32	0.881359	0.600000	0.616173	0.888903	0.899581	0.724420	0.849551	0.985926	0.906119	0.816892
64	0.835445	0.875415	0.929524	0.825899	0.846616	0.864681	0.840069	0.966667	0.951357	0.881741
128	0.882557	0.751233	0.899303	0.861868	0.854048	0.832873	0.817969	0.975096	0.923408	0.866484
256	0.773418	0.823845	0.855714	0.769434	0.794883	0.855478	0.838311	0.990000	0.923576	0.847185

Table 11. Precision values obtained by the same methodology - First Dataset with MAE.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	ensemble	average
2	0.770458	0.600000	0.600000	0.750852	0.746194	0.600000	0.822911	0.733094	0.748318	0.707981
4	0.240000	0.600000	0.600000	0.600000	0.600000	0.600000	0.600000	0.000000	0.600000	0.493333
8	0.869678	0.600000	0.600000	0.714619	0.854352	0.600000	0.788958	1.000000	0.787597	0.757245
16	0.882438	0.600000	0.600000	0.806733	0.848472	0.598384	0.813369	0.938229	0.816352	0.767109
32	0.802725	0.600000	0.600000	0.762479	0.807989	0.600000	0.795577	1.000000	0.795239	0.751557
64	0.882736	0.600000	0.600000	0.894553	0.879284	0.673461	0.839122	0.978744	0.912343	0.806694
128	0.870097	0.600000	0.698276	0.839078	0.890596	0.801880	0.847304	0.954335	0.907713	0.823253
256	0.791896	0.798756	0.886401	0.792746	0.790266	0.804737	0.816419	0.897697	0.922234	0.833461

Table 12. Precision values obtained by the same methodology - First Dataset with MAAE.

In the first dataset, when we analyze the accuracy we have that the naive bayes Gaussian classifier presents a drop of (40%, 23%, for first and second loss respectively) if compared to the accuracy. The average difference, in precision minus accuracy, is 6%, indicating that the precision metric achieves slightly higher results on average in the samples. The k_neighbors classifier is the classifier, in the first loss function, that has the least average difference in results, while the svm_linear method shows the same result for the second loss set. At the other end, we have the largest variation in both gaussian_nb data sets.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	ensemble	average
2	0.400000	0.500000	0.500000	0.000000	0.000000	0.200000	0.000000	0.000000	0.000000	0.177778
4	0.733781	0.000000	0.800000	0.788375	0.670690	0.441892	0.760461	0.901095	0.857197	0.661499
8	0.823843	0.800000	0.980000	0.853339	0.817331	0.971852	0.841619	0.987500	0.988235	0.895969
16	0.857868	1.000000	0.979144	0.862932	0.869046	0.965722	0.843127	0.965152	0.974609	0.924178
32	0.943593	0.947251	0.927162	0.848473	0.861083	0.915744	0.852046	0.917396	0.938637	0.905709
64	0.916540	0.892930	0.902664	0.803868	0.847252	0.893022	0.842228	0.948086	0.918116	0.884967
128	0.915294	0.895945	0.852936	0.762151	0.809500	0.859518	0.788800	0.939528	0.903233	0.858545
256	0.930307	0.863353	0.869493	0.768955	0.807719	0.840146	0.792065	0.958701	0.879169	0.856657

Table 13. Precision values obtained by the same methodology - Second Dataset with MAE.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	ensemble	average
2	0.400000	0.500000	0.500000	0.000000	0.000000	0.300000	0.000000	0.000000	0.000000	0.188889
4	0.744177	0.000000	1.000000	0.739486	0.683738	0.787324	0.763358	0.913765	0.882316	0.723796
8	0.877945	1.000000	0.987879	0.849861	0.871662	0.942000	0.872110	0.917184	0.925018	0.915962
16	0.920560	0.954474	0.904111	0.810101	0.841281	0.903250	0.817114	0.882190	0.914344	0.883047
32	0.919487	0.983333	0.943688	0.844350	0.887551	0.910233	0.834831	0.917046	0.933478	0.908222
64	0.921607	0.888863	0.878184	0.783616	0.822289	0.866394	0.846359	0.871348	0.894940	0.863733
128	0.873810	0.873781	0.858687	0.817420	0.833871	0.865899	0.830227	0.879029	0.871860	0.856065
256	0.938120	0.848311	0.847865	0.745542	0.791865	0.832058	0.801048	0.916435	0.866233	0.843053

Table 14. Precision values obtained by the same methodology - Second Dataset with MAAE.

