DOI: https://doi.org/10.36910/6775-2524-0560-2025-59-34
UDC 681.3
Stefanyshyn Ivan, Postgraduate Student
https://orcid.org/0009-0008-6930-528X
Pastukh Oleh, Dr. Sc. Prof.
https://orcid.org/0000-0002-0080-7053
Ternopil Ivan Puluj National Technical University, Ternopil, Ukraine

HIGH-PERFORMANCE COMPUTING FOR MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE IN BRAIN-COMPUTER INTERFACES WITH BIG DATA

Stefanyshyn I., Pastukh O. High-Performance Computing for Machine Learning and Artificial Intelligence in Brain-Computer Interfaces with Big Data. The article explores approaches to optimizing the processing of big data of EEG signals in BCI by combining dimensionality reduction methods and HPC. The relevance of the problem is due to the fact that modern BCIs generate large datasets of signals, the processing of which in real time often creates a critical load on hardware and software resources. The aim of the work is to establish an optimal balance between classification accuracy, model robustness, and data processing time using various dimensionality reduction methods – PCA, ICA, LDA – in combination with the MLP classifier and the Dask library for parallel calculations. A series of experiments was conducted by varying the number of components for each decomposition. It was found that when using PCA with n_components=0.999 or LDA with n_components=13, the accuracy and f1_weighted remain practically the same as in the model without dimensionality reduction, but the processing time is reduced by 1.5-4 times, depending on the settings. The use of fewer components allows for even higher performance, but is accompanied by a noticeable decrease in accuracy, which is critical for neuroengineering and rehabilitation tasks. The use of Dask for organizing parallel calculations made it possible to effectively scale experiments and avoid excessive load on individual system nodes. A comparative analysis of the accuracy, robustness, f1_weighted, roc_auc_ovr_weighted metrics and execution time showed that the optimal settings of matrix layouts allow preserving key information in the signal without significant loss of classification quality. The developed approach has proven its effectiveness for tasks where resource limitations are combined with requirements for stability and accuracy of the system in real-time mode. The practical value of the results lies in the possibility of adapting the proposed pipeline for a wide range of biomedical and engineering applications, where speed, reliability, and robustness of brain signal processing are critical.
Keywords: machine learning, electroencephalogram, motor imagery, high-performance computing, big data, information technologies, brain-computer interfaces

Стефанишин І. М., Пастух О. А. Високопродуктивні обчислення для машинного навчання та штучного інтелекту в інтерфейсах мозок-комп'ютер з великими даними. У статті досліджено підходи до оптимізації обробки великих обсягів EEG даних у BCI шляхом поєднання методів зменшення розмірності і високопродуктивних обчислень. Актуальність проблеми обумовлена тим, що сучасні BCI генерують великі масиви сигналів, обробка яких у реальному часі часто створює критичне навантаження на апаратні та програмні ресурси. Метою роботи є встановлення оптимального балансу між точністю класифікації, стійкістю моделей і часом обробки даних за допомогою різних методів зменшення розмірності – PCA, ICA, LDA – у комбінації з класифікатором MLP і бібліотекою Dask для паралельних розрахунків. Проведено серію експериментів із варіюванням кількості компонентів для кожного розкладу. Встановлено, що при використанні PCA з n_components=0.999 або LDA з n_components=13 точність і f1_weighted залишаються практично такими ж, як у моделі без зменшення розмірності, проте час обробки зменшується у 1,5-4 рази залежно від налаштувань. Використання меншої кількості компонентів дозволяє досягати ще більшої швидкодії, однак супроводжується помітним зниженням точності, що є критичним для завдань нейроінженерії та реабілітації. Застосування Dask для організації паралельних розрахунків дало змогу ефективно масштабувати експерименти та уникнути надмірного навантаження на окремі вузли системи. Порівняльний аналіз метрик accuracy, стійкості, f1_weighted, roc_auc_ovr_weighted і часу виконання показав, що оптимальні налаштування матричних розкладів дають змогу зберігати ключову інформацію в сигналі без істотної втрати якості класифікації. Розроблений підхід довів свою ефективність для задач, де обмеженість ресурсів поєднується з вимогами до стійкості й точності роботи системи в режимі реального часу. Практична цінність результатів полягає в можливості адаптації запропонованого пайплайна для широкого спектра біомедичних та інженерних застосувань, де критичними є швидкість, надійність та масштабованість обробки сигналів мозку.
Ключові слова: машинне навчання, електроенцефалограма, рухова візуалізація, високопродуктивні обчислення, великі дані, інформаційні технології, інтерфейси мозку та комп'ютера

Formulation of the problem.

BCIs are a cutting-edge technology that opens up new horizons in human-machine interaction. They are already used in medical fields, neurorehabilitation, mind control of devices, as well as in augmented and virtual reality [1, 16, 21]. Thanks to the ability to read and analyze EEG and other neural signals, these systems help people with disabilities, improve treatments for neurological disorders, and are even used in the eSports field to analyze cognitive processes.

However, the work of BCI is associated with serious challenges. One of the main ones is the processing of large volumes of data. Every second of BCI operation generates many signals that need to be analyzed quickly and accurately [2]. However, not all of this data is equally important: some contains critical information, while others can only create an unnecessary load on computing resources. Therefore, the efficiency of BCI operation largely depends on the ability to separate the necessary data from the irrelevant ones, optimizing the process of their processing[3].

