On Open Workflows for Processing of Standardized Electroencephalography Data

Roman Mouček[a] and Filip Kupilík
Department of Computer Science and Engineering, Faculty of Applied Sciences, University of West Bohemia, Univerzitnı 8, Plzen, Czech Republic

Keywords: Deep Learning, EEG Data Standards, EEG Workflows, EEG Pipelines, Electroencephalography, Event-related Potentials, Human Brain, Machine Learning, Reproducibility.

Abstract: With increasing amounts of experimental data, openness, fairness, and reproducibility of scientific experimental work have become important factors for researchers, journals and funding bodies. However, these kinds of challenges are not easily and directly achievable. The goal of this paper is to contribute to these efforts by introducing advances in building more mature lifecycle of electroencephalography/event-related potential data. The progressive data standardization initiatives, data formats, and trends in using machine and deep learning methods for processing of domain data are described and discussed. An open processing workflow based on the analysis of current software tools for preprocessing, processing and classification of electroencephalography/event-related potential data is proposed, implemented and verified on a publicly available dataset.

1 INTRODUCTION

The electrical activity of the human brain has been investigated, among other, by the methods and techniques of electroencephalography (EEG) and event-related potentials (ERPs) when brain data are collected and processed to answer the questions of broad scientific interest.

The EEG method has many advantages: affordability, non-invasiveness, routine examination protocols, and the opportunity to measure spontaneous activity. However, it also has a significant disadvantage, which is evident in scientific experiments. The resulting picture of brain activity (the EEG signal) is very rough since it represents a huge number of sources of neuronal activity. It is challenging to derive the corresponding neurocognitive processes from the measured brain activity.

ERPs are changes in brain activity that are time-locked to particular events (stimuli). Generally, they have a very small amplitude (up to tens of microvolts) and can be assessed in small time windows (tens or hundreds of milliseconds).

When performing EEG/ERP experiments, it is essential to run a laboratory with appropriate equipment for event (stimuli) presentation, collect EEG/ERP data, analyze, and interpret them.

Experimental work, data processing and analysis, interpretation, as well as subsequent publishing of findings, have been traditional activities that dominated the lifecycle of EEG/ERP data. However, during the last decade, these activities were extended with the requirements for the long-term effectiveness and efficiency of experimental work. Requirements for openness, fairness, reproducibility, and citability of important data lifecycle artefacts have arisen and started to be demanded by respected journals and funding bodies.

It brought an essential change to the culture of scientific work (not only) in this field. Open, well-annotated, standardized, and shared EEG/ERP data and well-described and publicly shared procedures and workflows have slowly but continuously entered experimental work and the entire EEG/ERP data lifecycle. These approaches in a discipline of neuroinformatics started to influence traditional scientific fields related to neural systems investigations. Neuroscience, cognitive sciences, or neurolinguistics are typical examples of them.

The goal of this paper is to contribute to the openness and reproducibility of the experimental work in EEG/ERP research by making a step in defining open workflows for processing standardized EEG/ERP data. More specifically, the achievable goal is to in-

[a] https://orcid.org/0000-0002-4665-8946

Mouček, R. and Kupilík, F.
On Open Workflows for Processing of Standardized Electroencephalography Data.
DOI: 10.5220/0010345006670676
In *Proceedings of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2021) - Volume 5: HEALTHINF*, pages 667-676
ISBN: 978-989-758-490-9

tegrate a selected set of machine learning (ML) and deep learning (DL) methods (suitable for EEG/ERP data processing) with open-source software tools developed to process standardized EEG/ERP data.

The paper is organized as follows. The state of the art section presents a traditional EEG/ERP data lifecycle and its possible improvements, FAIR data principles, electrophysiological data standardization initiatives, and standardized data and metadata formats used in the EEG/ERP domain. Then current trends in using ML/DL methods to process the EEG/ERP data are introduced, and popular ML/DL libraries containing these classification methods are mentioned.

