Accuracy of software and hardware of computer systems for human-machine interaction

Volodymyr Stefanyshyn[1,†], Ivan Stefanyshyn[1,*,†], Oleh Pastukh[1,†], Vasyl Yatsyshyn[1,†] and Ihor Yakymenko[2,†]

[1] *Ternopil Ivan Puluj National Technical University, 56, Ruska str., Ternopil, 46001, Ukraine*
[2] *West Ukrainian National University, 11 Lvivska str., Ternopil, 46009, Ukraine*

Abstract

This article examines popular classifiers such as Bagging Classifier, Nearest Neighbors Classifier, Boosting Classifier, Support Vector Classifier for the highest performance accuracy. Classifiers will be tested for accuracy based on human brain activity data. Brain activity data collected during repetitive mechanical movements.

Keywords

neuro-interface, brain-computer interaction, parallel programming, classifier, robustness

1. Introduction

In the process of development of modern technologies, the mechanism of interaction between a person and a computer has become one of the main approaches for interaction with the world. The search and improvement of existing approaches contributes to the further development of the scientific and technological progress of society. Today, there are a large number of approaches for interaction with technical devices. One of the main principles of interaction that is used to restore missing or impaired locomotor parts of the body is the principle of using neural signals. We focused our attention precisely on the signals of the cerebral cortex. The correct interpretation of the appropriate sets of signals into specific mechanical movements is important for helping those who need prosthetic limbs or restoration of their motility.

The main goal of this study was to identify the most accurate way of processing brain signals using machine learning, high-precision computing and cloud services to parallelize the calculation process. Creation of an information system that allows determining the will of the user in mechanical movements.

[1]*BAIT'2024: The 1st International Workshop on "Bioinformatics and applied information technologies", October 02-04, 2024, Zboriv, Ukraine*

* Corresponding author.
† These authors contributed equally.
v_stefanyshyn3006@tntu.edu.ua (V. Stefanyshyn); ivan_stefanyshyn0707@tntu.edu.ua (I. Stefanyshyn); ol_pas@tntu.edu.ua (O. Pastukh); vyatcyshyn@gmail.com (V. Yatsyshyn); jiz@wunu.edu.ua (I. Yakymenko)
0009-0007-2829-8995 (V. Stefanyshyn); 0009-0008-6930-528X (I. Stefanyshyn); 0000-0002-0080-7053 (O. Pastukh); 0000-0002-5517-6359 (V. Yatsyshyn); 0000-0001-6956-5753 (I. Yakymenko)

2. Experiment

In this paper, we decided to investigate different classifiers for their accuracy in identifying correct predictions based on input data. The topic of neuro signal processing is widely used, but most of the research on this topic is poorly studied. It is important to investigate certain aspects related to the processing of human brain signals, which we will try to highlight in this article.

The data obtained during the physical experiment were used as the basis of the study, during which the subject performed repetitive mechanical movements of the fingers of the hand and turns of the hand. Brain activity data were collected during the performance of a specific mechanical movement. As a result of the experiment, it was possible to obtain 14 unique data sets, each of which was responsible for showing the will to perform one or another movement. After receiving the data, we created several models that performed programmed high-precision manipulations with the data to build a model for predicting likely volitional movements. We created several separate information systems, each of which was based on a specific classifier model. In the process of work, you used the following classifiers: Bagging Classifier, Nearest Neighbors Classifier, Boosting Classifier, Support Vector Classifier.

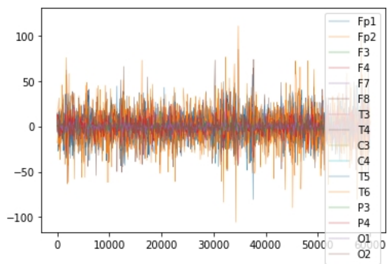

Figure 1: Visualized data of one of the packages

The purpose of this experiment was to find the best classifier that works with the highest accuracy. The results of this study are important for further research work related to the recognition of brain signals for motor-motor systems. Figure 1 shows the visualized data of one of the packages.

A person without motility disorders was chosen as the subject. A 16-channel encephalograph was used for data collection.

3. Machine learning and high-precision computing

The Python programming language was chosen for the work, as there is a wide set of libraries and modules for this language, which helps speed up the research process. The main libraries used are pandas, numpy, matplotlib.pyplot, collections, sklearn.

For optimization, cloud services were used to parallelize the calculation process. This is due to the fact that brain activity is a constant process and quite fast, so data collection must occur at a speed close to real time. In the experiment, we used a data collection frequency of 500 Hz. As a result of data collection, we received more than 6,000 units of sets of 16 sensors and we made 14 such sets. As a result, at the time of research, there are more than 84,000 lines, where each line corresponds to the values of the sensors at the corresponding moment in time.

