Brain Signal Classification

Burak B. Günden, Efe Ertekin, Arda Özdemir, Yasin Yılmaz
gundenb,ertekinef,ozdemirard,yilmazy@mef.edu.tr
MEF University
January 14, 2022

***Abstract*—In this study, ERP (Event-related potential) signal classification was made. The P300 dataset, which contains the data of school-age children, was used as the dataset of these signals. There are 138 male and 112 female subjects in this dataset. Four different algorithms were used for the classification process. Two of them are machine learning, and the remaining two are deep learning algorithms. Convolutional Neural Network (CNN), Recurrent Neural Network/LSTM (Long short-term memory), Linear Discriminant Analysis (LDA), Support Vector Machines (SVM), and Logistic Regression (LOR) were used. Accuracy values hover around 64% for all algorithms. Algorithms could not provide a clear advantage over each other. No major effects of deep learning algorithms have been observed.**

***Index Terms*—ERP, P300, CNN, Convolutional Neural Network, RNN, Recurrent Neural Network, LDA, Linear Discriminant Analysis, Logistic Regression, SVM, Support Vector Machines, Classification**

I. Introduction

In recent years, artificial intelligence has been developed rapidly within its increasing usage and now it is more capable of creating solutions to real-life problems. The creation of the Brain-Computer Interfaces (BCI) made it possible for people to work on ERP and EEG data. This technology enables researchers to work on data that is a creation of the electrical activity of the human brain. There are various types of BCI that differ. As an example, there are BCIs with 8 channels and 16 channels. When it comes to analyzing and interpreting this data, artificial intelligence methods have come to the fore. In line with this purpose, various artificial intelligence methods were created for the classification and expression of the obtained data.

In the scope of the project, a dataset collected by an experiment called Guess the number(which is a P300 ERP BCI experiment) was used. In this experiment, the participants were asked to pick a number between 1 and 9. Then they were exposed to visual stimuli. Experimenters tried to guess the number by ERP data during the time participants were exposed to visual stimuli. Lastly, the exact and guessed numbers were recorded. In this experiment, only three EEG channels were used for recording. To store the dataset, the BrainVision format is used.[1]

In the project, the epochs are classified into two classes which are the thought number (target epoch) and non-thought number (non-target epoch). The MNE library is used for applying filtering and artifact removal. For filtering, low pass and high pass filtering are used. For artifact removal, the peak-to-peak amplitude rejection is used. Three linear classifiers(LDA, SVM, LOR) and two neural networks(CNN and RNN) are implemented. Keras and scikit-learn libraries are used in the implementation of these methods. Lastly, the evaluation metrics are given of these four methods.

II. Algorithms

A. Preprocessing Methods

Preprocessing is the process that is responsible for converting the collected raw data into a usable form in classification algorithms. Five different preprocessing methods were used in this project. Detailed explanations are below.

1) Low-pass and high-pass filtering: In this project, filtering was used in the preprocessing phase. The filters used are low pass and high pass filters. While the low pass filter attenuates the components above the specified cutoff frequency, it protects the components below it. The high pass filter, on the other hand, attenuates the components below the specified cutoff frequency, while passing the components above it. The filter method on the MNE library is used for filtering on the dataset. The frequency of the low pass filter was selected as 0.1, the value of the high pass filter was selected as 30 and filtering was done on the data [2].

2) Epoch Extraction: Epochs are the time interval between events (stimuli). Events, The occurring time of the event, and the description of the event are obtained from the raw signal data and assigned to a variable called events_loaded. This raw signal is available as a result of the data loading phase. To do that kind of extraction, MNE is used. The extracted epochs are labeled with the id of the event, however, this extraction is made for the classification so they are marked as target (0)[3].

3) Baseline Correction: Baseline correction is another preprocessing technique that is used in the project. The dataset is a time-resolving signal. There might be noise in the data set. According to this information, the baseline correction technique is used to separate the real spatiotemporal values from the other noise effects. Also, extracted epochs are sent and used in the baseline correction phase [4][5].