Subsequently, software tools suitable for the analysis of standardized EEG/ERP datasets and their ability to work together with ML/DL libraries are presented. As the result of the comparison of these software tools, the feasibility of the selected set of tools and libraries creating a workflow (pipeline) for EEG/ERP standardized data processing, is verified by implementing (reproducing) an analysis based on the existing and already published EEG/ERP dataset; the replicated workflow is presented. The last session concludes the obtained findings.

2 STATE OF THE ART

In this section we introduce a traditional EEG/ERP experiment, related data lifecycle, FAIR data principles, current electrophysiological data standardization initiatives, standardized data and metadata formats used in the EEG/ERP domain, and ML/DL methods suitable to process the EEG/ERP data.

2.1 EEG/ERP Experiment

A traditional EEG/ERP experiment and related data lifecycle include the following steps and activities:

- research planning - includes experimental design and its implementation respecting the advantages and disadvantages of the EEG/ERP method and its general principles, rules and strategies, approval of the experiment design by the ethics committee, the document of informed consent, and lab setting,
- experimental work - includes preparation of the participants for an experiment and instructions given to them, performance of the experiment protocol, visual inspection of the course of the experiment and its outputs, and collection and storage of the EEG signal, event markers, and related metadata,
- data preprocessing - includes referencing, channel selection, filtering, ERP epochs extraction, artifacts detection, their removal or subtraction, baseline correction, and epochs averaging,
- data processing and analysis - includes grand averaging, feature extraction, detection (classification) of ERP components (power spectral analysis in case of pure EEG recordings) and their further detailed investigation,
- data visualization and interpretation - usually include visualization of channel spectra, topographical maps, tabular and graphical representation of averages and grand averages on various electrodes for selected stimuli, periodograms, classification results, and statistical interpretation of the findings,
- research publication - includes the publication of research findings (experimental design, statistics related to participants, found ERP components or EEG frequencies), and their interpretation.

2.2 Traditional Lifecycle of EEG/ERP Data

The traditional EEG/ERP data lifecycle does not attribute too much value to the collected data, quality of the metadata set, and reproducibility of the entire experiment. Its typical features are that collected data and metadata are stored in proprietary formats, tabular forms, simple files, or even various paper forms. There is usually little effort to preserve such data and metadata longer than for the time required to interpret and publish experiment findings in a scientific journal. There is also little interest in data and metadata sharing and transforming data and metadata into well-organized structures. Specific processing algorithms and workflows are usually described by referencing the used software tools. If a non-standard, custom, or proprietary processing tool is used, the processing workflow (pipeline) is hardly reproducible.

To cope with the troubles of the experimental work and drawbacks of the traditional EEG/ERP data lifecycle and aim for their improvements, opportunities to develop open and standardized workflows that are based on common EEG/ERP preprocessing/processing methods enriched/integrated with ML/DL classification methods are addressed in this paper.

2.3 FAIR Data Principles

The general effort to enhance the openness and reproducibility of research and reusability and efficient management of raw and derived data led to the design and joint endorsement of a concise and measurable set of principles referred to as the FAIR Data Principles. These principles for scientific data management and stewardship were first formally published in Scientific Data (Wilkinson et al., 2016). The publication explained the rationale behind them, gave some initial implementation, and significantly influenced the view of scientific data from a long-term point of view.

The FAIR principles, in more detail also explained in (Wilkinson et al., 2016), are the following ones: findability, accessibility, interoperability, and reuse of digital data. They put specific emphasis on enhancing the ability of machines to automatically find and use the data, in addition to supporting its reuse by individuals (Wilkinson et al., 2016). Since the interpretation of the FAIR principles started to be various, several original authors revisited them in (Mons et al., 2017). These FAIR principles are highly reflected in the standardization efforts introduced below.

2.4 Standardization Initiatives

The absence of unified, extensible, open, and widely accepted descriptive structures, models, and formats for electrophysiology data has been naturally discussed in an international forum with the coordinating role of International Neuroinformatics Coordinating Facility - INCF (International Neuroinformatics Coordinating Facility, 2020). Except for proprietary formats used for simple storage of electrophysiological data, there currently exist several competing proposals of flexible data structures, terminologies, and formats for annotation, organization, and long term storage of electrophysiological data and metadata.

