Brain Wave Pattern Classification from Virtual Training Environment by Self-Organizing Maps

Bruno Senzio-Savino[1], Mohammad Reza Alsharif[1], Carlos E. Gutierrez[2], Chirstian Penaloza[3], Katsumi Yamashita[4], Faramarz Alsharif[1], Mahdi Khosravy[1] and Mousa Shamsi[5]
[1]Department of Information Technologies, University of the Ryukyus
1 Senbaru, Nishihara, Okinawa 903-0213, Japan
[2]Neural Computation Unit, Okinawa Institute of Science and Technology,
1919-1 Tancha, Onna, Kunigami District, Okinawa 904-0495, Japan
[3]Hiroshi Ishiguro Laboratories, Advanced Telecommunications Research Institute International,
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan
[4]Graduate School of Engineering, Osaka Prefecture University,
3-7-30 Habikino, Sakai-shi, Osaka, 583-0872 Japan
[5]Sahand University of Technology,Sahand New Town, Iran
E-mail : [1]b.senzio@gmail.com,[1]asharif@ie.u-ryukyu.ac.jp ,[1]dr.mahdi.khosravy@ieee.org, [2]carlosengutierrez@gmail.com,
[3]penaloza@atr.jp, [4]yamashita@eis.osakafu-u.ac.jp, [5]shamsi@sut.ac.ir

Abstract: The main purpose of this research is to perform and analyse the performance of a simple brain wave password signal pattern feature classifier by Self Organizing Maps (SOM) and the processing time dedicated to build it. These signals are composed by attention/meditation signal patterns obtained with the aid of a virtual training environment. The signal treatment for the obtained data is explained and the output results from 10-fold cross validation are presented for different feature vector classification by a simple Kohonen layer SOM. The best result represents a 60% average recognition rate for an average processing time of about 132 seconds.

Keywords—**BMI, SOM, Graphical User Interface, Classification**

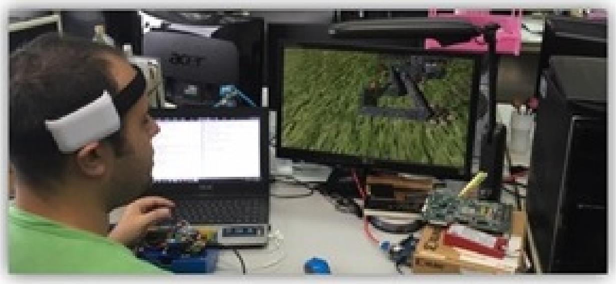

Figure 1. Brainwave pattern signal extraction environment.

1. Introduction

Recently, the use of different commercial brain wave detection devices by Electroencephalograph (EEG) has become popular for Brain Computer Interfaces. However, the use of simple commercial devices cannot perform in a precise way due to the lack of the necessary components because of their cost factor.

One of the most popular devices is the Neurosky Mindwave [1], which provides a reliable sort of outputs that require effort to be generated in a correct manner. In order to perform a signal that can be generated by multiple users and that can be processed and recognized by a computer system, a set of specific patterns and processing has been proposed[2].

While some studies focus on obtaining specific features [3], other research has been using subject identification with pass-thoughts for authentication purposes [4] with multiple [5] [6] or one electrode [7] devices with good results.

However, the previous research has been focusing on the use of raw EEG signals for feature classification and selecting specific features for classification. In this paper, the brainwave signals consist of both an Attention and Meditation vectors composed by a power which varies from 0-100% on a discrete time range of 180 seconds at 1 Hz sampling frequency. These signals are generated by a user while using a Virtual Training Environment as shown in Figure 1. Also, the Unsupervised Learning by SOM selects the features automatically.

1.1 Brain Wave Password Generation

The brain wave password consists of a series of Attention/Meditation values that need to reach a value bigger than a custom threshold (e.g.$Th = 60\%$) at a specific time in order to be executed correctly. The training environment was designed is a way that a specific combination is generated by the test users according to previous results [8]. The experiment environment provides a plot after each test is performed. A sample of a resulting plot is shown on Figure 2.

1.2 The 3D Training Environment

The proposed training environment was designed with the use of Blender for the animated models and structures as well as Unity3D for its ease of use and deployment. The environment has a series of features such as music and interaction with the experiment variables that are described as follows. The environment setup was created for the test subjects to familiarize with the objects and understand the way of interacting with them; an additional relaxation music track can be used for aiming in obtaining higher meditation behaviours.