The usage of methods for reducing the amount of processed data is an important step in ensuring high speed and accuracy of neural signal recognition. Important signals have priority access to computing power, while less important ones can be filtered or aggregated to reduce the overall load [4]. This approach allows not only to speed up the operation of BCI, but also to reduce power consumption and increase the overall robustness of the system.

In this article, we will consider which data is crucial for the operation of BCI, what factors determine its importance, and how to effectively reduce unnecessary data without losing the accuracy of the system. Analysis of these aspects will allow a better understanding of how to optimize computing processes and improve the performance of modern BCIs.

An analysis of the latest research and publications.

This work is a continuation of our previous research [5-8], in which we investigated the robustness, accuracy, and computational efficiency of various machine learning algorithms for EEG signal classification in BCI systems. Building on the results obtained earlier, this study focuses on the practical implementation of dimensionality reduction techniques and parallel computing tools to further improve the robustness and effectiveness of BCI algorithms when working with large-scale neural datasets. The current analysis develops the proposed methodological framework and provides a more in-depth comparative assessment of dimensionality reduction strategies under different experimental conditions.

Recent advances in data dimensionality reduction and machine learning have led to the development of a variety of methods that improve the accuracy and efficiency of classification models. Dimensionality reduction methods, such as PCA, ICA, and LDA, have become the main steps in data preprocessing in many scientific works, especially in the context of EEG signal classification and motor activity prediction [9-22]. These methods allow for the reduction of the complexity of datasets while preserving important information, which in turn improves the performance of models.

PCA is actively used in research due to its ability to reduce the number of features while preserving as much variation as possible in the dataset. For example, in a study conducted by Djelloul K. and Belkacem A.N. [9], PCA was used to classify EEG signals, which allowed for simplifying the feature space before applying classifiers such as MLP. ICA, on the other hand, is particularly useful for separating independent sources in mixed signals, which is important in neurocomputing tasks. A study by Vélez-Lora H.J. et al. [10] showed how ICA can be used to extract independent components from EEG signals, which significantly improves classification accuracy in motor imagery tasks.

LDA, which provides maximum separation between different classes, is also an important tool in the feature selection process. In the work of Kabir M.H. et al. [11], LDA was applied to select the most discriminative features before using classification algorithms, which ensures high-quality results when further training models.

One of the main directions of modern research is the integration of high-performance computing (HPC) into machine learning. HPC allows for a significant increase in the speed of processing large amounts of data and reduces the time to train models. In the study of Kabir M.H. and colleagues [11], have shown how the use of parallel computing can accelerate the process of feature extraction and classification, which is critical for real-world applications in medicine and neurocomputer interfaces. Optimization techniques, including the use of multithreading and GPU acceleration, are actively used to reduce computational costs, allowing for efficient processing of large data sets.

Optimization of computation is key to processing large data sets and improving model performance, which is especially important for practical applications in areas such as neuroengineering and biomedical signal processing. HPC significantly reduces processing time, which is important for real-world applications of models in living systems.

Therefore, recent studies emphasize the importance of using dimensionality reduction methods such as PCA, ICA, and LDA in combination with HPC to optimize machine learning. These approaches are important for solving complex tasks in the analysis of large data sets, such as EEG signal classification and motor movement prediction.

Formulation of the purpose and objectives of the research.

In previous studies, we performed calculations based on the full set of data obtained from BCIs [5-8]. This approach allowed us to achieve maximum accuracy and robustness, but also created a significant load on computing resources, which could affect the system's processing speed and overall efficiency. Up to now, we have not conducted a systematic analysis of which data are most significant for BCI operations and whether their number can be reduced without significant losses in performance.

This article aims to investigate the possibility of reducing the amount of processed data without significantly affecting the accuracy, robustness, and computation time of BCIs. We propose methods that allow us to reduce the load on the system by discarding less significant data. The selection of relevant information is based on the analysis of its impact on the results of calculations and the efficiency of the algorithms.

In this study, we will perform a comparative analysis of the obtained results, comparing the performance of the BCI algorithms when using the full amount of data with the performance after applying the optimized approach. The performance evaluation will be based on many metrics.

We will also investigate the impact of different filtering parameters on the system performance to determine the optimal settings that will minimize the loss of accuracy while reducing the amount of computation. This will allow us to form clearer conclusions regarding the possibility of using selective data processing in BCIs and offer recommendations for future research in this area.

Presenting the main material.

This study is based on a real-world experiment in which we used the NEUROKOM computer-based electroencephalograph [23] to collect EEG data during the execution of test tasks (Fig. 1). The main goal of the experiment was to obtain accurate and detailed recordings of brain activity, allowing us to better understand which signals are key to BCI operations and which can be filtered out without significant loss of accuracy.