4) Artifact Removal with the peak-to-peak amplitude rejection: Artifacts are the obtained EEG signals which are not formed as a result of the brain activity. To be able to work on the EEG signals artifacts must be removed. The peak to peak amplitude is used for rejecting the artifacts in the data. Peak to peak amplitude is the circular change between the highest and lowest amplitude values. In the project, epochs mentioned (target value and non-target) are concatenated and one single instance is obtained. Then peak-to-peak amplitude rejection is applied on this single instance to remove artifacts. Using a rejection dictionary, peak-to-peak signal amplitudes in unreasonable limits are rejected. The limit is set to 150 µV. For the remaining processes, the MNE library is used.[6]

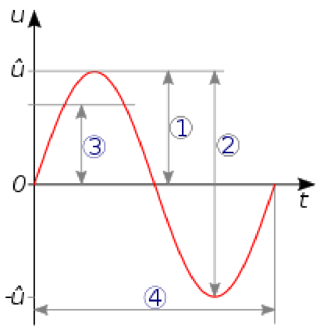


Fig. 1. Amplitude Rejection Example (two represents the peak-to-peak amplitude)

5) Windowed Means Feature Extraction: The dataset is coming from a P300-based Brain-Computer Interface experiment and also, windowed means feature extraction is compatible with P300 format. The windowed means feature extraction is used for linear classifiers. It takes the average of time intervals and merges these averages with the related EEG channels. As a result of the windowed mean feature extraction method, reduced spatiotemporal feature vectors are obtained. The windowed means method takes the dataset as a 3D tensor. The dimensions of the tensor are epochs_count, channels_count, and values_count. There is a class called param. It contains configuration parameters such as interval after the stimulus, the number of time windows, and the time before the stimulus that was used for epoch extraction, etc. After the individual average calculations, there is an output as a 2D tensor. Output dimensions are epochs_count and features_count. Lastly, the result of the method is normalized using zero mean and unit variance [7].

B. Classifiers

Classifiers are machine learning algorithms that are used to give data inputs a class label. In this project, there are four different algorithms. Half of them are based on machine learning and the other half are based on neural networks. Detailed explanations are below.

1) Convolutional Neural Network: CNN (Convolutional Neural Network) is a Deep Learning algorithm, which enables data with certain depths to be recognized after being trained. This classification method requires less preprocessing than other algorithms. One of the features that make CNN stand out is that it needs fewer parameters, especially compared to ANN (Artificial Neural Network) [8]. In addition to these, it is easier to use the features suitable for the problem in feature extraction, and more abstract features are obtained as the layer progresses.

Convolution Layer is the first part of this algorithm. Here it is desirable to extract high-level features. An example is the edges of a picture. But in the first layers of the convolution layers, low-level features such as color are obtained. High-level features can be extracted with multiple convolution layers. There is an example in the Figure2.

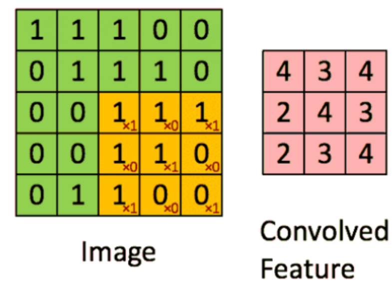


Fig. 2. Convoluting a 5x5x1 image with a 3x3x1 kernel to get a 3x3x1 convolved feature

The Pooling layer is used to speed up the computer's operations by reducing the size of the extracted features. By using the dominant features, smaller data is obtained with this process. It has 2 different methods. One is Max Pooling and the other is Average Pooling. In Max Pooling, the candidate cells switch to the highest pooled version. In Average, candidate cells are averaged. Figure3 shows the difference between these methods.

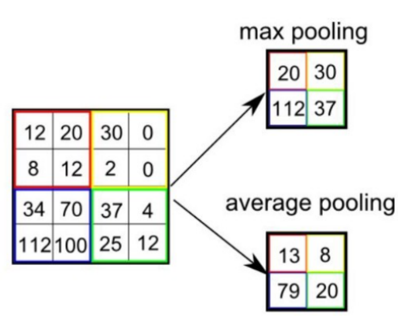


Fig. 3. Pooling Layer

Fully Connected Layer is now the layer where classification is made. Data converted to column vectors can now be sent to the feed-forward neural network and backpropagation can be applied in all iterations. The model can now distinguish between dominant and low-level features at certain time intervals. It classifies them with some techniques (ex: Softmax Classification).

