# Brain-Computer Interface Based on Classification of Statistical and Power Spectral Density Features

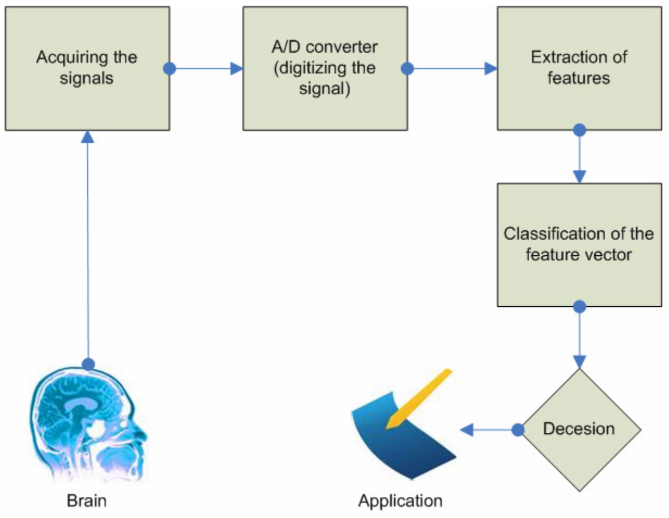
Ramez T. Mina[1], Amir Atiya[2], Mohamed I. Owis[1], Yasser M. Kadah[1].
[1,] Systems and Biomedical Engineering Dept., Cairo University, Giza, Egypt
[2] Computer Engineering Dept., Cairo University, Giza, Egypt
e-mail: rtmmina@gmail.com

*Abstract*- **Brain Computer Interface::BCI is a system that reads the thoughts of a patient. Research showed that the encephalogram (EEG) signals are different in the information they carry according the patient's thoughts.**
**In this paper, we're comparing the classification methods on a dataset of BCI to get the best result of discrimination between up and down movements. We are featuring the signals using statistics and power spectral analysis (PSD), and classifying the resulting features using minimum distance, voting K-nearest, perceptron and backpropagation classifiers.**
***Keywords* - BCI, feature extraction, Classification, statistics and power spectral density (PDS).**

## I. Introduction

Many people who suffer from severe motor disabilities, who are locked-in and can't move by themselves, need a way to communicate with their environments.
Brain Computer Interface (BCI) is one of the best research areas that has been developed in the last decades to get a new way of communication for those people.
BCI research depends mainly on reading the signals from the brain and tries to get the discriminant information from them, which could help in classifying these signals to many applications. One of these applications and which has the most interest for many researchers, is to distinguish between two directions of movements in these signals, which could help locked-in people in a wheel chair decide their movements

In the last four decades, BCI research has been developed for the Single-trail EEG associated with hand movements, and which will we focus on in this paper.
***Brief history***
Researchers discovered that EEG signals differ from left hand and right hand movements (or up and down movements). They could also distinguish event-related synchronization and desynchronization (ERS/ERD)[1]. Statistics techniques on the μ rhythm power were applied which showed ERD followed by ERS[2]
Many classification techniques were used, like Fisher discriminant analysis [1], Bayes theorem was used for classification [2] and Support Vector Machines-SVM[3]
BCI research went in a lot of areas, starting from the hardware to acquire the signals from the brain (EEG) and digitize them to be introduced to digital filters to remove the noise and prepare them for the next step, that is, to extract the dominant features and classify them, which will give a complete figure for the last step, namely the application. [4]
Fig. 1 shows a block diagram for a BCI system model.
The system has many phases and areas of research and they change according to the final application. The one we are focusing on has the application of moving up or down, according to what direction the patient thinks of. In this paper, we are focusing on the extraction of the features that discriminate between the two directions, and the classification techniques that will decide the direction.


Acquiring the signals
A/D converter (digitizing the signal)
Extraction of features
Classification of the feature vector
Decesion
Brain
Application
Fig. 1 BCI model's block diagram

Though the applications for BCI are numerous ranging from understanding the thought of the patient, to trying to let him move a cursor on a screen of a computer or attempting to read the EEG signals to control a small/miniature mobile robot in a closed area, we, nevertheless, have a lot of challenge in producing such a system; for example, the accuracy of the system we want to approach depends on how much signals we have, which will increase the cost, and this could be overcome by improving the preparation of the data before introducing them to the classification techniques[3].
*Convention:*
*In the first international meeting for BCI technology, they have the convention of* BCI *for a system that does* not depend on the brain's normal output pathways of peripheral nerves and muscles [5].
They measured the feedback of the signal after presenting a stimulus on a computer screen, and they analyzed this feedback signal to classify the unlabeled signals online later.