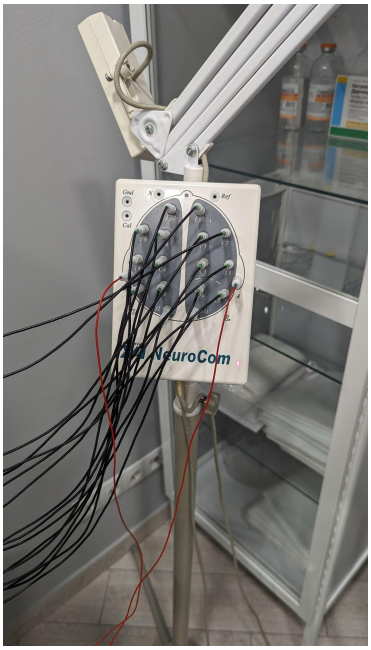

Fig. 1. Photo taken during the experiment

Several EEG data files were collected during the experiment. The following image shows an example of one such file, demonstrating characteristic patterns of brain activity during the test task (Fig. 2).

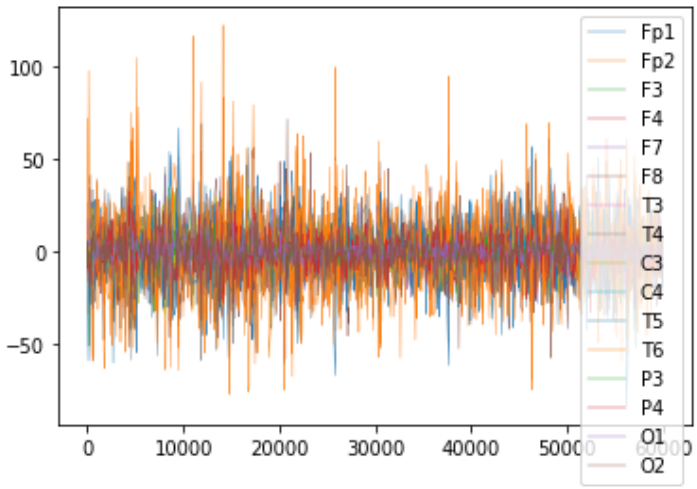

Fig. 2. Illustration of EEG signals during little finger movements

The collected data are extremely large in volume, as the brain generates a significant amount of information every second. Processing such large data sets requires significant computing resources and time. This can create a high load on the BCI, affecting its speed and robustness. That is why one of the key aspects of this research is to identify less significant data that can be filtered out without critical losses in accuracy.

Explanation of investigation.

Our algorithm is based on a classification task, where we use the MLP to analyze the collected EEG data. MLP is an effective choice due to its ability to perform parallel computations, which is especially important for working with large amounts of data, as in our case. The MLP structure allows us to calculate the weights of neurons in different layers simultaneously, using multiple computing cores, which significantly reduces the training time of the model [24].

To optimize the computations and increase the efficiency of working with large data sets, we use the Dask library. It is a powerful tool for parallelizing computations, which allows us to distribute tasks across multiple processors or even a cluster of servers [25]. One of the main advantages of Dask is that it integrates with popular scientific computing libraries such as NumPy, Pandas, and Scikit-learn, making it highly suitable for our current task.

In our experiment, to reduce the amount of data to be processed, we apply dimensionality reduction methods, which allow us to preserve important information while reducing the amount of data that needs to be processed. This is especially important when working with large datasets such as EEG.

One of the main methods we use is PCA. PCA reduces the number of dimensions in the data by identifying principal components that retain the most variability in the data. This allows us to reduce the dimensionality without significantly losing important information. With PCA, we can transform the data into new axes that represent linear combinations of the original features, thus simplifying the calculations [24].

Another important method is ICA, which focuses on finding statistically independent components. ICA is particularly useful in signal analysis, as in the case of EEG, where the signals may be mixed due to noise or artifacts. It allows us to isolate cleaner components that can be useful for classification, reducing the number of features needed [24].

LDA is another method used to reduce dimensionality with a focus on maximizing the separation between classes in the data. LDA allows us to preserve the greatest separation between classes, which makes it useful for classification tasks. This method helps not only reduce the number of features, but also improves classification accuracy by preserving important linear distinctions between classes [24].

The usage of these methods allows us to reduce the amount of data to be processed while preserving the essential information necessary for classification. They allow us to optimize the processing process and reduce the load on the system, which is important for achieving high efficiency and accuracy in processing large data sets.

For each of the dimensionality reduction methods, we will conduct three measurements with different settings for the number of components. The first measurement will be carried out with a large number of

components to preserve as much information from the data as possible. The second measurement involves the use of a small number of components, which will reduce the amount of data and reduce computational complexity, although with some loss of accuracy. The third measurement will include the optimal number of components, which will provide a balance between reducing the size of the data and maintaining sufficient accuracy for subsequent classification.

This approach will make it possible to evaluate how different settings for the number of components affect the accuracy and efficiency of models, and will also help to choose the optimal parameters for each of the methods in the context of a specific problem.

Explaining the calculation pipeline.

In our study, we use a pipeline that includes Scikit-learn, Joblib, Dask, clusters, and matrix decomposition methods.

Initially, we used Scikit-learn to build the classification model. The MLP classifier is chosen due to its ability to perform parallel computations, which allows for efficient handling of large datasets. We also apply data reduction methods such as PCA at the data preprocessing stage, which reduces the complexity of the model without losing important information [24-25].

