# Processing of Cerebral Cortex Neurosignals from EEG Sensors and Recognizing Specific Types of Mechanical Movements Elements of Pacient Limbs under the Cognitive Feedback Influenses

Myhaylo Petryk[a], Oleh Pastukh[a], Mykhaylo Bachynskiy[a], Ivan Mudryk[a], Volodymyr Stefanyshyn[a]

[a] *Ternopil Ivan Puluj National Technical University, 56, Ruska Street, Room 1-104 Ternopil, 46001, Ukraine*

**Abstract**

New methods of processing cognitive neurosignals of the cerebral cortex based on the analysis of digital data from EEG sensors using machine learning methods, used to model the elements of mechanical movements of the patient's limbs, are considered in this article. The results of the application of this approach consist in determining the reverse cognitive effects on the study of movement elements (thumbs of the left and right hands) as signs for recognizing specific types of movement of human limbs.

**Keywords 1**

Neural networks, cognitive signals, neuro feedback, mechanical movements of patients' limbs, analysis, big data sets, machine learning, hardware and software.

## 1. Introduction

The latest machine learning technologies based on deep neural networks in the development of signal processing information systems are improving the solution of problems related to the recognition and identification of human movements caused by cognitive influences of the nodes in the cortex of the brain. This is associated with a whole range of current medical applications, such as restoring the motor functions of people affected by various negative technological and military actions by creating effective means of prosthetics for this category of patients, treating patients with signs of a range of critical neurological disorders such as Alzheimer's and Parkinson's disease. [1]. Analysis of digital signals from nodes in the cortex of the brain (CC) is crucial for understanding the role of feedback in the cognitive control of human movements and their restoration to a normal state. The complexity of identifying the states of the human motor support mechanism (MSM) lies in the imperfection of existing diagnostic methods, their low accuracy, and the lack of mathematical and software tools for identifying the reverse influence of cognitive influences of CC nodes on their behavior. Studies of neural systems related to the analysis of patient behavior have been conducted by a number of researchers, such as Legrand A.-P., Vidailhet M., Wang J.-S., Luis E. D., Viviani P, and others [2-5]. They focused primarily on analyzing the state of MSM in patients using classical methods of digital processing based on Fourier transformation [2-4]. However, such methods require further development to ensure high-quality analysis and recognition of movements under the influence of cognitive signals from CC.

CITI'2023: 1st International Workshop on Computer Information Technologies in Industry 4.0, June 14–16, 2023, Ternopil, Ukraine
EMAIL: petrykmr@gmail.com (A. 1); olegpastuh@gmail.com (A. 2); m.bachynskyi@gmail.com (A. 3), i1mudryk@ukr.net (A. 4); volodymyr_stefanyshyn3006@tntu.edu.ua (A. 5)
ORSIDE: 0000-0001-6612-7213 (A. 1); 0000-0002-0080-7053 (A. 2); 0000-0003-4139-7633 (A. 3) ; 0000-0002-4305-1911 (A. 4); 0009-0007-2829-8995 (A. 5)

This article proposes a high-performance information technology for processing digital EEG signals from CC to study the state of the human MSM based on machine learning using deep neural networks, which allows identifying movement elements with consideration of cognitive feedback from CC nodes.

## 2. Experiment

The research idea was to use machine learning for analyzing cognitive feedback effects of SS neurosignals based on processing of encephalographic data of a patient who underwent an experiment [6], taking into account the approaches developed in [7,8]. The NEUROKOM computer electroencephalograph with a 16-channel selection of encephalograms and transmission to a personal computer using the appropriate protocol, which is the fifth generation of developed computer electroencephalography complexes, was chosen for studying the brain's electroencephalography (EEG) signals (Fig. 1).

The helmet with the installed hardware and software platform of the manufacturer was used for conditioning EEG signals and post-processing on a PC. The data was stored in both raw text and visualized representations at each point in time.

To obtain input data, the encephalograph was connected to the patient, which allowed measuring the potentials of his brain activity.

**Figure 1.** Visualization of the experiment

During the experiment, the patient sat comfortably in a chair with a backrest and armrests and performed bending movements of the index finger of their left hand for 2 minutes, followed by the same movements with the index finger of their right hand. The obtained data were analyzed using machine learning based on deep networks, which allowed for high-precision recognition of specific finger movements (which finger was flexed at a given time) and the resulting cognitive feedback effects of the neural nodes of the central nervous system, as determined by EEG signals from 16 sensors of the encephalograph.