The problem of parallelization of the calculation process is due to the fact that we need to get the resulting model as possible in the minimum time. It is cloud services that make it possible to break this process into several streams. It should also be noted that when writing data processing scripts, there is an option for automatic parallelization at the most resource-intensive stage, namely during cross validation when the hyper parameter n_jobs=-1 is specified.

4. The process of creating models

All received packets were read and converted into convenient data structures using open Python libraries. In each package, certain manipulations were carried out, such as removing redundant columns and assigning to each a unique identifier corresponding to a specific mechanical movement.

The next step was to combine the data of all packages into one dataset. The total data set was divided into training and testing data. Training data is needed so that our model is formed on their basis, and test data is needed to check the accuracy of the model's work on data with which it was not familiar until that moment.

It is important to standardize and normalize the data before starting the cross-validation process on the appropriate classifier. This will help reduce the time in the following stages and help bring all the data to a uniform appearance.

For each classifier, we defined accuracy, f1_weighted, roc_auc_ovr_weighted. To determine the overall accuracy of the corresponding classifier, we used formula 1.

$$accuracy_{general} = \frac{averag_{accuracy} + averag_{¿} + averag_{¿}}{n_{accuracy}} \tag{1}$$

For each classifier, 10 cross-validations are indicated, for the possibility of later obtaining a finer accuracy when calculating the overall accuracy of the work.

4.1. Determining the accuracy of the Bagging Classifier

Let's start with the Bagging Classifier. After cross-validating the data using the Bagging Classifier, we obtained the values of 10 cross-validations. As a result, we got a matrix of results for accuracy, f1_weighted, roc_auc_ovr_weighted (Table 1).

Table 1

Accuracy Values for the Bagging Classifier Over 10-fold Cross-Validation

fold	accuracy	f1_weighted	roc_auc_ovr_weighted
1	0.819713	0.806813	0.977443
2	0.772561	0.812020	0.969926
3	0.807273	0.818676	0.974408
4	0.814630	0.806024	0.963862
5	0.732456	0.805767	0.975457
6	0.805202	0.725127	0.975999
7	0.807452	0.764344	0.970187
8	0.763633	0.770487	0.974177
9	0.781882	0.806658	0.976452

10	0.806952	0.779895	0.965161

As a result, we got the following average values:

“f1_weighted” = 0,789581; “accuracy” = 0,791175; “roc_auc_ovr_weighted” =0,972307.

The overall accuracy for the Bagging Classifier is 0.85102174. This classifier showed quite good accuracy and has prospects for its improvement.

4.2. Determining the accuracy of the Nearest Neighbors Classifier

After cross-validating the data using the Nearest Neighbors Classifier, we obtained the values of 10 cross-validations. As a result, we got a matrix of results for accuracy, f1_weighted, roc_auc_ovr_weighted (Table 2).

Table 2

Accuracy Values for the Nearest Neighbors Classifier Over 10-fold Cross-Validation

fold	accuracy	f1_weighted	roc_auc_ovr_weighted
1	0,735992	0,728328	0,959569
2	0,817689	0,760152	0,953828
3	0,760538	0,81189	0,964756
4	0,779882	0,799885	0,957359
5	0,79981	0,804551	0,964878
6	0,773502	0,801238	0,96854
7	0,802298	0,778309	0,963378
8	0,804286	0,818221	0,964317
9	0,800893	0,801252	0,966126
10	0,811154	0,773253	0,949287

As a result, we got the following average values:

“f1_weighted” = 0,787707; “accuracy” = 0,788604; “roc_auc_ovr_weighted” =0,961203.

The overall accuracy for the Nearest Neighbors Classifier is 0.845838. This classifier showed quite good accuracy and has prospects for its improvement.

4.3. Determining the accuracy of the Boosting Classifier

After cross-validating the data using Boosting Classifier, we obtained the values of 10 cross-validations. As a result, we got a matrix of results for accuracy, f1_weighted, roc_auc_ovr_weighted (Table 3).

Table 3

Accuracy Values for the Boosting Classifier Over 10-fold Cross-Validation

fold	accuracy	f1_weighted	roc_auc_ovr_weighted
1	0,118189	0,197807	0,525179
2	0,120195	0,112457	0,526259
3	0,097865	0,102165	0,509568
4	0,199554	0,071354	0,531156
5	0,12929	0,079863	0,517634
6	0,131313	0,12465	0,527898
7	0,123457	0,092154	0,561245
8	0,094524	0,133295	0,569852
9	0,105679	0,085731	0,544824
10	0,089197	0,188205	0,555112

As a result, we got the following average values:

"f1_weighted" = 0,118768; "accuracy" = 0,120926; "roc_auc_ovr_weighted" =0,536872.