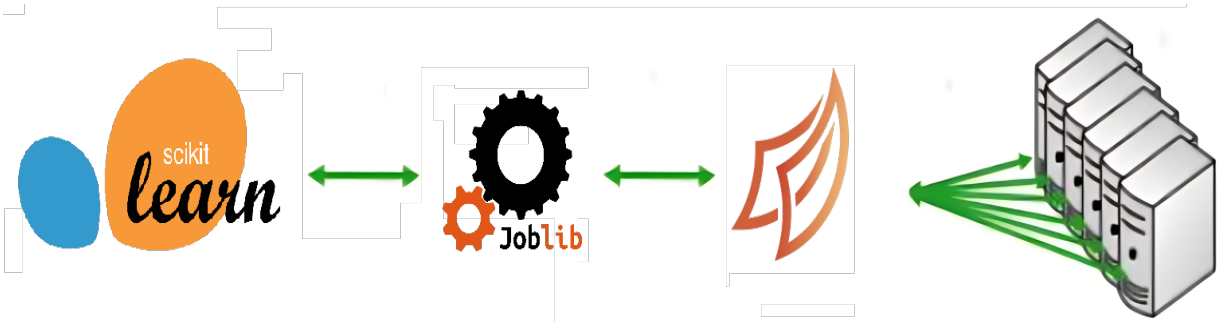

Fig. 3. Software–hardware computer calculation pipeline [25]

After training the model, we use Joblib to serialize it and save it quickly, which avoids retraining.

Dask integrates to distribute the computation across multiple cores or nodes in a cluster, which speeds up the processing of large data sets. In addition, data reduction techniques help reduce the load on the system, allowing Dask to effectively scale the training process [25].

The final stage is the use of clusters for computation, which makes it possible to scale the computation and process large amounts of data without overloading resources.

Such a pipeline allows for fast and efficient data processing, reducing the load on the system thanks to parallel computing and data reduction techniques.

Evaluation of dimensionality reduction methods.

In this section, we will examine the efficiency and performance of code implementations of data reduction techniques, such as PCA, ICA, and LDA. For each of these decompositions, three separate measurements were performed with different component values, which allowed us to evaluate their behavior under varying parameter conditions. The purpose of this analysis is not only to verify the accuracy of the models but also to determine their robustness and the time resources required to perform cross-validation.

Each of the tests includes the application of MLP with a cross-validation function consisting of 10 folds. This approach allows us to obtain 10 accuracy values for each of the tests, which are subsequently used to analyze and compare the results between different data reduction techniques. In addition, an important aspect is to determine the cross-validation computation time for each of the experiments, which helps in assessing the resource efficiency of different methods.

Metrics.

The following metrics are used to evaluate the performance of the models in this study: accuracy, robustness, f1_weighted, roc_auc_ovr_weighted, and computation time. The mathematical equations of these are as follows:

$$accuracy = \frac{TP}{TP + TN + FP + FN}$$

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$f1\_weighted = 2 * \frac{precision * recall}{precision + recall}$$

$$roc\_auc\_ovr\_weighted = \int_0^1 TPR(t)dFPR(t)$$

where *TP* = True Positive, *TN* = True Negative, *FP* = False Positive, and *FN* = False Negative, *TPR* – True Positive Rat, *FPR* – False Positive Rate [26].

Together, these metrics provide a comprehensive picture of the quality of models, determining not only their accuracy, but also their ability to adapt to different testing conditions, their performance on large data sets, and their ability to correctly classify even in complex situations with class imbalance.

Since metrics are used to evaluate models, they reflect the accuracy, classification ability, and robustness of the model. Representing these metrics as a scalar allows you to reduce the values obtained during cross-validation to a single indicator for each metric, simplifying model comparisons.

This has the advantage over the arithmetic mean, which can be sensitive to extreme values or anomalous samples. The scalar value gives a more stable and generalized assessment of the model, reducing the influence of individual folds that may differ from the general trend. Thus, using a scalar for metrics provides a more objective and accurate assessment of the model's performance.

Evaluation of PCA.

In this test, PCA was used to reduce the dimensionality of the input data. The number of components was selected based on the retained variance, namely 0.98, 0.99, and 0.999. This allowed us to investigate the effect of different degrees of data compression on the performance of the model.

With a variance of 0.98, the number of features was reduced from 16 to 3, 0.99 – 5, and 0.999 – 10. Thus, different amounts of information about the input data were retained, which affected both classification accuracy and model stability. The features obtained after decomposition were used to train the MLP, followed by 10-fold cross-validation, which enabled us to evaluate the classification accuracy for each test.

Figure 1 shows the values of the metrics f1_weighted, accuracy, and roc_auc_ovr_weighted after cross-validation for all model variants, including MLP and various PCA settings.