During the experiment, the patient's movements were recorded by the encephalograph at a sampling rat of 500 Hz, which provided sufficient data. The total number of signal measurements for each of the 16 sensors of the encephalograph was 6003. Various experimental conditions were also taken into account, such as preventing sound stimuli, which reduced the risk of obtaining erroneous results. The obtained data were saved and used for further analysis and modeling using machine learning. The sets of output data are presented in Figures 2 and 3.

| | Fp1 | Fp2 | F3 | ... | F8 | T3 | T4 |
|---|---|---|---|---|---|---|---|
| 0 | 47.635586 | 99.990128 | 53.815628 | ... | 70.912117 | 60.892860 | 77.042313 |
| 1 | 227.615021 | 395.452789 | 228.134918 | ... | 295.278961 | 247.877380 | 307.193726 |
| 2 | 213.238861 | 368.382751 | 210.709122 | ... | 256.713959 | 220.542252 | 264.541565 |
| 3 | 205.526337 | 357.155548 | 204.634644 | ... | 265.179443 | 219.313721 | 267.275696 |
| 4 | 289.747101 | 479.594482 | 279.491913 | ... | 348.705597 | 297.387726 | 359.141846 |
| ... | ... | ... | ... | ... | ... | ... | ... |

**Figure 2**: The input data set from the encephalograph for the left hand's index finger (LHF).

| | Fp1 | Fp2 | F3 | ... | F8 | T3 | T4 |
|---|---|---|---|---|---|---|---|
| 0 | 233.182831 | 373.688751 | 221.000488 | ... | 226.944946 | 242.776093 | 292.732971 |
| 1 | 249.224884 | 387.722748 | 229.652008 | ... | 173.831833 | 250.884491 | 293.522827 |
| 2 | 191.165009 | 291.293518 | 174.061279 | ... | 111.312622 | 192.037201 | 221.403030 |
| 3 | 188.758698 | 292.291504 | 174.736237 | ... | 120.718735 | 193.142883 | 227.539597 |
| 4 | 229.419113 | 363.833740 | 215.723663 | ... | 182.673569 | 238.476196 | 283.923065 |
| ... | ... | ... | ... | ... | ... | ... | ... |

**Figure 3**: Input dataset of the encephalograph for the right thumb (RHF)

## 3. Machine Learning

After conducting the experiment, we obtained a significant amount of data that needed to be processed and analyzed. For this, we used the object-oriented programming language Python and specialized libraries: pandas, numpy, matplotlib.pyplot, collections, sklearn [6]. Thanks to this, we were able to build a model that can determine with 99.98% accuracy which finger was bent at a particular moment in time. The machine learning process allowed us to achieve this high accuracy and ensure the reliability of the results obtained.

### 3.1. Data preparing for machine learning

We read data from two files, "eeg_0.txt" and "eeg_1.txt", using the pd.read_csv() function from the pandas library. These files contain electroencephalogram (EEG) data from 16 sensors for two different patients who performed a finger flexion movement. Each file contains 6003 rows (number of EEG measurements for each finger) and 17 columns (each with 6002 values), where the first column contains the date and time, and the other 16 columns contain the EEG signal values for each sensor located in defined neurozones of the patient's brain cortex (encoding Fp1, Fp2, F3, etc.) (Fig. 1, 2).

Next, we used the iloc() function to select all rows and columns from 2 to the second-to-last for each table, meaning we discarded the first and last columns that do not contain EEG values.

To determine which finger corresponds to each data row, we created a new column "target", where the LHF finger is assigned the identifier "0" for table df1, and the RHF finger is assigned the identifier "1" for table df2. Therefore, we can use this column as the class label for training our machine learning model (MLM). In this case, "target" is an identifier of one of the fingers, and has a value of either 0 or 1.