Meanwhile, in the second set of data, generated by the two loss functions, the difference between precision and accuracy is greater in the smallest dimensions, while the values are more stable, and close to the accuracy values in the largest dimensions. Such stability behavior is also observed in the absolute values.

Specificity and Sensitivity — When we analyze the specificity, we have that the SVM method, with different kernels, cannot obtain a separation of the hyperspaces of the attributes to distinguish the non-schizoid events. In this way, we have that the classifier cannot distinguish when the person is without epileptic attack. From a medical point of view, there are not so many implications for this, since the weighting of importance is inclined to detect true positives. The SVM sensitivity for these cases, in high dimensions (above 32) presents reasonable values, approximately 70% in the worst scenarios. In general, the panorama of the accumulated sensitivity indicates that the worst classifiers, regardless of the number of dimensions, are the Gaussian naive bayes, and the K-neighbors

for high dimensions. The ensemble classifier has average cumulative sensitivity (71% in the worst case scenario), with the exception of lower case scenarios 2.
The performance of the Gaussian classifier may be related to the fact that the inputs are highly dependent on each other, thus violating the method's premise of independence. In the case of the k-neighbors classifier, given the presence of the values 0 in various dimensions, as shown previously, which can affect the distance assumptions necessary for the method.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	ensemble	average
2	0.645	0.000	0.000	0.580	0.660	0.000	0.610	0.765	0.515	0.419444
4	0.620	0.000	0.000	0.635	0.645	0.000	0.700	0.770	0.560	0.436667
8	0.600	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.177778
16	0.850	0.000	0.000	0.800	0.815	0.345	0.745	0.995	0.780	0.592222
32	0.845	0.000	0.110	0.870	0.865	0.605	0.785	0.990	0.880	0.661111
64	0.770	0.865	0.930	0.805	0.785	0.820	0.760	0.980	0.950	0.851667
128	0.845	0.635	0.895	0.825	0.805	0.775	0.725	0.985	0.915	0.822778
256	0.640	0.745	0.800	0.690	0.695	0.780	0.755	0.995	0.905	0.778333

Table 15. Specificity values obtained by the same methodology - First Dataset with MAE.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	ensemble	average
2	0.660	0.000	0.00	0.590	0.630	0.000	0.760	0.770	0.590	0.444444
4	0.600	0.000	0.00	0.000	0.000	0.000	0.000	1.000	0.000	0.177778
8	0.835	0.000	0.00	0.475	0.800	0.000	0.685	1.000	0.645	0.493333
16	0.840	0.000	0.00	0.755	0.780	0.000	0.715	0.965	0.705	0.528889
32	0.715	0.000	0.00	0.620	0.730	0.000	0.685	1.000	0.660	0.490000
64	0.845	0.000	0.00	0.875	0.830	0.405	0.780	0.985	0.880	0.622222
128	0.830	0.000	0.42	0.795	0.855	0.750	0.775	0.970	0.890	0.698333
256	0.700	0.725	0.87	0.750	0.700	0.720	0.725	0.930	0.915	0.781667