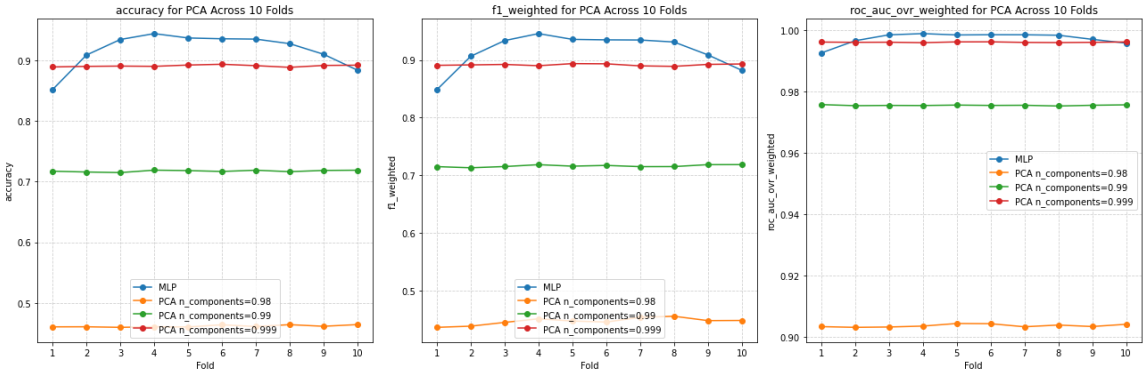

Fig. 4. MLP classification metrics depending on the number of PCA components

Table 1 shows all the final values of the metrics after the calculations, including accuracy, robustness, f1_weighted, roc_auc_ovr_weighted, and the computation time for each option. The data in the table allows us to compare the effectiveness of different approaches and choose the optimal option for a specific problem, taking into account both accuracy and processing time.

Table 1. Metric results for models using PCA

Method	MLP	PCA, n_components=0.98	PCA, n_components=0.99	PCA, n_components=0.999
Accuracy scalar	2.901777	1.459408	2.269285	2.818502
Robustness	0.027762	0.001653	0.001349	0.001464
f1_weighted scalar	2.898321	1.413019	2.264443	2.819112
roc_auc_ovr_weighted scalar	3.154061	2.857855	3.085097	3.150142
Computation time	8679.3	2040.66	2595.61	5937.73

The model without PCA showed the best results in all metrics: accuracy (2.901777), f1_weighted (2.898321), and roc_auc_ovr_weighted (3.154061). This indicates a high ability of the model to correctly classify the data. However, the computation time was the largest among all options – computation time 8679.3 seconds. The robustness of the model accuracy was 0.027762, which is a fairly good indicator of the robustness of the model with different data. When applying PCA with n_components=0.98, which preserves 98% of the variation, the accuracy decreased. The accuracy value became 1.459408, and f1_weighted decreased to 1.413019. Although the roc_auc_ovr_weighted metric decreased to 2.857855, the model remained quite stable, with reduced robustness (0.001653). The computation time was significantly reduced to 2040.66 seconds, making this option suitable when processing time is critical.

By increasing the n_components parameter to 0.99, the accuracy and robustness improved, with accuracy 2.269285 and f1_weighted 2.264443. The roc_auc_ovr_weighted value reached 3.085097. This option showed a good balance between accuracy and computation time, with computation time 2595.61 seconds. Robustness remained high, with 0.001349.

In the variant with n_components=0.999, which preserves 99.9% of the variation, the results became almost identical to the MLP without PCA. The accuracy value reached 2.818502, f1_weighted – 2.819112, and roc_auc_ovr_weighted – 3.150142. The computation time decreased to computation time 5937.73 seconds, which makes this variant the optimal compromise between accuracy, robustness, and computation time.

In general, for tasks where accuracy and robustness are critical, it is best to use PCA with n_components=0.999 or 0.99, as they give results that are close to the MLP without PCA, with reduced computation time. If the main thing is to reduce computation time with some loss of accuracy, then the option with n_components=0.98 may be acceptable, although with a noticeable decrease in results according to the f1_weighted and roc_auc_ovr_weighted metrics.

Evaluation of ICA.

In this section, we will focus on using ICA as a matrix decomposition method. We conducted three separate tests, in which the parameter n_components was assigned values of 3, 8, and 10, where the dimensionality reduction resulted in the corresponding number of classes. After applying ICA to the input data, the resulting components were fed to an MLP, which was trained and evaluated using 10-fold cross-validation.

Figure 5 shows the values of the metrics f1_weighted, accuracy, and roc_auc_ovr_weighted after cross-validation for variants using ICA, where the number of components varies from 3 to 10.

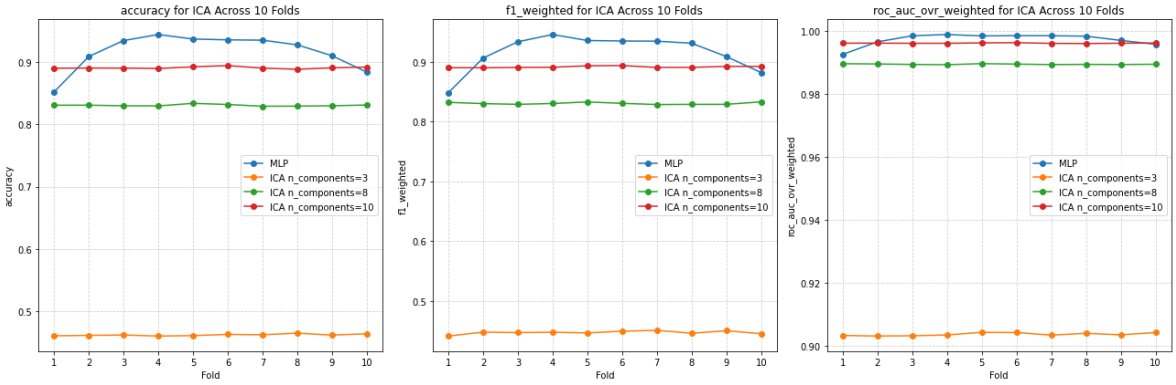

Fig. 5. MLP classification metrics depending on the number of ICA components

Table 2 shows the final values of the metrics after calculations, including accuracy, f1_weighted, roc_auc_ovr_weighted, and computation time for each variant. This data allows us to compare the effectiveness of different ICA variants and choose the most suitable one for a specific task.