Below is the dataset (table) for the LHF finger in the MLM, where the EEG values are recorded for each sensor, and the finger identifier is located in the "target" column (Figure 3).

| | Fp1 | Fp2 | F3 | ... | O2 | A0 | target |
|---|---|---|---|---|---|---|---|
| 0 | 47.635586 | 99.990128 | 53.815628 | ... | 74.143684 | 287.254150 | 0 |
| 1 | 227.615021 | 395.452789 | 228.134918 | ... | 304.247162 | 452.812836 | 0 |
| 2 | 213.238861 | 368.382751 | 210.709122 | ... | 243.288956 | 363.776917 | 0 |
| 3 | 205.526337 | 357.155548 | 204.634644 | ... | 269.405182 | 357.279694 | 0 |
| 4 | 289.747101 | 479.594482 | 279.491913 | ... | 358.493256 | 319.379272 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... |

**Figure 4**: Input data set for LHF finger in MLM

## 3.2. Visualization of electroencephalography (EEG) signals

To visualize the data, we used the matplotlib.pyplot library and created a plot that displays the values of the encephalogram for both **LHF** and **RHF** fingers during a 2-minute experiment. In the graphs (Fig. 5, 6), one can see how the encephalogram values change over time and how they differ for different fingers (LHF, RHF). This visualization helps to get a general idea of the characteristics of the data and their distribution.