The purpose of the training task is to guide the player (training subject) throughout the predefined path, as specified in Figure 3. When concentrated, the user can move the player forward and then move the player to the right if the user begins to enter meditation. In either case, the user can stop the

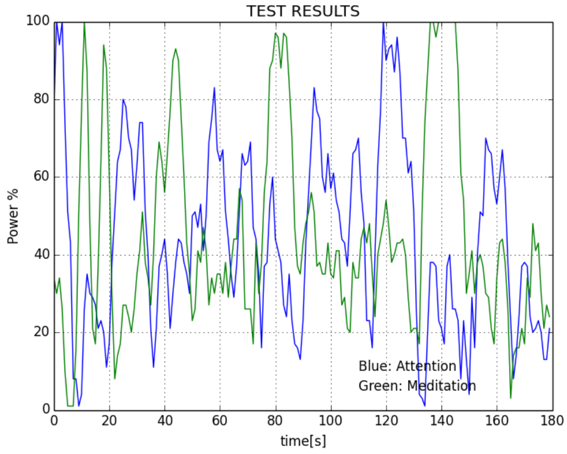

Figure 2. Example of a resulting pattern.

player with eye artifact noise.

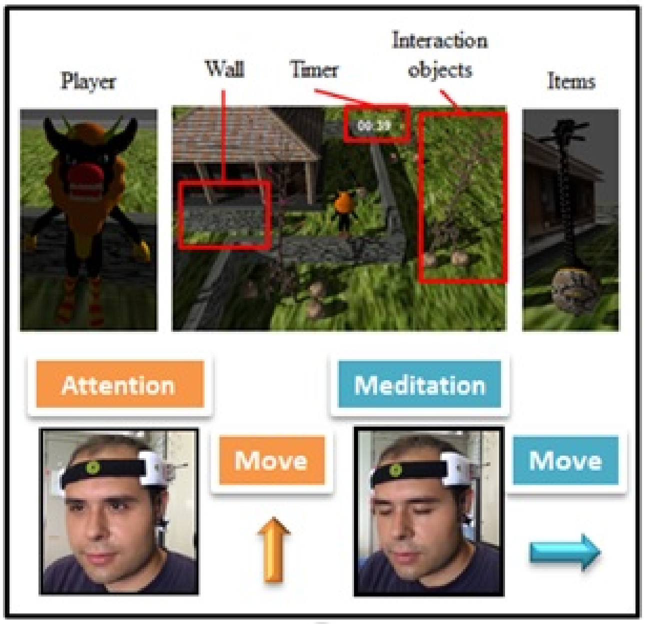

Figure 3. Environment description.

If the mind-wave signal vectors came into the expected pattern in the required time, the attempt will be classified as successful and the class +1 will be labeled for the output EPOCH. An unsuccessful attempt is recorded with the -1 label.

1.3 Feature extraction and SVM classifier

The purpose of obtaining a feature vector was to use a set of data that combines the expected Attention and Meditation signals previously obtained in order to create a good real time classifier by using Support Vector Machine (SVM). For this paper, the best feature vector is also used for the Self-Organizing Map.

From previous experiments [9], a good feature vector was found for the Supervised Learning Classifier by SVM. Say we have Attention vectors A and Meditation vectors B. Feature vector **X** from Equation 1 showed the best classification results with LinearSVC Kernel when reduced to a two dimension vector by PCA. For this case, 30 training samples and 5 test samples have been used for classification.

$$X = \phi\left[\frac{A^2(A-B)}{A+B+\phi}\right]; \phi = 0.001 \quad (1)$$

1.4 Dimensionality Reduction

In this research, whitening was applied to the feature vectors for different cases, three (PCA3), five (PCA5), seven(PCA7) and ten (PCA10) dimensions. The eigenvalue plots after applying Singular Value Decomposition [10] for the Attention A, Meditation B and Feature vector **X** suggest that at least ten dimensions can be considered as the most important number of components to perform training for the Self-Organizing Map implementation.

2. Self-Organizing Maps

2.1 The Kohonen Layer SOM

The Kohonen Layer Self-Organizing Maps provide a way of representing multidimensional data in much lower dimensional spaces, usually one or two dimensions (vector quantization) [11].

In its simplest form it produces a similarity graph of input data. It converts nonlinear statistical relationships between high-dimensional data into simple geometric relationships of their image points on a low-dimensional display. Usually, a regular two-dimensional grid of nodes is used.