Table 2. Metric results for models using ICA

Method	ICA, n_components=3	ICA, n_components=8	ICA, n_components=10
Accuracy scalar	1.459989	2.627146	2.818024
Robustness	0.001395	0.001337	0.001551
f1_weighted scalar	1.414423	2.626093	2.819056
roc_auc_ovr_ weighted scalar	2.857514	3.129153	3.15034
Computation time	1922.93	5209.5	5939.67

For the ICA variant with n_components=3, the accuracy decreased significantly: accuracy reached 1.459989, f1_weighted reached 1.414423, and roc_auc_ovr_weighted decreased to 2.857514. Despite the decrease in accuracy, the computation time was significantly shorter – 1922.93 seconds. The robustness in this variant remained at 0.001395, indicating a high sensitivity of the model to changes in the data.

When the number of components was increased to 8, the accuracy improved. Accuracy reached 2.627146, f1_weighted became 2.626093, and roc_auc_ovr_weighted reached 3.129153. The computation time of this variant was consumed_time 5209.5 seconds, which is longer than for n_components=3, but significantly shorter than for MLP without ICA. The robustness remained at a good level of 0.001337.

In the variant where 10 components are stored, the accuracy became the highest among all ICA variants. Accuracy reached 2.818024, f1_weighted was 2.819056, and roc_auc_ovr_weighted was 3.15034. This variant also demonstrated the best robustness, with a value of 0.001551. The computation time of this variant was 5939.67 seconds, which is on par with PCA at n_components=0.999.

Overall, the results show that when using ICA, the number of components significantly affects the accuracy and computation time. If accuracy and robustness are the main metrics, ICA with n_components=10 is the best option, as it gives the best results with moderate computation time. If reducing computation time is a priority, ICA with n_components=3 may be an acceptable option, although with a noticeable decrease in

accuracy. ICA with n_components=8 provides a good balance between accuracy and speed and is a good compromise for most problems.

Evaluation of LDA.

We now turn to the results obtained from using LDA. In the experiments, the parameter n_components was set to 3, 8, and 13, which corresponded to the respective number of features. The resulting data was passed to the MLP model, and classification performance was evaluated using 10-fold cross-validation.

Figure 6 shows the values of the metrics f1_weighted, accuracy, and roc_auc_ovr_weighted after cross-validation for variants using LDA, where the number of components varies from 3 to 13.

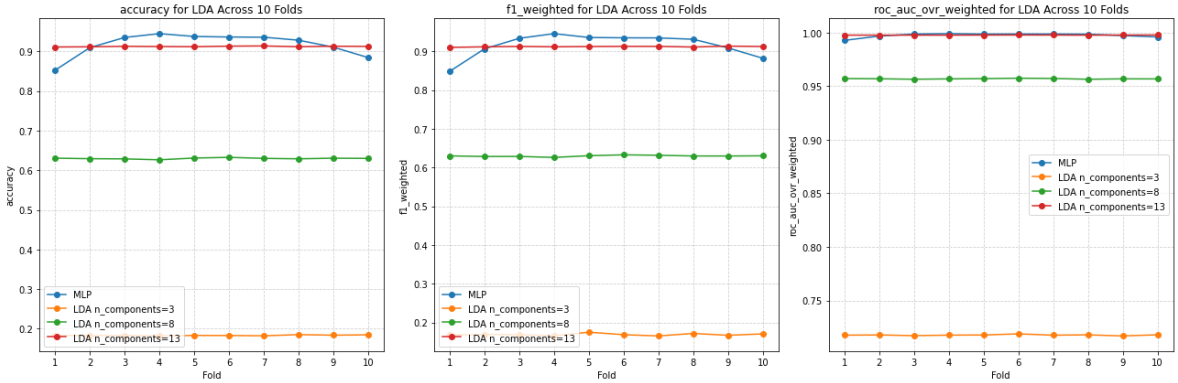

Fig. 6. MLP classification metrics depending on the number of LDA components

Table 3 shows the results for three LDA variants: with 3, 8, and 13 components. All of these variants showed different results in terms of accuracy and the ability of the model to distinguish between classes.

Table 3. Metric results for models using LDA

Method	LDA, n_components=3	LDA, n_components=8	LDA, n_components=13
Accuracy scalar	0.579981	1.991004	2.884275
Robustness	0.00098	0.001539	0.000788
f1_weighted scalar	0.534332	1.991676	2.884082
roc_auc_ovr_weighted scalar	2.270142	3.025835	3.154652
Computation time	1493.27	4816.02	5706.26

For the LDA variant with 3 components, the accuracy was significantly lower, with accuracy 0.579981, f1_weighted – 0.534332, and roc_auc_ovr_weighted decreased to 2.270142. The computation time was significantly shorter – 1493.27 seconds, which makes this variant suitable for tasks where reducing processing time is a priority. The robustness in this variant was very high, 0.00098.