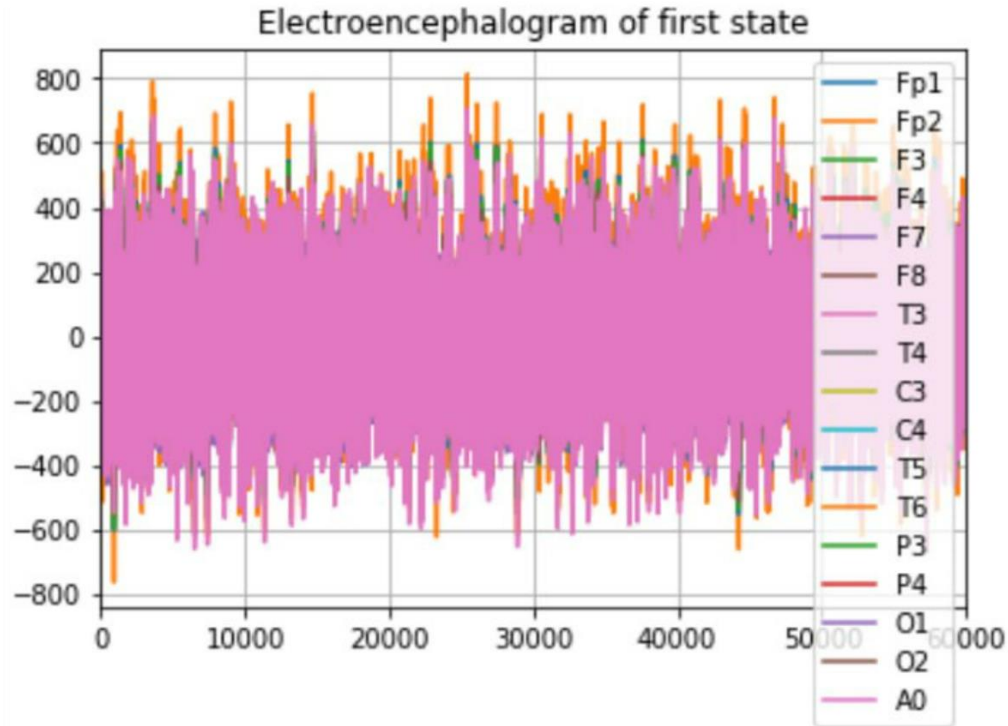


**Figure 5**: Visualization of the first set of input data for the LHF MLM model.

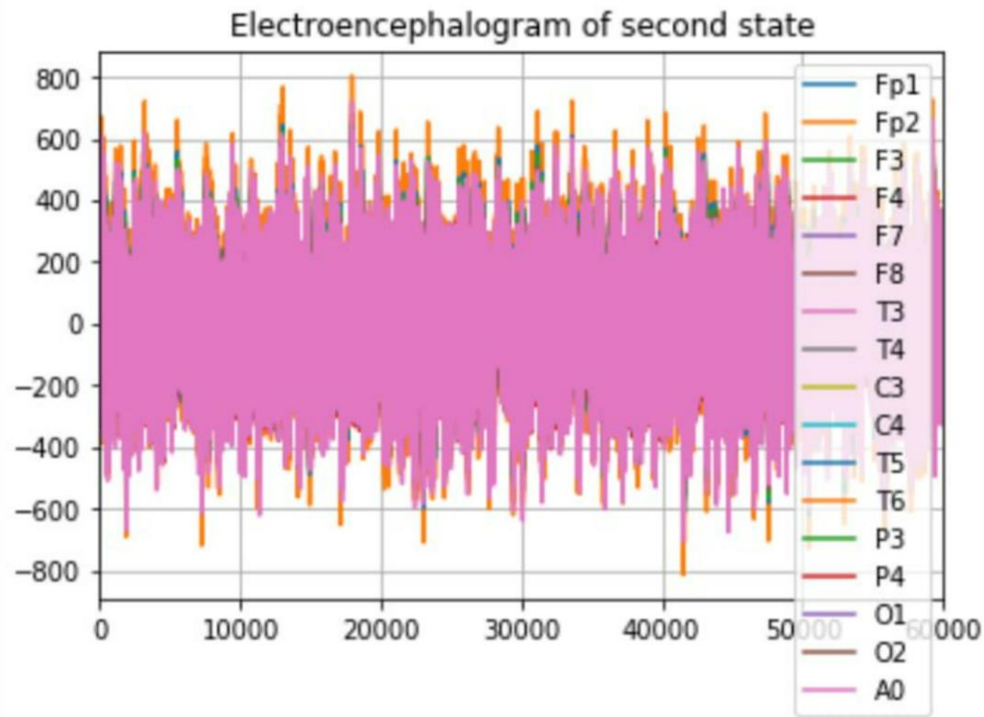


**Figure 6:** Figure 6: Visualization of the second set of MLM model for RHF

## 3.3. Data preprocessing

Before loading data into the model, it is necessary to prepare and clean it from unnecessary information. In our case, we merge the data from two sets (LHF and RHF) into one set to have more examples for training our neural network. After that, we standardize the data to have a mean of 0 and a standard deviation of 1. The result without normalization is shown in Fig. 7, and the result with normalization is shown in Fig. 8. This process ensures the uniformity of the data, which increases the convergence of the model and reduces training time.

We use the StandardScaler class from the scikit-learn library to standardize the MLM sets. After standardization, we split our MLM data set into two sets in a 75:25 ratio. 75% of the data will be used to train our neural network, and 25% will be used to test its effectiveness. We use the train_test_split class from the scikit-learn library to split the data into training and testing sets.

In addition to standardization, other operations are performed on the MLM data, such as normalization, clipping, or removing missing values. The use of different preprocessing methods affects the model results, so it is important to experiment with different approaches and choose the one that achieves the best results.