Assume vector **x** = $\{\xi_1, \xi_2, ..., \xi_n\}^T \in \Re^n$. With each element in the SOM array, we associate a parametric real vector $\mathbf{m_i} = \{\mu_1, \mu_2, ..., \mu_n\}^T \in \Re^n$ that is called the model. Assuming a general distance measure between **x** and $\mathbf{m_i}$ denoted $d(\mathbf{x}, \mathbf{m_i})$, the image of an input vector **x** on the SOM array is defined as the array element m_c that matches best with **x**. In this research, the most common approach for determining the winner c, described by Equation 2 is used.

$$c = argmin_i\{d(\mathbf{x}, \mathbf{m_i})\} \quad (2)$$

Differing from the traditional vector quantization, $\mathbf{m_i}$ is to be defined in such a way that the mapping is ordered and descriptive of the distribution of **x**. For effective mapping, it will suffice that the distance measure is defined over all occurring **x** items and a sufficiently large set of models $\mathbf{m_i}$ [12].

2.2 Implementation

In order to test the capabilities of Unsupervised Learning Method for classification, a small classification test was performed on the feature vector with a dimension reduction of three by PCA [13], because it allows distinguishing the colors of each of the regions.

A 30x30 grid Kohonen layer map with Gaussian kernel and exponential learning rate was implemented for the distance function described previously. At a glance, it looks that the map is not performing well when the number of iteration is low ($n < 10000$) [14].

However, once the number of iteration has been increased ($n > 20000$), the areas are getting defined and the map seems to be formed clearly, as presented in Figure 4 (the training is independent for each picture).

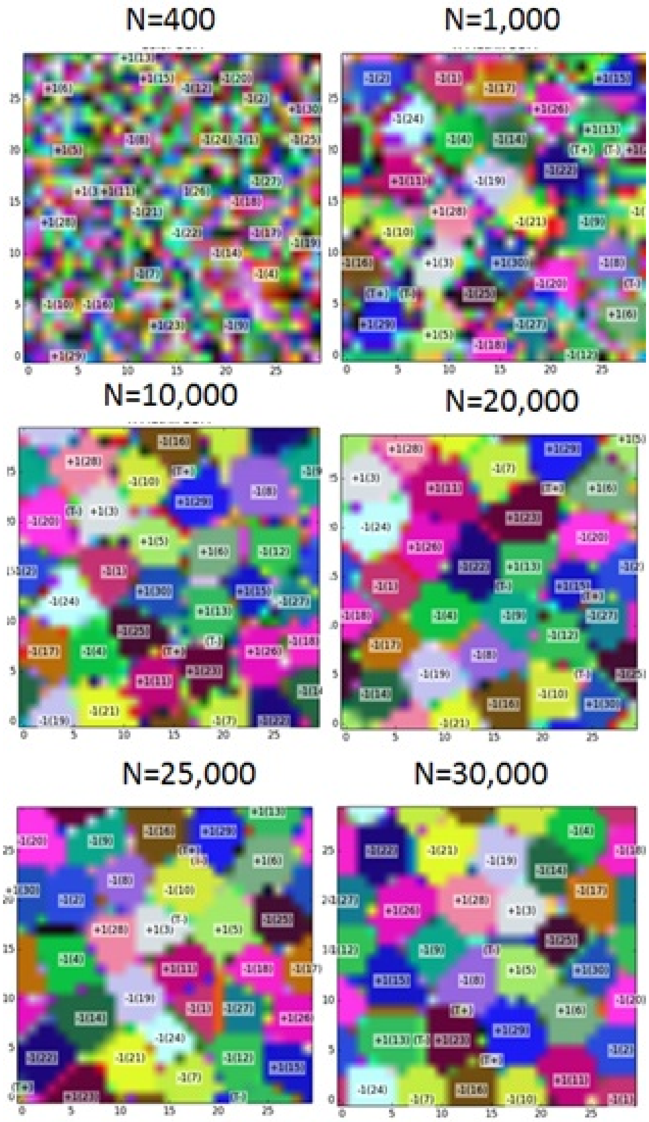

Figure 4. The Kohonen Map for feature **X** with PCA3.

Each region represents the winners for the Feature vector **X** in the training data (which is labeled +1 or -1 depending on the case). After the SOM is created from the training vectors, the test vectors (containing the tags +1 or -1 for successful/unsuccessful attempt) are mapped and compared to the output SOM in order to verify if they are being recognized correctly by SOM.

3. Results

The use of different PCA components for the main Feature vector **X** and other features composed by the Attention and Meditation values were tested with the Python library.