The common denominator of these proposals is that they follow the FAIR principles and have been inspired by open and proprietary data structures and formats used in electrophysiology, neurophysiology, electroencephalography, and bioinformatics.

All of them also have to cope with contradictory requirements: the proposed data/metadata descriptions should be, on the one hand, harmonized and, if possible, standardized to convey a global descriptive terminology and support the reproducibility of scientific data, procedures, and results. On the other hand, they should be flexible enough to meet the requirements of individual laboratories. However, even in the case of the standards endorsed by INCF, these are still in working progress when concerning the maturity of the whole ecosystem – the entire lifecycle of electrophysiology data has not been covered so far. This fact also makes more extensive use of existing standards difficult.

Three global initiatives are currently worth mentioning. The descriptive structures they have proposed usually go beyond the initially intended domain boundaries and can be considered to be used and extended for the description and long-term storage of diverse physiological data. The outcome of the first initiative, proposed by the German National Node for Neuroinformatics (G-node), is the Neuroscience Information Exchange format (NIX) enriched with odML terminology for the description of metadata (Zehl et al., 2016).

Neurodata Without Borders (NWB) (Teeters et al., 2015) (Rübel et al., 2019) data structures were introduced by the University of California and further elaborated by the scientific community in the U.S. The BIDS structure (Brain Imaging Data Structure) (Gorgolewski et al., 2016) introduced at the McGill University in Montreal has been later extended with the descriptive structures for EEG data (Pernet et al., 2019).

In addition to these three global initiatives two data formats are broadly used for storage of EEG recordings: the European Data Format (EDF/EDF+) and the BrainVision Core Data Format. The latter one defines the structure of data in three files: a text header file containing metadata, text tag file containing information about events (stimuli) in data, and binary data file containing raw EEG data and other signals recorded together with EEG. All these files must be stored in one folder; the header and data files are required. The header and tag files have a fixed structure (key/value pairs) and their complete description, including a description of all usable keys, can be found in (Brain Pruducts, 2019).

European data format (EDF) and especially European data format 'plus' (EDF+), have been considered as standard formats for the exchange of physiological data since 2003. EDF+ is a more flexible but still simple format which is compatible to EDF except that an EDF+ file may contain interrupted recordings. When compared to EDF, EDF+ can not only store annotations but also electromyography, evoked potentials, electroneurography, electrocardiography and many more types of investigations. (Kemp and Olivan, 2003)

Due to the robustness and widespread use of European data format and BrainVision Core Data Format 1.0, the BIDS project recognizes them as the recommended standards for storing EEG/iEEG data (Brain Products, 2020).

2.5 Machine and Deep Learning for EEG/ERP Data Processing

As it has happened in many scientific disciplines, ML and especially DL methods invaded also the EEG/ERP domain. The question if DL methods truly present advantages as compared to more traditional EEG processing approaches remains an open question according to a systematic review of DL-based method for analysis of electroencephalography data that was published in (Roy et al., 2019).

Effective and efficient EEG/ERP data processing as part of the whole EEG/ERP data lifecyle suffers from several limitations. EEG has a low signal-to-noise ratio (SNR) as the measured brain activity is often hidden under multiple sources of noise of similar or greater amplitude called 'artifacts'. Various techniques are used to minimize the impact of noise sources and extract brain activity from the EEG signal. EEG as a non-stationary signal is also highly variable across time. Then classifiers trained on a temporally-limited amount of data might provide poor generalization to data recorded at a different time on the same individual. High inter-subject variability also limits the usefulness of EEG applications. (Roy et al., 2019)

To cope with the issues mentioned above it seems to be reasonable to extend traditional and domain-specific EEG/ERP workflows with DL methods to clean, extract features and classify EEG/ERP data. Further automation of these workflows can be done by utilizing existing data standards, using tools for EEG/ERP data processing based on these standards and integrating them with existing ML/DL libraries.