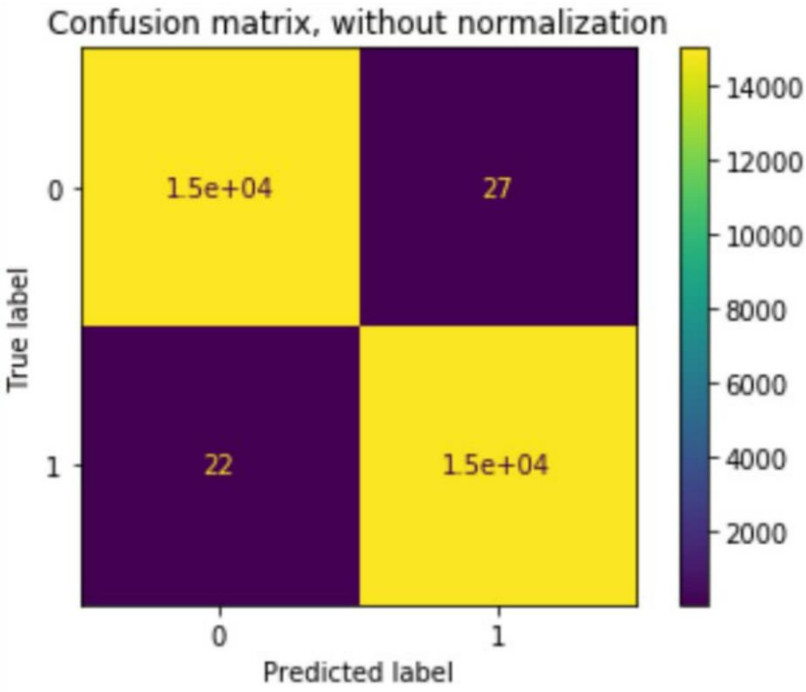


**Figure 7:** Results on non-normalized data MLM

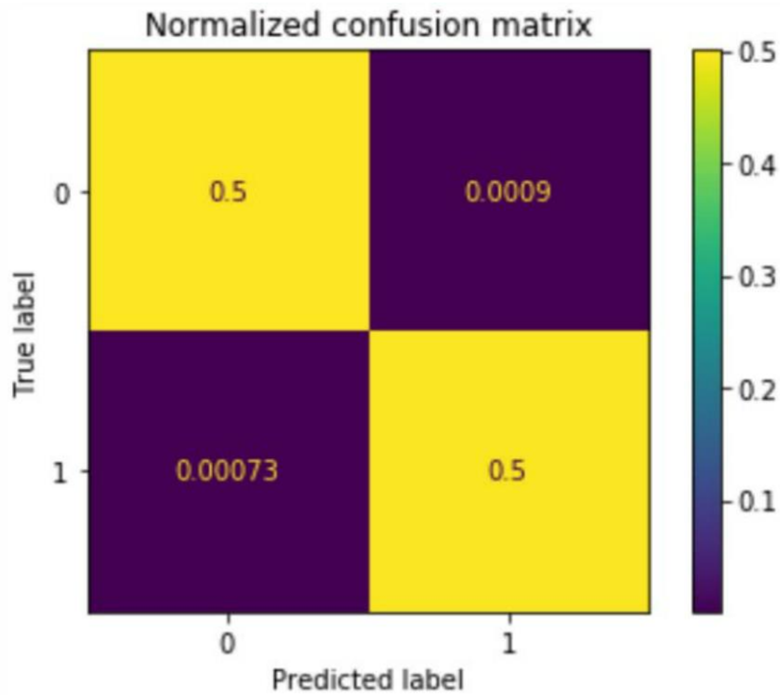


**Figure 8**: Results on normalized data MLM

## 3.4. Training and testing model

After completing the process of preparing the data for analysis and processing, we proceed with the creation and training of the neural network. This process includes the following stages: model initialization, training, prediction, and evaluation of its effectiveness.

During the training of the model, the neural network was trained using input data and backpropagation of error to establish appropriate weights between neurons. This process can take a long time, depending on the complexity of the task and the size of the data used. The code snippet used for training and testing is shown in Fig. 9.

```
#Initialization
mlp = MLPClassifier(activation='logistic')
#Training
mlp.fit(x_train, y_train)
#Prediction
y_pred = mlp.predict(x_test)
#Predict class probabilities
y_pred_prob = mlp.predict_proba(x_test)

#Compute the 'accuracy'
accuracy_score = accuracy_score(y_test, y_pred)
#Compute the 'f1-score'
f1_score = f1_score(y_test, y_pred)
#Compute the 'roc-auc'
roc_auc_score = roc_auc_score(y_test, y_pred)
```

**Figure 9**: Python code for training and testing MLM (LHF & RHF)

We obtained the following results accordingly:

- accuracy_score = 0.999133
- f1_score = 0.999121
- roc_auc_score = 0.999122.

"**Testing and evaluating the effectiveness** of the MLM (LHF & RHF) allows us to determine the accuracy of the model, which indicates how accurately it predicts the output data. Various metrics such as accuracy, f1-score, and ROC AUC can be used for this purpose. The visualization of the confusion matrix is presented in Figure 10.