## II. Methodology

*Description of the dataset*
The dataset was taken from a healthy patient. He was asked to move a cursor up or down on a computer screen, and then they measured the feedback. Each trial took 6s, though the dataset is only for the feedback period which is 3.5s (from 2s to 5.5s) with a sample rate 256 hz, so we have 896 samples for each trial.
The dataset is a measure of 6 channels from the scalp, 135 for the up direction and 133 for the down direction.

Feature extraction:
Features' extraction went in two directions, one was statistical

and the other was measuring the power density spectral.
In the statistical techniques, we applied mean, variance, amplitude (maximum point in the signal) [3] and minimum on the training EEG signals And for the power spectral density we applied Welch and Thomson multitaper methods.
Power spectral [6]
Power spectral density (PSD), is a way to describe the power distribution contained in the signal.
From here, we used two different methods for extracting features using PSD, Welch and multitaper[7].
*Welch:* which is the averaged periodograms of overlapped, windowed signal sections[8], in welch method we do the following,

i. Divide the time series data into overlapped segments.
ii. compute a modified periodogram of each segment
iii. average the PSD estimates

Other steps for pwelch.
The vector x is segmented into eight sections of equal length, each with 50% overlap. Any remaining (trailing) entries in x that cannot be included in the eight segments of equal length are discarded. Each segment is windowed with a Hamming window that is the same length as the segment.[8]
*Multitaper:* which is the spectral estimate from combination of multiple orthogonal windows (or "tapers")[8].

After applying both statistical and PSD feature extraction methods, we need to check the significance of these methods, which means that could discriminate between the up and down movements, for this we applied T test on these methods for the up and down EEG signals.
Though, the features were a lot (were 1249 features), which increased the pool dimensions.
To minimize these dimensions, instead of taking T test with p value equal to 0.05, we took the least 100 values of p values (our new P value is 2.2604e-005).
Again, we minimized the feature vector length used after getting the correlation coefficient. For each one of the features, we got the sum of its correlation coefficient values with the rest of the features, and took the 40 features that have the least sum.

For the features extraction, we used the built-in functions in matlab version 7. Mean, var, max, min, pwelch and pmtm were used to apply the methods, and ttest2 was used to get the T test and P values, and corr2 to get the correlation coefficients.

Classifiers:
In classification, we applied many techniques, starting from the simple one to the highly complicated one, from the weak one to the highly efficient one.

1- Minimum Distance Classifier:
One of the simplest techniques for classifications and it goes with simple steps [9].

i. Divide the pool into the two clusters, one for the up movement and the other for the down movement.
ii. Calculate the mean vector for each cluster
iii. Calculate the distance between the test sample and the two mean vectors of the up and down movements.
iv. The test sample will belong to the cluster whose mean vector has the short distance to the sample.

2- Voting k-Nearest Neighbor Classifier:
Another simple classifier and it has very efficient accuracy [9].

i. Divide the pool into the two clusters, one for the up movement and the other for the down movement
ii. Calculate the distance of the test sample vector to each vector of the training data.
iii. Put all the resulting distances with their corresponding clusters in an array.
iv. Sort the array.
v. Take the first sample (or the first three, five…etc. samples).
vi. Check which cluster has more samples that are near to the test sample, and this would be the class of the test sample.

3- Perceptron Classifier
Perceptron is the simplest algorithm for the neural network. It uses one layer only, and it could have many neurons[10].
Fig. 2 shows the architecture of the perceptron.

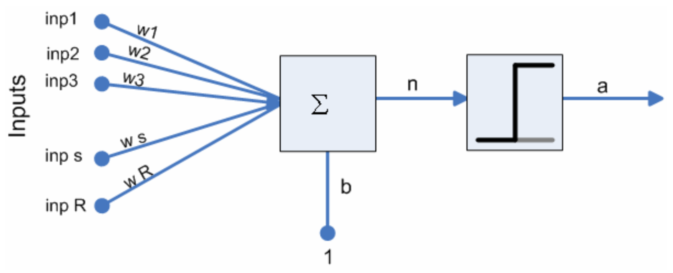


Fig. 2 Perceptron Model

The idea of the Perceptron is to decide if the sample is 0 or 1, so if the value of n is equal to or greater than 0, then the sample value is 1; otherwise it is equal to 0, though we need to use the function hardlim as a transfer function for the net input n, see Fig. 3.