The overall accuracy for Boosting Classifier is 0.258855. This classifier showed a poor accuracy result, answering correctly only a quarter of the time. Perhaps in the future, hyper-optimization of the parameters will improve the results.

4.4. Determining the accuracy of the Support Vector Classifier

After cross-validating the data with Support VectorClassifier, we obtained the values of 10 cross-validations. As a result, we got a matrix of results for accuracy, f1_weighted, roc_auc_ovr_weighted (Table 4).

Table 4

Accuracy Values for the Support Vector Classifier Over 10-fold Cross-Validation

fold	accuracy	f1_weighted	roc_auc_ovr_weighted
1	0,750161	0,790164	0,97341721
2	0,740468	0,755488	0,97192366
3	0,828296	0,813116	0,95916285
4	0,828296	0,781236	0,96676838
5	0,828296	0,762995	0,97857253
6	0,759192	0,840314	0,97610157
7	0,772828	0,822747	0,98311636

8	0,777040	0,812994	0,96719614
9	0,764573	0,750161	0,96527489
10	0,757369	0,807463	0,96208538

As a result, we got the following average values:
“f1_weighted” = 0,793667; “accuracy” = 0,780651; “roc_auc_ovr_weighted” = 0,970361897;

The overall accuracy for the Support Vector Classifier is 0.848227. This classifier showed a poor accuracy result, answering correctly only a quarter of the time. Perhaps in the future, hyper-optimization of the parameters will improve the results.

4.5. The results of classifier accuracies

The general summaries of the results obtained by the classifiers are as follows:

- Bagging Classifier 85,1021%;
- Nearest Neighbors Classifier 84,5838%
- Boosting Classifier 25,8855%;
- Support Vector Classifier 84,8227%;

After analyzing the four most popular algorithms, we saw that three of them worked with good accuracy. Bagging Classifier turned out to be the most productive, but it is worth hyper-optimizing the parameters for Nearest Neighbors Classifier and Support Vector Classifier. There is a possibility that with certain parameters, one or another classifier can increase its accuracy for our problem. Boosting Classifier showed the lowest result when working with our problem. Figure 2 shows the normalized confusion matrix for the Bagging Classifier.

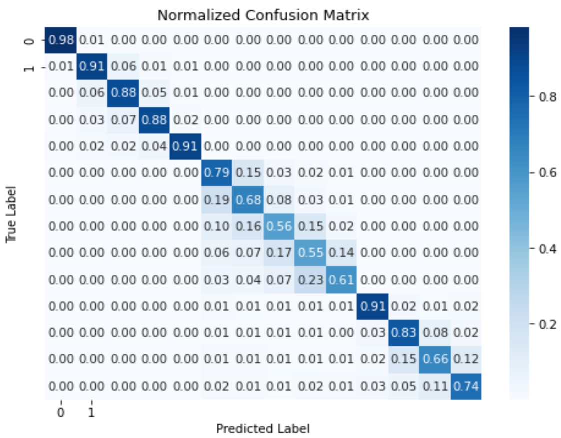

Figure 2: Normalized confusion matrix for Bagging Classifier

5. Conclusions

Research on approaches related to cognitive signal processing is important for the development of computer hardware and software technologies. Finding new ways and their research methods helps to find more accurate models for data processing. Given that all data are of a different nature, this raises the issue of finding the right models that will most accurately work in a specific situation.

As a result of our research, we saw that almost all the classifiers we considered showed good accuracy. The Boosting Classifier showed the worst accuracy, so it is not recommended for future research. In the future, it is worth optimizing the parameters to find sets of parameters for each classifier that would give the highest result.

References

[1] Musk E., Neuralink. An Integrated Brain-Machine Interface Platform With Thousands of Channels. Journal of medical Internet research. 2019. Vol. 21. No. 10.

[2] Gilja, V., Pandarinath, C., Blabe, C. H. etch. Clinical translation of a high-performance neural prosthesis. Nature Medicine. 2015.

[3] Ajiboye, A. B., Willett, F. R., Young, D. R., Memberg, W. D., etch. Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: a proof-of-concept demonstration. 2017.

[4] Pastukh O., Stefanyshyn V., Baran I., Yakymenko I., Vasylkiv V. Mathematics and software for controlling mobile software devices based on brain activity signals. The International Workshop on Information Technologies: Theoretical and Applied Problems (ITTAP-2023), 22.11.2023-24.11. 2023. 684 - 689.

[5] Medic XAI. URL: https://xai-medica.com/en/equipments.html

[6] Scikit-Learn & Joblib. URL: https://ml.dask.org/joblib.html

[7] Scikit-Learn. URL: https://scikit-learn.org/

[8] Bright brain – London's eeg, neurofeedback and brain stimylation centre. URL: https://www.brightbraincentre.co.uk/electroencephalogram-eeg-brainwaves