Table 16. Specificity values obtained by the same methodology - First Dataset with MAAE.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	ensemble	average
2	0.200	0.000	0.000	1.000	1.000	0.600	1.000	1.000	1.000	0.644444
4	0.760	1.000	1.000	0.830	0.705	0.390	0.795	0.940	0.905	0.813889
8	0.825	1.000	0.995	0.855	0.820	0.985	0.850	0.995	0.995	0.924444
16	0.870	1.000	0.990	0.865	0.865	0.980	0.850	0.980	0.985	0.931667
32	0.965	0.960	0.930	0.865	0.860	0.925	0.850	0.930	0.950	0.915000
64	0.945	0.910	0.910	0.825	0.850	0.905	0.850	0.950	0.930	0.897222
128	0.970	0.915	0.865	0.770	0.815	0.870	0.795	0.960	0.925	0.876111
256	0.970	0.880	0.875	0.790	0.825	0.845	0.805	0.970	0.905	0.873889

Table 17. Specificity values obtained by the same methodology - Second Dataset with MAE.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	ensemble	average
2	0.200	0.000	0.000	1.000	1.000	0.400	1.000	1.000	1.000	0.622222
4	0.770	1.000	1.000	0.790	0.700	0.600	0.825	0.950	0.930	0.840556
8	0.890	1.000	0.990	0.865	0.875	0.970	0.880	0.930	0.945	0.927222
16	0.955	0.980	0.930	0.830	0.850	0.925	0.835	0.895	0.935	0.903889
32	0.945	0.990	0.960	0.855	0.895	0.935	0.845	0.930	0.955	0.923333
64	0.970	0.910	0.890	0.810	0.820	0.880	0.850	0.880	0.910	0.880000
128	0.960	0.885	0.870	0.835	0.835	0.880	0.835	0.890	0.890	0.875556
256	0.965	0.875	0.855	0.750	0.790	0.840	0.805	0.940	0.880	0.855556

Table 18. Specificity values obtained by the same methodology - Second Dataset with MAAE.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	ensemble	average
2	0.740000	1.000000	1.000000	0.800000	0.726667	1.000000	0.776667	0.426667	0.830000	0.811111
4	0.710000	1.000000	1.000000	0.743333	0.700000	1.000000	0.796667	0.423333	0.813333	0.798519
8	0.400000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	1.000000	0.822222
16	0.770000	1.000000	1.000000	0.753333	0.810000	0.840000	0.810000	0.396667	0.830000	0.801111
32	0.760000	1.000000	0.936667	0.686667	0.806667	0.690000	0.793333	0.403333	0.763333	0.760000
64	0.773333	0.613333	0.636667	0.593333	0.786667	0.760000	0.840000	0.376667	0.676667	0.672963
128	0.770000	0.680000	0.613333	0.716667	0.756667	0.730000	0.816667	0.356667	0.673333	0.679259
256	0.790000	0.793333	0.783333	0.690000	0.783333	0.856667	0.836667	0.380000	0.766667	0.742222

Table 19. Sensitivity values obtained by the same methodology - First Dataset with MAE.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	ensemble	average
2	0.763333	1.00	1.000000	0.803333	0.730000	1.000000	0.743333	0.420000	0.806667	0.807407
4	0.400000	1.00	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	1.000000	0.822222
8	0.736667	1.00	1.000000	0.876667	0.773333	1.000000	0.783333	0.290000	0.876667	0.815185
16	0.780000	1.00	1.000000	0.626667	0.793333	0.993333	0.790000	0.373333	0.850000	0.800741
32	0.756667	1.00	1.000000	0.810000	0.753333	1.000000	0.800000	0.310000	0.873333	0.811481
64	0.770000	1.00	1.000000	0.706667	0.813333	0.810000	0.766667	0.460000	0.823333	0.794444
128	0.753333	1.00	0.760000	0.706667	0.793333	0.676667	0.836667	0.420000	0.730000	0.741852
256	0.743333	0.69	0.653333	0.596667	0.753333	0.753333	0.803333	0.403333	0.673333	0.674444