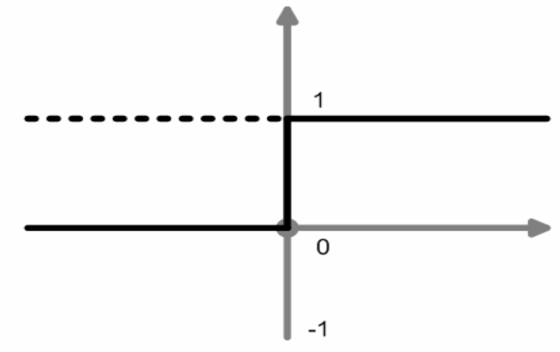


Fig. 3 Perceptron Transfer Function
The neural networks work in a way that looks like the curve fitting, as they try to adjust the weights that are initialized in the beginning to be able to estimate the right outputs when the new inputs are introduced.

Matlab version 7.0.1 has built-in functions and toolbox for the perceptron, and we used it in our classifier.

4- Backpropagation Classifier:
Considered as one of the best algorithms for neural networks
It is composed of many layers. The last one is called the output layers, and the preceding layers are called hidden layers.

As shown in Fig. 4, it works in this way. After applying the input vector to the network, it gets the output and compares it to the label/target it has, then gets the error difference. After that, it goes to the output layer and updates the weights of it, and from the new weights, we go backward to the preceding layer and calculate its new weights and so on with the rest of the preceding layers [11].

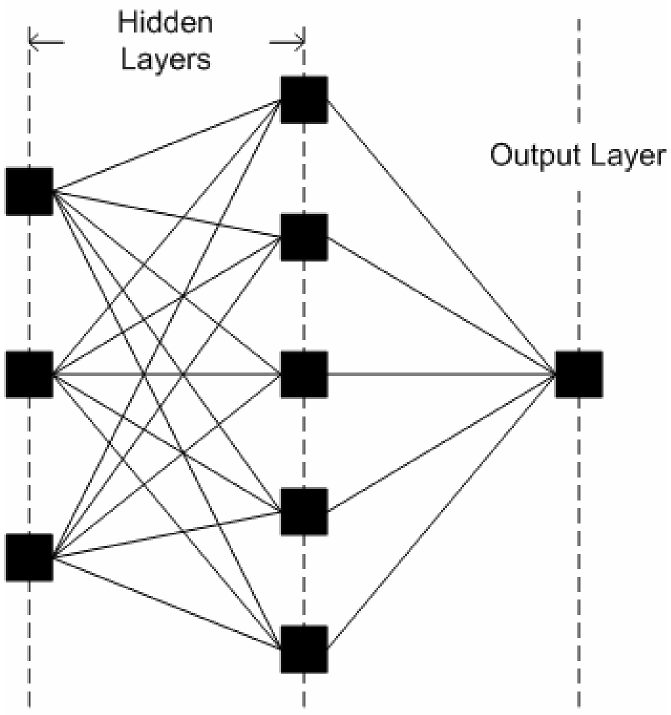


Fig. 4 Backpropagation Model

## III. Results

Table 1 shows the results we had for both training and test data.

The best results were mainly for the statistical features, in both training data and test data. And the best classifier was the back propagation for the training data and minimum distance for the test data.

Perceptron was disappointing in its results, unless for solid data, it gave almost the same result for all the features in both, training dataset and test dataset, which gave them almost 50% accuracy and error.

## IV. Discussion

Peceptron had almost all its results voting for one movement direction which was the down movement, except for applying it with no features on the dataset, and that gave 100% accuracy.

In both training and test dataset, it gave us that all signals are classified as down movements.

Statistical feature vector had the least cost, as they were 4 features for 6 channels, and mainly the vector is minimized after applying T test, and it also had the best results for all the classifiers

Backpropagation had almost the best results on both the training and test datasets.

Our model had bad accuracy using the minimum distance classifier for the test dataset.

## V. Conclusion

The power spectral density technique had the least accuracy, thought the statistical technique had really good accuracy; we are trying now to combine PSD with the statistical and different techniques to see what that will lead to.