2) LSTM Neural Network: Long short-term memory (LSTM) is one of the recurrent neural network architectures. LSTM is used in the Deep Learning field. It is called recurrent because it has feedback connections. These connections are the main difference from the feed-forward neural network. In addition to single data point processes, LSTM can also process a whole sequence of data. While it can process current data with LSTM cells, it can also store historical status. It also has a feedback mechanism. There is an example for the cell of LSTM in the Figure4. LSTM is especially useful for making predictions, processing, and classifying if the data is time-series data. The usage of LSTM has no difference from the other methods. Firstly, LSTM is compiled and the output model is fitted with the fit method. Lastly after the evaluation operation. Results are printed to the console [9].

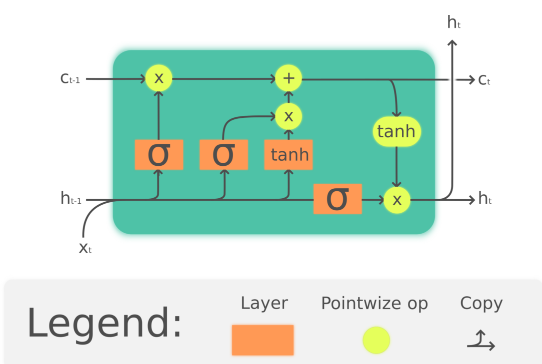


Fig. 4. LSTM Cell

3) Linear Discriminant Analysis: In this project, Linear discriminant analysis was performed using the sklearn library. In the linear discriminant analysis (LDA) method, it is aimed to show the original matrix in less dimensional space. There are 3 main steps to do this. The first step is to calculate separability, also known as between-class variance, between different classes. The second step calculates the distance between the mean and the instance of each class. This is called within-class variance. In the third step, a lower dimensional space is created. This maximizes the space between-class variance and minimizes the within-class variance [10].

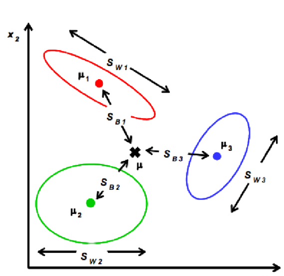


Fig. 5. LDA Example

4) Support Vector Machines: The SVM algorithm provides the separation of different data types from each other with a drawn hyperplane. This hyperplane to be drawn must be drawn in a line of sight. For this, the most extreme members of the data types (usually those closest to the opposite type) serve as support vectors. In the middle of the parallelism created by the support vectors, a line is drawn parallel to them. This line is the hyperplane. In order to obtain an optimal hyperplane, the maximum margin must be provided. Margin is the difference of the parallels drawn by the support vectors of these two different types of data [11].

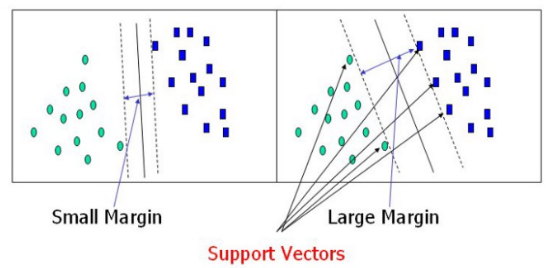


Fig. 6. Support Vector Machine Example

With this obtained hyperplane, the future data is estimated. Data falling on different sides of the line is classified differently. Also, SVM can handle high-dimensional data without the high risk of over-fitting problem thanks to its structure [12].

5) Logistic Regression: Logistic regression is like a regression problem where the dependent variable is a categorical variable. It is widely used in linear classification problems. Although it is called regression, there is a classification here.

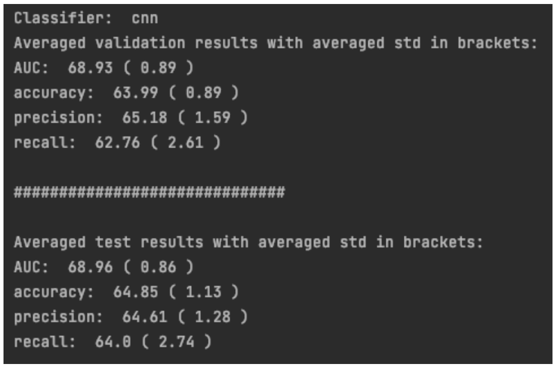
Logistic regression is a statistical method used to analyze a dataset with one or more independent variables that determine an outcome. The outcome is measured with a binary variable (there are only two possible outcomes). In logistic regression, the dependent variable contains data encoded in binary or binary, i.e. only 1 (TRUE, success, pregnant, etc.) or 0 (FALSE, error, non-pregnant, etc.).