Increasing the number of components to 8 allowed for improving the accuracy. Accuracy increased to 1.991004, f1_weighted became 1.991676, and roc_auc_ovr_weighted reached 3.025835. The computation time of this variant was 4816.02 seconds. Robustness remained at a good level, with 0.001539.

The variant that stores 13 components performed the best among all LDA variants. Accuracy reached 2.884275, f1_weighted – 2.884082, and roc_auc_ovr_weighted was 3.154652. The computation time was 5706.26 seconds, which is longer than in the variants with a smaller number of components. Robustness was very high – 0.000788.

The LDA method showed that with an increase in the number of components, the accuracy and robustness of the model improve significantly, although the computation time increases. The variant with n_components=13 gave the best results in terms of accuracy and robustness, making it the optimal choice for tasks where high accuracy is important. If computation time is critical, you can use n_components=8, which provides a good balance between accuracy and computation time. For very fast calculations, but with less accuracy, you can use n_components=3.

Conclusions.

According to the results obtained using different dimensionality reduction methods – PCA, ICA, and LDA, the choice of method and number of components significantly affects the performance of the model. Each method has its own advantages depending on the main goal: maximizing accuracy, optimizing computation time, or a balance between these two parameters. PCA showed the best results at higher values of n_components, with n_components=0.999 providing results closest to the baseline MLP classifier. This method is especially useful when accuracy is the main priority and computation time is of secondary importance. The model showed strong f1_weighted 2.819112, accuracy 2.818502, and roc_auc_ovr_weighted 3.150142, while the computation time was acceptable, 5937.73 seconds. However, for cases where processing time is critical, PCA with n_components=0.98 may be a better compromise, albeit with a slight loss in accuracy.

ICA with n_components=10 also showed good results in terms of accuracy, f1_weighted – 2.819056 and roc_auc_ovr_weighted – 3.15034, demonstrating a good balance between processing efficiency and model accuracy. For applications where faster results are needed without significant loss of accuracy, ICA with n_components=3 may be an acceptable choice, although accuracy is reduced.

As for LDA, this method showed moderate results, but variants with different numbers of components had a significant difference in performance. At n_components=13, LDA showed the best accuracy – 2.884275, f1_weighted – 2.884082, and roc_auc_ovr_weighted – 3.154652, indicating that a larger number of components is more beneficial for tasks where both accuracy and robustness of the model are critical. However, this is accompanied by an increased computation time – 5706.26 seconds. For tasks where computation time is important, LDA with n_components=8 can be used, which provides a good compromise between accuracy and speed, although with some loss of accuracy. In general, for most tasks where accuracy and robustness of the model are important, PCA with n_components=0.999 and LDA with n_components=13 are the optimal choices, although these methods require more computation resources. If the main concern is to reduce computation time, PCA with n_components=0.98 and ICA with n_components=3 may be acceptable, although with losses in accuracy. The ideal choice depends on the specific requirements of the problem, in particular, the balance between accuracy and computation time.

Overall, all data reduction methods have demonstrated the potential to significantly improve model robustness and reduce computing load without significant loss of accuracy. This is especially important for applications in the field of bionic devices, where stable and fast biosignal processing is a critical condition for effective human-machine interaction. The usage of data reduction methods such as PCA, ICA, and LDA can serve as an important step towards creating more adaptive and realistic next-generation BCIs.

References

1. Adolf A., Köllőd C. M., Márton G., Fadel W., Ulbert I. The Effect of Processing Techniques on the Classification Accuracy of Brain-Computer Interface Systems. Brain Sciences. 2024. Vol. 14, 1272. DOI: 10.3390/brainsci14121272.
2. Pastukh O., Stefanyshyn V., Baran I., Yakymenko I., Vasylkiv V. Mathematics and software for controlling mobile software devices based on brain activity signals. CEUR Workshop Proceedings. 2023. Vol. 3628. P. 684–689.
3. Xu F., Zheng W., Shan D., Yuan Q., Zhou W. Decoding spectro-temporal representation for motor imagery recognition using ECoG-based brain-computer interfaces. Journal of Integrative Neuroscience. 2020. Vol. 19, No. 2. P. 259–272. DOI: 10.31083/j.jin.2020.02.1269.
4. Bleuzé A., Mattout J., Congedo M. Tangent space alignment: Transfer learning for Brain-Computer Interface. Frontiers in Human Neuroscience. 2022. Vol. 16. DOI: 10.3389/fnhum.2022.1049985.
5. Stefanyshyn I., Pastukh O., Stefanyshyn V., Baran I., Boyko I. Robustness of AI algorithms for neurocomputer interfaces based on software and hardware technologies. CEUR Workshop Proceedings. 2024. Vol. 3742. P. 137–149.
6. Stefanyshyn V., Stefanyshyn I., Pastukh O., Yatsyshyn V., Yakymenko I. Accuracy of software and hardware of computer systems for human-machine interaction. CEUR Workshop Proceedings. 2024. Vol. 3842. P. 178–183.
7. Stefanyshyn V., Stefanyshyn I., Pastukh O., Kulikov S. Comparison of the accuracy of machine learning algorithms for brain-computer interaction based on high-performance computing technologies. Scientific Journal of the Ternopil National Technical University. 2024. No. 3 (115). P. 82–90. DOI: 10.33108/visnyk_tntu2024.03.082.