According to (Craik et al., 2019) it is not yet verified whether DL methods can achieve better results without use of any pre-processing method. However, it is surprising that in more than a quarter of the experiments examined, the artifacts were removed manually. This, in addition to being time consuming, also makes difficult to reproduce the used procedures.

Feature extraction is one of the most challenging steps in the traditional EEG/ERP data processing. Elimination of the feature extraction procedure by deep neural networks is the main goal of many experiments. For example, spectrograms commonly used to visualize EEG data are used as input to convolutional neural networks (CNNs), whose performance has proven successful mainly in image data recognition. Furthermore, it was found that CNNs, which accept pure EEG signal values for input, performed on average better than with spectrograms or calculated features (Craik et al., 2019). This finding contradicts the idea that feature extraction is an important step in improving the success of the EEG data classification (Craik et al., 2019).

The fundamental decision during the application of deep-learning methods is the correct selection of the neural network architecture. For the ERP classification, CNN is surprisingly the most used architecture, followed by a recurrent neural network (RNN). Since 2015, there has been a large increase in the use of CNNs, contrary to expectations that, due to the temporal nature of the EEG signal, RNNs will be used more than models that do not naturally work with time dependencies. The expansion of the use of CNN for the classification of EEG data can be explained by the success achieved in computer vision and the use of a hierarchical structure of the data, as well as recent discussions and findings concerning the effectiveness of CNNs for time series processing. However, the frequency of the RNN use is also increasing. (Roy et al., 2019)

In addition to correct selection of the neural network architecture, researchers must take a decision about the number of layers. According to (Roy et al., 2019), most projects used less than five layers for their models, and it is possible to conclude that shallower neural networks are currently more suitable for EEG data. In (Schirrmeister et al., 2017), they directly addressed the number of CNN layers used for EEG data and concluded that shallower CNNs successfully outperformed deeper CNNs.

In the traditional EEG data classification process, the methods for feature extraction and then traditional ML methods such as LDA and SVM are often used. Almost all experiments in (Roy et al., 2019), over which the successes of DL methods to traditional ML methods was compared, showed that DL methods improved the classification results.

Currently, the generally well-known and most popular libraries providing ML/DL methods, scikit-learn in case of traditional ML methods, and Keras (based on the TensorFlow library) and PyTorch in case of DL methods, are also used for EEG/ERP data processing.

3 SOFTWARE TOOLS FOR EEG/ERP DATA PROCESSING

The traditional EEG/ERP data lifecycle includes reading data and pre-/processing them before their classification. Currently, many various software tools allow us to do these steps. These tools differ in many aspects, e.g., in the set of data pre-/processing methods provided or the data formats they can read or write to. Some tools provide a user interface; others are offered as libraries. They also differ in their ability to perform/cooperate/to be integrated with ML/DL methods and related libraries.

The next subsection describes the tools available for the pre-/processing of EEG/ERP data and their current abilities to work/interact with ML/DL methods and libraries. Since our laboratory primarily stores EEG/ERP data in the BrainVision format and the BIDS standard recognizes this format as a domain standard, the software tools that can naturally read/write data in this format are a bit favoured.

3.1 Software Tools Overview

Brainstorm (The Brainstorm team, 2020) is an open-source tool for the analysis of brain recordings supporting the BrainVision, EDF and NWB formats. It provides methods for processing EEG data, such as automated artefact detection, epoch extraction, time-frequency transformations, and SVM and LDA classifiers. Figure 1 shows the UML component diagram providing the data formats, analytical tools and ML libraries the Brainstorm can work with. FieldTrip is