III. Results

In the project scope, classification methods and neural network methods applied within two different datasets that are validation and test. Their evaluation metric plays an important role to have a better understanding of their performance. To measure the success of the classification methods, the results for the following metrics were obtained. Accuracy, precision, recall, and AUC (area under the ROC curve).

The results of the models are shown below. According to the AUC score, the best one in the validation dataset is the CNN with 68.93/100. In the validation dataset, most accuracy comes with SVM (64.22/100), most precision comes with SVM (65.57/100), and lastly most recall comes with LDA (67.92).

On the other hand, when the test dataset is used, best AUC score comes with CNN (68.96/100), most accuracy comes with SVM (65.6/100), most precision comes with RNN (68.21/100), and most recall comes with LDA (68.37/100).

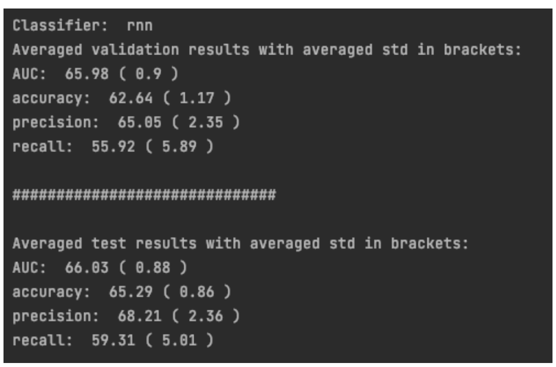


```
Classifier:  cnn
Averaged validation results with averaged std in brackets:
AUC:  68.93 ( 0.89 )
accuracy:  63.99 ( 0.89 )
precision:  65.18 ( 1.59 )
recall:  62.76 ( 2.61 )

#############################

Averaged test results with averaged std in brackets:
AUC:  68.96 ( 0.86 )
accuracy:  64.85 ( 1.13 )
precision:  64.61 ( 1.28 )
recall:  64.0 ( 2.74 )
```

Fig. 7. CNN Result

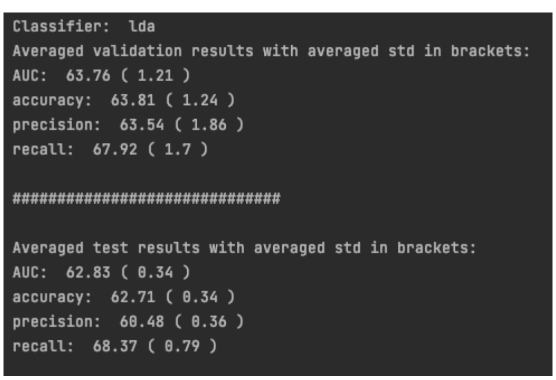


```
Classifier:  rnn
Averaged validation results with averaged std in brackets:
AUC:  65.98 ( 0.9 )
accuracy:  62.64 ( 1.17 )
precision:  65.05 ( 2.35 )
recall:  55.92 ( 5.89 )

#############################

Averaged test results with averaged std in brackets:
AUC:  66.03 ( 0.88 )
accuracy:  65.29 ( 0.86 )
precision:  68.21 ( 2.36 )
recall:  59.31 ( 5.01 )
```

Fig. 8. RNN Result

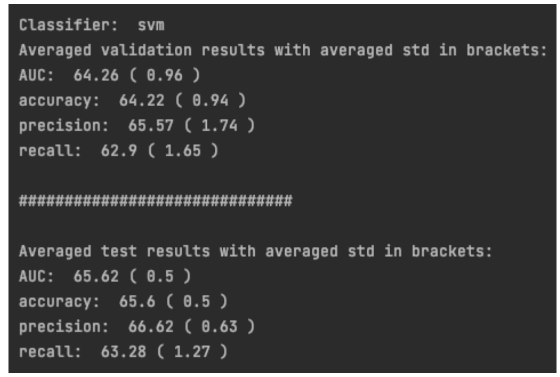


```
Classifier:  lda
Averaged validation results with averaged std in brackets:
AUC:  63.76 ( 1.21 )
accuracy:  63.81 ( 1.24 )
precision:  63.54 ( 1.86 )
recall:  67.92 ( 1.7 )

#############################

Averaged test results with averaged std in brackets:
AUC:  62.83 ( 0.34 )
accuracy:  62.71 ( 0.34 )
precision:  60.48 ( 0.36 )
recall:  68.37 ( 0.79 )
```