8. Bryk O., Stefanyshyn I., Stefanyshyn V., Pastukh O. Robustness evaluation of machine learning algorithms for neurocomputer interface software using distributed and parallel computing. Computer Systems and Information Technologies. 2024. No. 2. P. 82–88. DOI: 10.31891/csit-2024-2-11.
9. Djelloul K., Belkacem A.N. EEG Classification-based Comparison Study of Motor-Imagery Brain-Computer Interface. Proceedings - 2021 IEEE International Conference on Recent Advances in Mathematics and Informatics, ICRAMI 2021. 2021. DOI: 10.1109/ICRAMI52622.2021.9585902.
10. Vélez-Lora H.J., Méndez-Vásquez D.J., Delgado-Saa J.F. Classification of imaginary motor task from electroencephalographic signals: A comparison of feature selection methods and classification algorithms. Revista Mexicana de Ingenieria Biomedica. 2018. Vol. 39, No. 1. P. 95–104. DOI: 10.17488/RMIB.39.1.8.
11. Kabir M.H., Akhtar N.I., Tasnim N., Miah A.S.M., Lee H.-S., Jang S.-W., Shin J. Exploring feature selection and classification techniques to improve the performance of an electroencephalography-based motor imagery brain–computer interface system. Sensors. 2024. Vol. 24, No. 15. DOI: 10.3390/s24154989.
12. Kok C.L., Ho C.K., Aung T.H., Koh Y.Y., Teo T.H. Transfer learning and deep neural networks for robust intersubject hand movement detection from EEG signals. Applied Sciences (Switzerland). 2024. Vol. 14, No. 17. DOI: 10.3390/app14178091.
13. Lu Y., Wang W., Lian B., He C. Feature Extraction and Classification of Motor Imagery EEG Signals in Motor Imagery for Sustainable Brain–Computer Interfaces. Sustainability. 2024. Vol. 16, No. 15. Article number: 6627. DOI: 10.3390/su16156627.
14. Yu H., Baek S., Lee J., Sohn I., Hwang B., Park C. Deep Neural Network-Based Empirical Mode Decomposition for Motor Imagery EEG Classification. IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2024. Vol. 32. P. 3647–3656. DOI: 10.1109/TNSRE.2024.3432102.
15. Khan R. A., Rashid N., Shahzaib M., Malik U. F., Arif A., Iqbal J., Saleem M., Shahbaz Khan U., Tiwana M. A novel framework for classification of two-class motor imagery EEG signals using logistic regression classification algorithm. PLoS ONE. 2023. Vol. 18, No. 9. e0276133. DOI: 10.1371/journal.pone.0276133.
16. Saibene A., Caglioni M., Corchs S., Gasparini F. EEG-Based BCIs on Motor Imagery Paradigm Using Wearable Technologies: A Systematic Review. Sensors. 2023. Vol. 23, 2798. DOI: 10.3390/s23052798.
17. Kuo-Kai Shyu, Szu-Chi Huang, Kai-Jen Tung, Lung-Hao Lee, Po-Lei Lee, Yu-Hao Chen. Common Spatial Pattern and Riemannian Manifold-Based Real-Time Multiclass Motor Imagery EEG Classification. IEEE Access. 2023. Vol. 11. P. 139457–139464. DOI: 10.1109/ACCESS.2023.3340685.
18. Rashmi S., Ashok V. Deep feature extraction for EEG signal classification in motor imagery tasks. In: Applied Artificial Intelligence: A Biomedical Perspective. 2023. P. 253–265. DOI: 10.1201/9781003324430-19.
19. Rosenfelder M. J., Spiliopoulou M., Hoppenstedt B., Pryss R., Fissler P., della Piedra Walter M., Kolassa I.-T., Bender A. Stability of mental motor-imagery classification in EEG depends on the choice of classifier model and experiment design, but not on signal preprocessing. Frontiers in Computational Neuroscience. 2023. Vol. 17. Article 1142948. DOI: 10.3389/fncom.2023.1142948.
20. Ketu S., Mishra P.K. Hybrid classification model for eye state detection using electroencephalogram signals. Cognitive Neurodynamics. 2022. Vol. 16, No. 1. P. 73–90. DOI: 10.1007/s11571-021-09678-x.
21. Dumitrescu C., Costea I.-M., Semenescu A. Using brain-computer interface to control a virtual drone using non-invasive motor imagery and machine learning. Applied Sciences. 2021. Vol. 11, 11876. DOI: 10.3390/app112411876.
22. Saeidi M., Karwowski W., Farahani F. V., Fiok K., Taiar R., Hancock P. A., Al-Juaid A. Neural decoding of EEG signals with machine learning: A systematic review. Brain Sciences. 2021. Vol. 11, 1525. DOI: 10.3390/brainsci11111525.
23. XAI-MEDICA. URL: https://xai-medica.com/en/equipments.html.
24. Scikit-learn. API Reference. URL: https://scikit-learn.org/stable/api/index.html.
25. Dask. Scikit-Learn & Joblib. URL: https://ml.dask.org/joblib.html.
26. Vakili M., Ghamsari M., Rezaei M. Performance analysis and comparison of machine and deep learning algorithms for IoT data classification. arXiv preprint. 2020. DOI: 10.48550/arXiv.2001.09636.