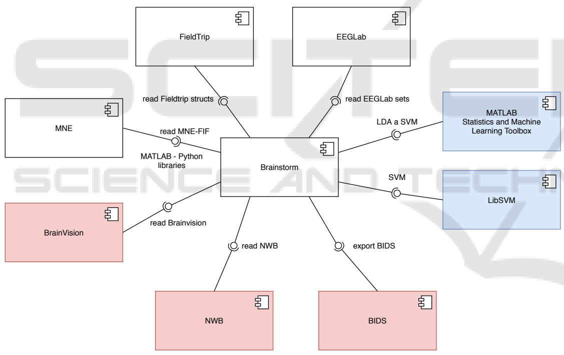

Figure 1: UML component diagram - integration of Brainstorm with data formats, analytical tools and ML libraries.

a MATLAB toolbox for analyzing MEG, EEG, iEEG and fNIRS data. It contains methods for their preprocessing and more advanced data analysis. It supports the BrainVision, BIDS, and NWB data formats and the format used in the EEGLab. The toolbox is free and can be added to the local MATLAB installation.

FieldTrip does not have any GUI; the methods provided can be used to create analytical pipelines in MATLAB. It is integrated with two external ML MATLAB toolboxes: Donders Machine Learning Toolbox (DMLT) and MVPA-Light. DMLT offers a large number of binary and discrete classifiers such as SVM, Naive Bayesian classifier, or regularized discriminant analysis (van Gerven et al., 2011). Figure 2 shows the UML component diagram providing data formats, analytical tools and ML libraries

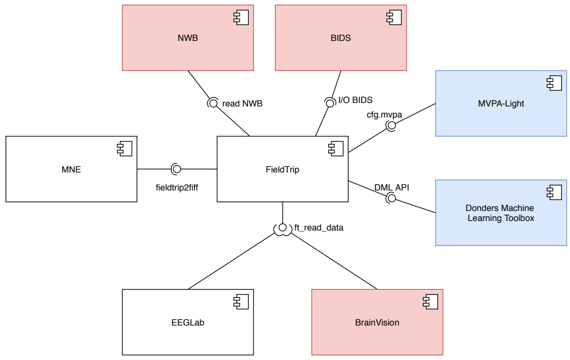

Figure 2: UML component diagram - integration of FieldTrip with data formats, analytical tools and ML libraries.

FieldTrip can work with. EEGLab is a software tool written in MATLAB for processing EEG, MEG and other types of electrophysiological data. It is an open-source project allowing users to extend the capabilities of the tool by implementing their plugins. It provides basic preprocessing methods such as filtering, re-sampling, epoch extraction, baseline correction, or time-frequency transformation and offers several procedures for removing artefacts from a noisy signal.

BCILAB (Kothe and Makeig, 2013) (BCILAB, 2020) is an open-source Matlab toolbox for BCI research that allows ML classification algorithms to be applied to data processed in EEGLab. It provides algorithms for signal processing and contains implementations of ML methods including SVM, LDA and linear regression. Figure 3 shows the UML component diagram providing data formats, analytical tools and ML libraries EEGLab can work with. Neo (Garcia et al., 2014) is a language-independent object model for handling electrophysiology data in multiple formats (NIX and BrainVision format are supported). The motivation for its development was to increase interoperability between Python tools for analysis, visualization and generation of electrophysiological data. Neo is limited purely to data representation and does not provide functions for data analysis or visualization. Its hierarchical data model is used by several different tools for data analysis, visualization or simulation, such as SpykeViewer, Elephant, PyNN, and EphyViewer.

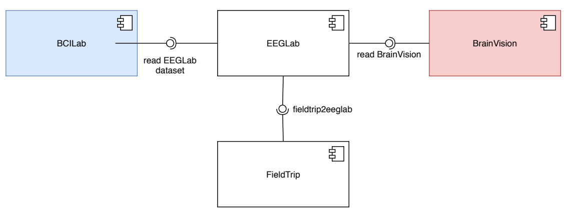

Figure 3: UML component diagram - integration of EEGLab with data formats, analytical tools and ML libraries.

SpykeViewer (Pröpper and Obermayer, 2013), (The NeuralEnsemble Initiative, 2020b) is a cross-platform application with a user interface for visualizing electrophysiological data.