Table 20. Sensitivity values obtained by the same methodology - First Dataset with MAAE.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	ensemble	average
2	0.800	1.000	1.000	0.000	0.000	0.400	0.000	0.000	0.000	0.355556
4	0.665	0.000	0.105	0.630	0.610	0.560	0.620	0.565	0.565	0.480000
8	0.780	0.065	0.270	0.770	0.765	0.425	0.750	0.410	0.470	0.522778
16	0.760	0.300	0.525	0.780	0.805	0.565	0.785	0.580	0.615	0.635000
32	0.515	0.755	0.795	0.765	0.850	0.820	0.845	0.805	0.810	0.773333
64	0.495	0.755	0.780	0.710	0.820	0.800	0.780	0.780	0.760	0.742222
128	0.415	0.745	0.795	0.725	0.790	0.805	0.775	0.685	0.740	0.719444
256	0.320	0.770	0.825	0.675	0.740	0.810	0.745	0.570	0.710	0.685000

Table 21. Sensitivity values obtained by the same methodology - Second Dataset with MAE.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	ensemble	average
2	0.800	1.000	1.000	0.000	0.000	0.600	0.000	0.000	0.000	0.377778
4	0.670	0.000	0.115	0.575	0.645	0.410	0.575	0.555	0.505	0.450000
8	0.790	0.395	0.615	0.765	0.815	0.585	0.830	0.745	0.710	0.694444
16	0.625	0.515	0.720	0.720	0.800	0.700	0.755	0.810	0.725	0.707778
32	0.680	0.500	0.665	0.775	0.830	0.670	0.800	0.805	0.705	0.714444
64	0.455	0.720	0.805	0.690	0.825	0.780	0.800	0.825	0.780	0.742222
128	0.430	0.795	0.805	0.725	0.830	0.780	0.785	0.810	0.765	0.747222
256	0.380	0.715	0.825	0.730	0.800	0.800	0.780	0.735	0.790	0.728333

Table 22. Sensitivity values obtained by the same methodology - Second Dataset with MAAE.

F-measure and ROC-AUC — When analyzing the behavior of the methods for the reason of the metrics of $F - measure$ and $ROC - AUC$ we have that in the first data set, the SVM and multi_layer methods present the best results. At the other end, we have the appearance with the naive bayes and k neighbor methods.
We analyze the behavior of the F-measure. The relationship between sensitivity and precision is captured by this measure, as the gaussian_nb, svm_linear, k_neighbors methods did not obtain good results, which is coherent with the accuracy result. The measures generally show close results with each other.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	ensemble	average
2	0.747682	0.750000	0.750000	0.768802	0.741828	0.750000	0.761796	0.538058	0.770530	0.730966
4	0.721623	0.750000	0.750000	0.748852	0.721429	0.750000	0.797900	0.536233	0.772368	0.727600
8	0.300000	0.750000	0.750000	0.750000	0.750000	0.750000	0.750000	0.000000	0.750000	0.616667
16	0.823806	0.750000	0.750000	0.797643	0.837828	0.735524	0.818306	0.563238	0.840096	0.768493
32	0.815433	0.750000	0.739445	0.773996	0.850141	0.705493	0.819139	0.568770	0.827045	0.761051
64	0.802133	0.711397	0.753345	0.686008	0.815012	0.807617	0.839822	0.539611	0.788601	0.749283
128	0.821429	0.700987	0.726700	0.781170	0.800827	0.775028	0.815637	0.518483	0.775147	0.746156
256	0.778737	0.808020	0.817393	0.723714	0.788024	0.854977	0.836741	0.547427	0.837520	0.776950

Table 23. F-measure values obtained by the same methodology - First Dataset with MAE.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	ensemble	average
2	0.766331	0.750000	0.750000	0.773725	0.737374	0.750000	0.780378	0.533160	0.775831	0.735200
4	0.300000	0.750000	0.750000	0.750000	0.750000	0.750000	0.750000	0.000000	0.750000	0.616667
8	0.796843	0.750000	0.750000	0.787354	0.810166	0.750000	0.784228	0.444679	0.829237	0.744723
16	0.827147	0.750000	0.750000	0.695623	0.818202	0.746855	0.798862	0.532933	0.831395	0.750113
32	0.777647	0.750000	0.750000	0.782385	0.778931	0.750000	0.795370	0.469463	0.831627	0.742825
64	0.822004	0.750000	0.750000	0.788701	0.844298	0.732084	0.800982	0.623790	0.864900	0.775195
128	0.807123	0.750000	0.695139	0.766585	0.836497	0.731669	0.841509	0.580532	0.805112	0.757130
256	0.764444	0.732545	0.748939	0.673721	0.770672	0.776770	0.808763	0.554930	0.775912	0.734077