Table 1
COMPARISON OF DIFFEREBNT CLASSIFIERS RESULTS ON THE DIFFERENT STEPS OF THE FEATURE EXTRACTION PROCESS

| **Classifiers** | | Minimum Distance | Voting K-nearest neighbor | Perceptron(500 epochs) | Backpropagation |
|---|---|---|---|---|---|
| | | Accuracy | Accuracy | Accuracy | Accuracy |
| **Training Data** | Full features | 71.2687 | 82.8358<br>79.8507 | 100 | 91.0448 |
| | P value(0.05) | 64.9254 | 66.7910<br>64.1791 | 49.6269 | 74.6269 |
| | Statistical | 76.1194 | 72.7612<br>75.7463 | 49.6269 | 73.1343 |
| | PSD | 64.9254 | 66.7910<br>64.1791 | 49.6269 | 71.6418 |
| | Least 100 P-values | 64.9254 | 66.7910<br>64.1791 | 49.6269 | 71.6418 |
| | Our Model | 63.4328 | 55.9701<br>61.9403 | 49.6269 | 62.6866 |

| | | | | | |
|---|---|---|---|---|---|
| **Test Data** | Full features | 88.0546 | 67.2355 65.8703 | 86.0068 | 79.8635 |
| | P value(0.05) | 52.9010 | 56.6553 53.5836 | 49.8294 | 63.1399 |
| | Statistical | 73.7201 | 60.7509 63.1399 | 49.8294 | 67.2355 |
| | PSD | 52.9010 | 56.6553 53.5836 | 49.8294 | 59.0444 |
| | Least 100 P-values | 52.9010 | 56.6553 53.2423 | 49.8294 | 54.2662 |
| | Our Model | 37.8840 | 47.0990 45.7338 | 49.8294 | 54.6075 |

## REFERENCES

[1] Y. Wang, Z. Zhang, Y. Li, X. Gao, Sh. Gao,"BCI Competition 2003—Data Set IV: An Algorithm Based on CSSD and FDA for Classifying Single-Trial EEG" IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. 51, NO. 6, JUNE 2004

[2]S. Lemm, Ch. Schäfer, and G. Curio, "BCI Competition 2003—Data Set III: Probabilistic Modeling of Sensorimotor _ Rhythms for Classification of Imaginary Hand Movements"IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. 51, NO. 6, JUNE 2004

[3]M. Kaper, *Student Member, IEEE*, P. Meinicke, U. Grossekathoefer, Th. Lingner, and H. Ritter," BCI Competition 2003—Data Set IIb: Support Vector Machines for the P300 Speller Paradigm" IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. 51, NO. 6, JUNE 2004

[4]N. Xu, X. Gao, B. Hong, X. Miao, Sh. Gao, *Senior Member, IEEE*, and F. Yang, "BCI Competition 2003—Data Set IIb: Enhancing P300 Wave Detection Using ICA-Based Subspace Projections for BCI Applications" IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. 51, NO. 6, JUNE 2004

[5] G. Blanchard and B. Blankertz, "BCI Competition 2003—Data Set IIa: Spatial Patterns of Self-Controlled Brain Rhythm Modulations" IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. 51, NO. 6, JUNE 2004.

[6]R. Palaniappan, *Student Member, IEEE*, R. Aramesran, *Senior Member, IEEE*, Sh. Nishida, *Fellow, IEEE*, and N. Saiwaki, *Member, IEEE,"* A New Brain–Computer Interface Design Using Fuzzy ARTMAP" IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING, VOL. 10, NO. 3, SEPTEMBER 2002

[7]B. D. Mensh, J. Werfel, and H. Sebastian Seung, "BCI Competition 2003—Data Set Ia: Combining Gamma-Band Power With Slow Cortical Potentials to Improve Single-Trial Classification of Electroencephalographic Signals"IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. 51, NO. 6, JUNE 2004.

[8]Matlab help for Signal Processing toolbox

[9]Y. M. Kadah, A. A. Farag, J. M. Zurada, A. M. Badawi, and A. M. Youssef, "Classification Algorithms for Quantitative Tissue Characterization of Diffuse Liver Diseasr from Ultrasound Images"
IEEE TRANSACTIONS ON MEDICAL IMAGING, VOL 15, NO 4, AUGUST 1996

[10]Matlab help for Neural Network toolbox

[11] j. M. Zurada, *Introduction to Artificial Neural Systems*. Boston, MA: PWS, 1992