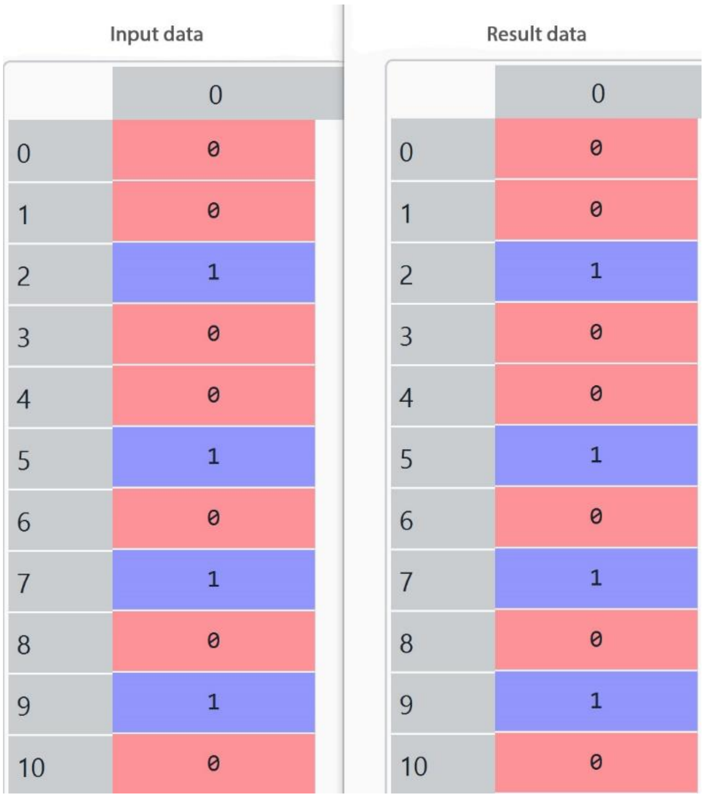


**Figure 10**: Testing on mixed data MLM (LHF & RHF). Visualization of the confusion matrix

Evaluating the performance of the model enables us to identify weaknesses and improve its accuracy. Figure 11 shows the probability estimation results of the model ranging from 0 to 1.

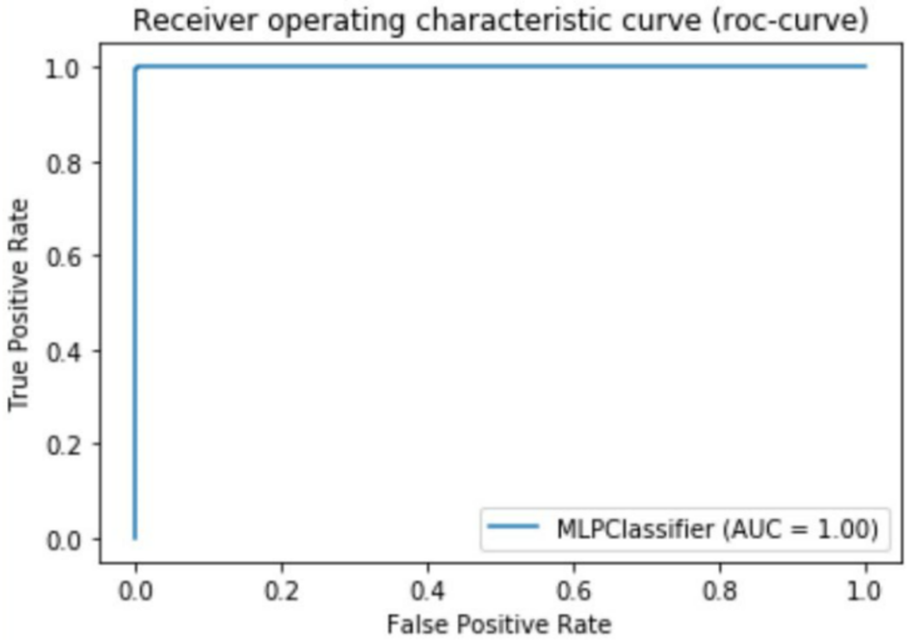


**Figure 11**: Probability estimation results of MLM (LHF & RHF) prediction.

As a result of training and testing the MLM (LHF & RHF) neural network, we achieved high accuracy, which enabled us to prepare it for use in recognizing patient movements under investigation.

## 3.5. Example of MLM for recognizing specific patient movements

The trained and tested MLM was fed a set of EEG data - signals from the patient's brain cortex (Fig. 13).

| | Fp1 | Fp2 | F3 | ... | O1 | O2 | A0 |
|---|---|---|---|---|---|---|---|
| 0 | 1.918273 | 1.954265 | 1.943454 | ... | 1.924739 | 2.035014 | 1.423534 |
| 1 | 2.050242 | 2.027658 | 2.019534 | ... | 1.739709 | 1.991673 | 1.166559 |
| 2 | 1.572614 | 1.523366 | 1.530675 | ... | 1.245908 | 1.518329 | 0.886970 |
| 3 | 1.552819 | 1.528585 | 1.536611 | ... | 1.294938 | 1.586738 | 0.997983 |
| 4 | 1.887311 | 1.902727 | 1.897050 | ... | 1.742044 | 1.928787 | 1.615408 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 5998 | -0.674928 | -0.624279 | -0.657318 | ... | -0.414106 | -0.499971 | -0.288605 |
| 5999 | 0.536146 | 0.519675 | 0.520583 | ... | 0.483025 | 0.611156 | 0.153392 |
| 6000 | 0.348343 | 0.251219 | 0.293960 | ... | 0.044674 | 0.297151 | -0.356446 |
| 6001 | 0.424987 | 0.324612 | 0.375976 | ... | 0.129893 | 0.385107 | -0.074802 |
| 6002 | 0.105723 | 0.013426 | 0.064099 | ... | -0.156115 | 0.071952 | -0.334860 |