Table 24. F-measure values obtained by the same methodology - First Dataset with MAAE.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	ensemble	average
2	0.533333	0.666667	0.666667	0.000000	0.000000	0.266667	0.000000	0.000000	0.000000	0.237037
4	0.696765	0.000000	0.161705	0.697813	0.637313	0.444939	0.675181	0.693245	0.677200	0.520462
8	0.800147	0.117744	0.420953	0.804906	0.788029	0.581803	0.791370	0.578139	0.633477	0.612952
16	0.804844	0.451311	0.681251	0.815904	0.832099	0.710299	0.812074	0.724108	0.753642	0.731726
32	0.649174	0.838128	0.841866	0.803400	0.851227	0.863427	0.846449	0.855327	0.867388	0.824043
64	0.615672	0.815687	0.824372	0.749573	0.828401	0.842246	0.803179	0.845814	0.824344	0.794365
128	0.549423	0.810495	0.818283	0.740639	0.798078	0.830497	0.779144	0.782451	0.808318	0.768592
256	0.453491	0.813083	0.839093	0.709581	0.769872	0.824317	0.753225	0.685260	0.777671	0.736177

Table 25. F-measure values obtained by the same methodology - Second Dataset with MAE.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	ensemble	average
2	0.533333	0.666667	0.666667	0.000000	0.000000	0.400000	0.000000	0.000000	0.000000	0.251852
4	0.699432	0.000000	0.180753	0.638923	0.660138	0.343339	0.651213	0.684111	0.636920	0.499425
8	0.831471	0.561523	0.751577	0.804433	0.841596	0.720080	0.849399	0.819612	0.800937	0.775625
16	0.735340	0.664176	0.790507	0.761272	0.817885	0.787931	0.781050	0.842615	0.803114	0.775988
32	0.774302	0.655077	0.763870	0.806769	0.856804	0.770129	0.815032	0.854661	0.794993	0.787960
64	0.591225	0.793741	0.830226	0.729029	0.820458	0.818502	0.814115	0.842316	0.827723	0.785260
128	0.553342	0.828892	0.825684	0.765760	0.827287	0.818231	0.794853	0.838655	0.808024	0.784525
256	0.516029	0.774741	0.832942	0.734553	0.792478	0.814926	0.788030	0.802740	0.825151	0.764621

Table 26. F-measure values obtained by the same methodology - Second Dataset with MAAE.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	ensemble	average
2	0.692500	0.500000	0.500000	0.690000	0.693333	0.500000	0.693333	0.595833	0.672500	0.615278
4	0.665000	0.500000	0.500000	0.689167	0.672500	0.500000	0.748333	0.596667	0.686667	0.617593
8	0.500000	0.500000	0.500000	0.500000	0.500000	0.500000	0.500000	0.500000	0.500000	0.500000
16	0.810000	0.500000	0.500000	0.776667	0.812500	0.592500	0.777500	0.695833	0.805000	0.696667
32	0.802500	0.500000	0.523333	0.778333	0.835833	0.647500	0.789167	0.696667	0.821667	0.710556
64	0.771667	0.739167	0.783333	0.699167	0.785833	0.790000	0.800000	0.678333	0.813333	0.762315
128	0.807500	0.657500	0.754167	0.770833	0.780833	0.752500	0.770833	0.670833	0.794167	0.751019
256	0.715000	0.769167	0.791667	0.690000	0.739167	0.818333	0.795833	0.687500	0.835833	0.760278