The average processing speed for each classification trial was performed for five iterations of the SOM in each trial. 10-fold cross validation tests were performed in order to provide the average recognition rate for the test vectors.

Some of the output map examples are shown in Figure 5 (since the use of multiple dimensions does not output a colored map, the use of blue, +1, and white, -1, tags are shown in the figure).

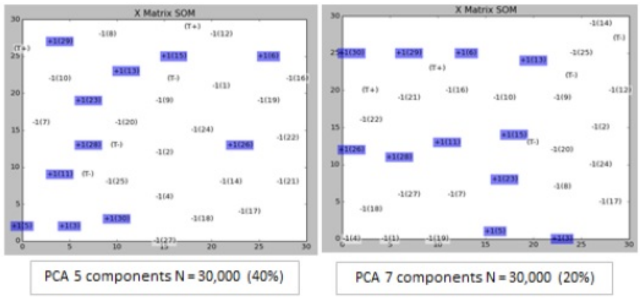

Figure 5. Examples of SOM for Feature vector **X** at 30,000 iterations (with recognition rates).

Table 1 presents these results for different number of iterations (I = 20,000, II=25,000 and III=30,000). The performance rate is the ratio between the recognition rate and the average SOM processing time.

4. Conclusions and Further Work

In this research, we have concluded from the results that Feature vector **X** provided the best classification/processing time ratio for the maximum number of components available (PCA10) for the attention/meditation brainwave patterns.

However, the fact that the recognition rate is not very high suggests that a different feature vector should be found for this methodology.

The use of the normal Python libraries suggests that there are some features that are not completely understandable; therefore, on-line implementation might not be suitable with this methodology. Also, the learning rate is drecreasing at an exponential rate, which could signify a longer processing time for the training stage. In the future, the use of larger datasets might be implemented to increase the classifier performance in terms of recognition rate.

Also, as suggested, different neighbourhood and learning functions will be implemented to reduce the amount of calculations per iteration (the use of a table for the exponential function is also taken into account for further implementation).

The presented feature vectors might not be, as well, the best option for analysis, the use of a teacher signal and other feature implementations (like AR coefficients) should also be performed.

Table 1. 10-fold cross validation results for different features and recognition rates by SOM

Feature	**PCA3**			**PCA5**			**PCA7**			**PCA10**		
A	I	II	III	I	II	III	I	II	III	I	II	III
Recognition rate	0.52	0.52	0.68	0.4	0.4	0.52	0.6	0.64	0.52	0.4	0.56	0.40
Processing time (s)	455.95	572.90	686.31	528.89	645.54	786.77	568.69	737.39	855.77	670.03	828.05	982.87
Average time(s)	91.19	114.58	137.262	105.77	129.10	157.35	113.73	147.47	171.15	134.00	165.60	196.57
Performance ratio	0.57	0.45	0.49	0.37	0.30	0.33	0.52	0.43	0.301	0.30	0.34	0.20
B	I	II	III	I	II	III	I	II	III	I	II	III
Recognition rate	0.20	0.24	0.32	0.20	0.20	0.60	0.24	0.52	0.40	0.48	0.40	0.76
Processing time (s)	474.03	595.53	709.85	567.26	688.07	837.12	594.92	745.91	882.59	662.13	833.47	966.87
Average time(s)	94.81	119.11	141.97	113.452	137.614	167.42	118.98	149.18	176.52	132.43	166.69	193.374
Performance ratio	0.21	0.20	0.23	0.18	0.15	0.36	0.20	0.35	0.23	0.36	0.24	0.39
X	I	II	III	I	II	III	I	II	III	I	II	III
Recognition rate	0.26	0.32	0.4	0.4	0.44	0.56	0.40	0.60	0.64	**0.60**	0.60	0.60
Processing time (s)	444.21	557.27	665.48	518.43	698.27	839.45	608.77	744.97	865.29	**658.90**	797.70	1027.56
Average time(s)	88.84	111.45	133.10	103.69	139.65	167.89	121.75	148.99	173.06	**131.78**	159.54	205.512
Performance ratio	0.29	0.29	0.30	0.39	0.32	0.33	0.33	0.40	0.37	**0.46**	0.37	0.29
A+B	I	II	III	I	II	III	I	II	III	I	II	III
Recognition rate	0.44	0.56	0.60	0.40	0.36	0.64	0.20	0.24	0.44	0.40	0.12	0.32
Processing time (s)	484.21	1367.57	777.77	518.97	643.56	777.77	575.89	711.74	2006	640.12	811.07	1023.04
Average time(s)	96.84	273.51	155.55	103.79	128.71	155.55	115.18	142.34	401.2	128.02	162.21	204.61
Performance ratio	0.45	0.20	0.39	0.39	0.28	0.41	0.17	0.17	0.11	0.31	0.74 E-2	0.16
A-B	I	II	III	I	II	III	I	II	III	I	II	III
Recognition rate	0.44	0.4	0.56	0.48	0.52	0.4	0.48	0.44	0.44	0.4	0.56	0.60
Processing time (s)	483.21	643.56	777.77	518.97	643.56	777.77	518.97	643.56	777.77	658.90	797.70	1027.56
Average time(s)	96.64	128.71	155.55	103.79	128.71	155.55	103.79	128.71	155.55	131.78	159.54	205.51
Performance ratio	0.45	0.31	0.36	0.46	0.40	0.26	0.46	0.34	0.28	0.30	0.35	0.29
A/B	I	II	III	I	II	III	I	II	III	I	II	III
Recognition rate	0.28	0.54	0.54	0.20	0.20	0.32	0.12	0.36	0.36	0.40	0.64	0.72
Processing time (s)	448.45	570.56	679.58	518.97	643.56	777.77	518.97	643.56	777.77	662.13	833.47	966.87
Average time(s)	89.69	114.11	135.916	103.79	128.71	155.554	103.79	128.71	155.55	132.42	166.69	193.37
Performance ratio	0.31	0.47	0.40	0.19	0.15	0.21	0.12	0.28	0.23	0.30	0.38	0.37