Elephant (Electrophysiology Analysis Toolkit) is an open-source library for the analysis of electrophysiological data in Python. It focuses on generic analytic functions for action potentials and time-series records from electrodes, such as local field potentials (LFP) or intracellular voltage. The aim of the Elephant project is, in addition to a common platform for analytical codes from different laboratories, to provide a consistent and homogeneous framework for analysis built on a modular basis (Elephant authors and contributors, 2020).

PyNN is a simulator-independent language for creating spiking neural networks. Its goal is to allow users to write code for a simulation model only once and then run it on any simulator that PyNN supports. PyNN provides a library of standard models of neurons, synapses, and synaptic plasticities that have been proven to work the same on various supported simulators. It also provides a set of commonly used connectivity algorithms. (The NeuralEnsemble Initiative, 2020a)

EphyViewer is a Python library for visualizing electrophysiological data. It supports the display of possible representations of a given dataset (signal, epochs, events, action potentials) and provides a standalone user interface that allows users to view data from the Neo data format (Garcia and Gill, 2019).

When analyzing the Neo library, no possibility to use ML classification methods directly was found. The only way to apply ML methods to EEG/ERP data stored in the Neo format was by utilizing the Elephant and Pandas libraries. However, Neo provides functions for manipulating objects in its custom data format. The data can be extracted in a format that can be used as an input into DL algorithms within the Keras library and ML algorithms within the scikit-learn library. Figure 4 shows the UML component diagram providing data formats, analytical tools and ML libraries can Neo work with. Pandas is an open-source Python library that provides flexible data structures designed to work with different types of data; it can work with the NIX, BIDS and NWB formats. It is designed as a basic high-level building block for performing practical data analysis. It is useful for different types of data such as ordered and unordered time series or any array data with descriptions of rows and columns. Pandas is built on the NumPy library; it can be integrated into the scientific computing environment with many other libraries such as MNE. (The pandas development team, 2020)

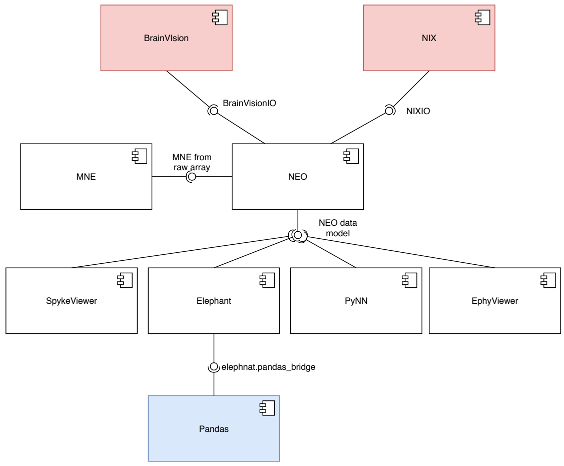

Figure 4: UML component diagram - integration of Neo with data formats, analytical tools and ML libraries.

From the internal pandas' structure, it is easy to extract data in a format that matches the inputs to ML classification methods. Although no use cases for EEG data were found, there are use-cases for pandas time series processed using the Keras library. The sklearn-pandas project was created to integrate pandas structures into the machine-learning pipelines of the scikit-learn library. Another integration of the pandas library with the library of deep-learning methods is the keras-pandas project (Herger, 2020). Figure 5 shows the UML component diagram providing data formats, analytical tools and ML libraries pandas can work with. MNE (Gramfort et al., 2013) is an open-source Python tool providing algorithms for data preprocessing, resource localization, statistical analysis, and estimation of functional connectivity between distributed areas of the brain. It is integrated with the basic Python libraries for scientific computations (NumPy, SciPy) and visualization (Matplotlib).