Table 27. ROC-AUC values obtained by the same methodology - Boon Dataset with MAE.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	ensemble	average
2	0.711667	0.5000	0.500000	0.696667	0.680000	0.500000	0.751667	0.595000	0.698333	0.625926
4	0.500000	0.5000	0.500000	0.500000	0.500000	0.500000	0.500000	0.500000	0.500000	0.500000
8	0.785833	0.5000	0.500000	0.675833	0.786667	0.500000	0.734167	0.645000	0.760833	0.654259
16	0.810000	0.5000	0.500000	0.690833	0.786667	0.496667	0.752500	0.669167	0.777500	0.664815
32	0.735833	0.5000	0.500000	0.715000	0.741667	0.500000	0.742500	0.655000	0.766667	0.650741
64	0.807500	0.5000	0.500000	0.790833	0.821667	0.607500	0.773333	0.722500	0.851667	0.708333
128	0.791667	0.5000	0.590000	0.750833	0.824167	0.713333	0.805833	0.695000	0.810000	0.720093
256	0.721667	0.7075	0.761667	0.673333	0.726667	0.736667	0.764167	0.666667	0.794167	0.728056

Table 28. ROC-AUC values obtained by the same methodology - Boon Dataset with MAAE.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	ensemble	average
2	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.500000
4	0.7125	0.5000	0.5525	0.7300	0.6575	0.4750	0.7075	0.7525	0.7350	0.646944
8	0.8025	0.5325	0.6325	0.8125	0.7925	0.7050	0.8000	0.7025	0.7325	0.723611
16	0.8150	0.6500	0.7575	0.8225	0.8350	0.7725	0.8175	0.7800	0.8000	0.783333
32	0.7400	0.8575	0.8625	0.8150	0.8550	0.8725	0.8475	0.8675	0.8800	0.844167
64	0.7200	0.8325	0.8450	0.7675	0.8350	0.8525	0.8150	0.8650	0.8450	0.819722
128	0.6925	0.8300	0.8300	0.7475	0.8025	0.8375	0.7850	0.8225	0.8325	0.797778
256	0.6450	0.8250	0.8500	0.7325	0.7825	0.8275	0.7750	0.7700	0.8075	0.779444

Table 29. ROC-AUC values obtained by the same methodology - CHBMIT Dataset with MAE.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	ensemble	average
2	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.500000
4	0.7200	0.5000	0.5575	0.6825	0.6725	0.5050	0.7000	0.7525	0.7175	0.645278
8	0.8400	0.6975	0.8025	0.8150	0.8450	0.7775	0.8550	0.8375	0.8275	0.810833
16	0.7900	0.7475	0.8250	0.7750	0.8250	0.8125	0.7950	0.8525	0.8300	0.805833
32	0.8125	0.7450	0.8125	0.8150	0.8625	0.8025	0.8225	0.8675	0.8300	0.818889
64	0.7125	0.8150	0.8475	0.7500	0.8225	0.8300	0.8250	0.8525	0.8450	0.811111
128	0.6950	0.8400	0.8375	0.7800	0.8325	0.8300	0.8100	0.8500	0.8275	0.811389
256	0.6725	0.7950	0.8400	0.7400	0.7950	0.8200	0.7925	0.8375	0.8350	0.791944

Table 30. ROC-AUC values obtained by the same methodology - CHBMIT Dataset with MAAE.

6 Conclusion

In this article, we re-implemented the approach proposed by [1], an investigation of the accurately tested feature sizes and proposed a new classifier. This classification approach, based on deep learning for detecting epileptic seizures using EGG had not been explored previously. We adopted a self-concealment that allows us to construct a smaller representation space. Among the variety of metrics, our method stands out

according to the ROC curve. The proposed approach still needs further.
The original authors leave gaps that make it impossible to fully reproduce the results obtained. For example, the lack of information about the neural classifier used in the last sub-section. As well as the lack of information about the sampling process of the first and second data sets. In addition to providing an example, exact information about the number of times or batch size, the results obtained can be considered at most a replication.
The method is easily portable to other tasks and codes are a byproduct of this work. As future work, we can employ data augmentation that would allow the method to better learn the parameters adopted.