References

[1] NeuroSky Inc., "MindSet Communications Protocol," *Step-By-Step Guide to Parsing a Packet*, Neurosky Inc., pp.6-8, June 2010.

[2] B. Senzio-Savino, M.R. Alsharif, C.E. Gutierrez, K. Yamashita, "Path Detection in Virtual Environment for Synchronous EEG by Density Based Support Vector Machine," *Journal of Information and Communication Engineering*, Vol 1, pp.36-40, December 2015.

[3] K. Fladby, "Brain Wave Based Authetication," *Gjøvlk University College*, pp.23-26, 2008.

[4] P. Kumari, A. Vaish, "Brainwave Based Authentication System: Research Issues and Challenges," *International Journal of Computer Engineering and Applications*, Vol. IV, Issue I& II pp.89-104, December 2014.

[5] J. Klonovs, C. Kjeldgaard Petersen, H. Olesen, A. Hammershøj, "Development of a Mobile EEG-based Biometric Authentication System," *Biometric Authentication System. Paper presented at WWRF Meeting, Berlin, Germany*, pp.1-6, 2012.

[6] J. Thorpe, P.C. van Oorschot, A. Somayaji, "Passthoughts: Authenticating With Our Minds," *NSPW '05 Proceedings of the 2005 workshop on New security paradigms* , pp.45-56, April 2005.

[7] J. Chuang, H. Nguyen, C. Wang, B. Johnson, "I Think, Therefore I am: Usability and Security of Authentication Using Brainwaves," *Proceedings of the Seventeenth International Conference on Financial Cryptography and Data Security(FC'13)*, pp.5-14, April 2013.

[8] B. Senzio-Savino, K. Yamada, "Test and development of a mind wave signal pattern password application," *Proceedings of the 15th System Integration Division Annual Conference SICE2014*, pp.1-3, December 2014.

[9] B. Senzio-Savino, M.R. Alsharif, C.E. Gutierrez, K. Yamashita, "Synchronous Emotion Pattern Recognition with a Virtual Training Environment," *International Conference on Artificial Intelligence*, Proceedings, pp.650-654, June 2015.

[10] "Singular Value Decomposition with Python," *http://docs.scipy.org/doc/scipy/reference/generated /scipy.linalg.svd.html*, 2008-2014.

[11] "Self-Organizing Maps in Python," *http://www.ai-junkie.com/ann/som/som1.html*, 2006-2015.

[12] T. Kohonen, "Self-Organizing Maps," *Springer*, pp. 106-115 ,1997.

[13] "Principal Component Analysis in Python," *http://scikit-learn.org/stable/modules/generated/ sklearn.decomposition.PCA.html*, 2010-2014.

[14] Center for Cognitive Neuroscience "Multivariate Pattern Analysis in Python," *http://www.pymvpa.org/examples/som.html*, 2006-2015.