**Figure 12:** Input EEG data set for recognizing patient movements

The output of the network indicated that the given set of input EEG signals from the brain cortex (Fig. 12) corresponded to the patient's **RHF** finger movement (Fig. 13) with an accuracy of 0.999133.

| | 0 |
|---|---|
| 0 | 1 |
| 1 | 1 |
| 2 | 1 |
| 3 | 1 |
| 4 | 1 |
| 5 | 0 |
| 6 | 1 |
| 7 | 1 |
| 8 | 1 |

a)

| | 0 | 1 |
|---|---|---|
| 0 | 1.01952e-11 | 1 |
| 1 | 1.24621e-07 | 1 |
| 2 | 3.95419e-07 | 1 |
| 3 | 3.62384e-08 | 1 |
| 4 | 3.82401e-09 | 1 |
| 5 | 0.523216 | 0.476784 |
| 6 | 0.227528 | 0.772472 |
| 7 | 0.00860331 | 0.991397 |
| 8 | 0.0680281 | 0.931972 |

b)

**Figure 13**: Results of MLM recognition of the patient's **RHF** finger movement (feature 1**):** a) illustration of the number of feature matches (**1/0**): 5998 for "**1**", 4 for "**0**", b) probability values for features (**1/0**).

Therefore, as a result of training and testing the MLM neural network (LHF & RHF), we achieved a high level of accuracy, which allowed us to prepare it for use in recognizing the investigated movements of patients.

## 4. Concluisions

The authors propose a high-performance information technology for processing digital EEG signals from the central cortex (CC) to investigate the state of the human motor system based on machine learning using deep neural networks. This approach allows for the identification of movement elements with consideration of the cognitive feedback loops of the CC neural nodes as features for recognizing specific types of human limb movements. High-performance algorithms for recognizing movement elements have been developed based on this approach, which enables parallel computing.

## 5. Acknowledgements

The research results mentioned in this work were partly supported by Grant SSHN Campus France, 2021 and Projects DI 247-22 M.P. (0122U001979), funding from the Ministry of Education and Science of Ukraine.

# 6. References


[1] Bhidayasiri R., Mari Z. Digital phenotyping in Parkinson's disease: Empowering neurologists for measurement-based care. Parkinsonism Relat Disord. 2020 Nov;80:35-40. DOI: 10.1016/j.parkreldis.2020.08.038.

[2] Legrand A.P., Rivals I., Richard A., Apartis E., Roze E., Vidailhet M., Meunier S., Hainque E. New insight in spiral drawing analysis methods – Application to action tremor quantification. J Clinical Neurophysiology, 128 (10), 1823–1834. (2017)

[3] Wang J.-S., Chuang F.-C. An Accelerometer-Based Digital Pen with a Trajectory Recognition Algorithm for Handwritten Digit and Gesture Recognition. IEEE Transactions on Industrial Electronics, 59(7), 2998-3007 (2012)

[4] Louis, E. D., Gillman, A., Böschung, S., Hess, C. W., Yu, Q., & Pullman, S. L. High width variability during spiral drawing: Further evidence of cerebellar dysfunction in essential tremor. Cerebellum, 11, 872-879 (2012).

[5] Viviani P, Burkhard PR, Chiuvé SC, dell’Acqua CC, Vindras P. Velocity control in Parkinson’s disease: a quantitative analysis of isochrony in scribbling movements. Exp Brain 2009;194:259–83.

[6] Scikit-learn, Machine Learning in Python, Pedregosa et al., Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011. URL: https://scikit-learn.org/.

[7] Petryk M., Gancarczyk T., Khimich O. Methods of Mathematical Modeling and Identification of Complex Processes and Systems on the basis of High-performance Calculations (neuro- and nanoporous feedback cyber systems, models with sparse structure data, parallel computations). Scientific Publishing University of Bielsko-Biala. Bielsko-Biala, Poland, 2021, 194 p.https://www.sbc.org.pl/dlibra/publication/584139/edition/549297

[8] Mudryk I., Petryk M.. High-performance modeling, identification and analysis of heterogeneous abnormal neurological movement’s parameters based on cognitive neuro feedback-influences. Innovative Solutions in Modern Science. 2021. New York. TK Meganom LLC. V1(45), 235-249 doi: 10.26886/2414-634X.1(45)2021.16.