References

1. T. Wen and Z. Zhang. "Deep Convolution Neural Network and Autoencoders-based Unsupervised Feature Learning of EEG Signals." In: **IEEE Access** PP (May 2018), pp. 1–1.
2. M. Saab and J. Gotman. "A system to detect the onset of epileptic seizures in scalp EEG." In: **Clinical Neurophysiology** 116.2 (2005), pp. 427–442.
3. L. Kuhlmann, A. N. Burkitt, M. J. Cook, K. Fuller, D. B. Grayden, L. Seiderer, and I. M. Mareels. "Seizure detection using seizure probability estimation: Comparison of features used to detect seizures." In: **Annals of biomedical engineering** 37.10 (2009), pp. 2129–2145.
4. A. Shoeb, H. Edwards, J. Connolly, B. Bourgeois, S. T. Treves, and J. Guttag. "Patient-specific seizure onset detection." In: **Epilepsy & Behavior** 5.4 (2004), pp. 483–498.
5. A. Shoeb, A. Kharbouch, J. Soegaard, S. Schachter, and J. Guttag. "A machine-learning algorithm for detecting seizure termination in scalp EEG." In: **Epilepsy & Behavior** 22 (2011), S36–S43.
6. I. Ullah, M. Hussain, H. Aboalsamh, et al. "An automated system for epilepsy detection using EEG brain signals based on deep learning approach." In: **Expert Systems with Applications** 107 (2018), pp. 61–71.
7. K. C. Chua, V. Chandran, U. R. Acharya, and C. M. Lim. "Application of higher order spectra to identify epileptic EEG." In: **Journal of medical systems** 35.6 (2011), pp. 1563–1571.
8. N. Nicolaou and J. Georgiou. "Detection of epileptic electroencephalogram based on permutation entropy and support vector machines." In: **Expert Systems with Applications** 39.1 (2012), pp. 202–209.
9. U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, and H. Adeli. "Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals." In: **Computers in biology and medicine** 100 (2018), pp. 270–278.
10. R. Hussein, H. Palangi, R. Ward, and Z. J. Wang. "Epileptic seizure detection: A deep learning approach." In: **arXiv preprint arXiv:1803.09848** (2018).
11. G. Xun, X. Jia, and A. Zhang. "Detecting epileptic seizures with electroencephalogram via a context-learning model." In: **BMC medical informatics and decision making** 16.2 (2016), p. 70.
12. A. Emami, N. Kunii, T. Matsuo, T. Shinozaki, K. Kawai, and H. Takahashi. "Autoencoding of long-term scalp electroencephalogram to detect epileptic seizure for diagnosis support system." In: **Computers in biology and medicine** (2019).
13. F. Chollet et al. "Keras: The python deep learning library." In: **Astrophysics Source Code Library** (2018).
14. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. "Tensorflow: A system for large-scale machine learning." In: (2016), pp. 265–283.
15. A. D. la Fuente and R. Aduviri. "[Re] Variational Sparse Coding." Python. In: **ReScience C** 5.2 (May 2019), #2.
16. A. Shoeb and J. Guttag. "Application of Machine Learning to Epileptic Seizure Detection." In: ICML'10 (2010), pp. 975–982.
17. R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. Elger. "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state." In: **Physical review. E, Statistical, nonlinear, and soft matter physics** 64 (Jan. 2002), p. 061907.
18. C. Kamath. "Analysis of EEG dynamics in epileptic patients and healthy subjects using Hilbert transform scatter plots." In: **Open Access Library Journal** 2.1 (2015), p. 1.
19. A. H. Shoeb. "Application of machine learning to epileptic seizure onset detection and treatment." PhD thesis. Massachusetts Institute of Technology, 2009.
20. Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T. H. Falk, and J. Faubert. "Deep learning-based electroencephalography analysis: a systematic review." In: **Journal of neural engineering** (2019).