Fig. 9. LDA Result

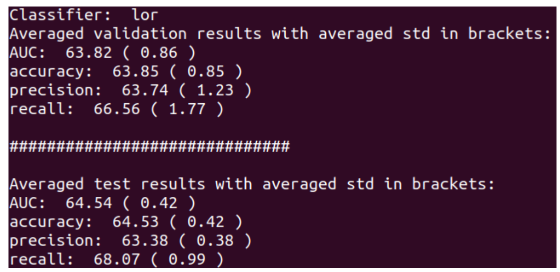


```
Classifier:  svm
Averaged validation results with averaged std in brackets:
AUC:  64.26 ( 0.96 )
accuracy:  64.22 ( 0.94 )
precision:  65.57 ( 1.74 )
recall:  62.9 ( 1.65 )

#############################

Averaged test results with averaged std in brackets:
AUC:  65.62 ( 0.5 )
accuracy:  65.6 ( 0.5 )
precision:  66.62 ( 0.63 )
recall:  63.28 ( 1.27 )
```

Fig. 10. SVM Result



```
Classifier:  lor
Averaged validation results with averaged std in brackets:
AUC:  63.82 ( 0.86 )
accuracy:  63.85 ( 0.85 )
precision:  63.74 ( 1.23 )
recall:  66.56 ( 1.77 )

#############################

Averaged test results with averaged std in brackets:
AUC:  64.54 ( 0.42 )
accuracy:  64.53 ( 0.42 )
precision:  63.38 ( 0.38 )
recall:  68.07 ( 0.99 )
```

Fig. 11. Logistic Regression Result

IV. Conclusion

The aim of the project is to apply different classification methods on an ERP dataset called P300 and compare the methods among them to find a better approach to work on the EEG datasets obtained by BCIs for several purposes. In the project, three linear classifiers and two neural network methods were used to classify the dataset after the preprocessing method was applied to the datasets. As discussed in the results section, the linear classifier methods outperform the neural network methods on the classification of small datasets such as P300. So linear classifier methods are a better choice to classify the P300 dataset.

References

[1] L. Vařeka, "Evaluation of convolutional neural networks using a large multi-subject p300 dataset," *Biomedical Signal Processing and Control*, vol. 58, p. 101837, 2020.

[2] "Difference between low pass filter and high pass filter," Dec 2019. [Online]. Available: https://www.geeksforgeeks.org/difference-between-low-pass-filter-and-high-pass-filter/

[3] K. S. R. Murty and B. Yegnanarayana, "Epoch extraction from speech signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1602–1613, 2008.

[4] K. H. Liland, E.-O. Rukke, E. F. Olsen, and T. Isaksson, "Customized baseline correction," *Chemometrics and Intelligent Laboratory Systems*, vol. 109, no. 1, pp. 51–56, 2011.

[5] "Baseline correction," https://neuro.inf.unibe.ch/AlgorithmsNeuroscience/Tutorial_files/BaselineCorrection.html, accessed: 2022-01-09.

[6] X. Jiang, G.-B. Bian, and Z. Tian, "Removal of artifacts from eeg signals: a review," *Sensors*, vol. 19, no. 5, p. 987, 2019.

[7] L. Vareka and P. Mautner, "Using the windowed means paradigm for single trial p300 detection," in *2015 38th International Conference on Telecommunications and Signal Processing (TSP)*. IEEE, 2015, pp. 1–4.

[8] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*. Ieee, 2017, pp. 1–6.

[9] P. Davidson, R. Jones, and M. Peiris, "Detecting behavioral microsleeps using eeg and lstm recurrent neural networks," in *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. IEEE, 2006, pp. 5754–5757.

[10] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien, "Linear discriminant analysis: A detailed tutorial," *AI communications*, vol. 30, no. 2, pp. 169–190, 2017.

[11] "Support vector machine," https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47, accessed: 2022-01-09.

[12] D. A. Pisner and D. M. Schnyer, "Support vector machine," in *Machine Learning*. Elsevier, 2020, pp. 101–121.

Appendix

Code of the project can be reachable at: https://github.com/eertekin99/MNE