MNE supports the processing of a number of data

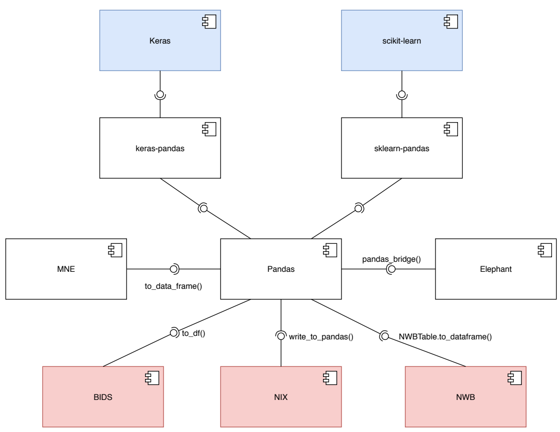

Figure 5: UML component diagram - integration of pandas with data formats, analytical tools and ML libraries.

types, such as EEG, MEG, ECG, SEEG, and ECoG. It supports the BIDS and NIX data formats, BrainVision and EDF formats, but also data in the formats of other analytical tools such as FieldTrip, EEGLab and Brainstorm. MNE cannot work directly with the Neo data format but provides instructions on how to convert data from Neo to MNE structures.

MNE allows its users to export the processed data to the structures of the pandas library. Several open-source projects use the MNE package to process EEG data, and then apply DL methods from the Keras library. The integration of these tools can be thus considered as widespread and proven. The developers of the MNE library also created and shared examples of how DL methods from the PyTorch library can be applied to the data processed in MNE. Figure 6 shows the UML component diagram providing data formats, analytical tools and ML libraries MNE can work with.

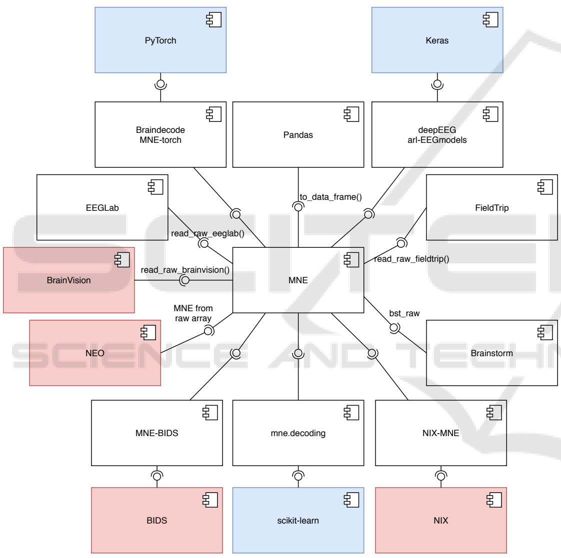

Figure 6: UML component diagram - integration of MNE with data formats, analytical tools and ML libraries.

3.2 Software Tools Comparison

Based on the overview of software tools for EEG/ERP processing given above, their abilities to work with machine and DL methods and libraries, and experience in working with them, we set the criteria and selected the software tools to form more automated and open workflows for the processing of standardized EEG/ERP data. The selection criteria are given and explained below. The evaluation results are summarized in Table 1. Except for licensing, all criteria are graded in the following three levels:

1. need to get the MATLAB license,
2. level and scope of documentation,
   - Low - weak documentation, missing comments in the code, no tutorials,
   - Medium - well-commented basic building blocks of the code, several use cases,
   - High - a large number of very detailed instructions and examples to help users understand the tool, thoroughly annotated code,
3. current availabilities to use ML/DL learning classification methods,
   - Low - the tool does not provide any possibilities of using ML classification methods, or only a very limited set of them (up to 3 classifiers),
   - Medium - the tool allows to use a more extensive set of ML classification methods, it does not contain options for the use of DL methods/neural networks,
   - High - the tool provides extensive possibilities of using machine and DL classification methods, or is integrated with libraries providing these methods,
4. the size of the sets of functionalities suitable for EEG/ERP data processing,
   - Low - the tool does not provide any functionalities for EEG/ERP data processing,
   - Medium - the tool provides a limited set of basic functions for EEG/ERP data processing,
   - High - the tool contains extensive possibilities for processing, analysis and visualization of EEG/ERP data,
5. number of supported formats and tool abilities to work/interact with other sw tools (the level of integration),
   - Low - the tool supports only a very limited set of EEG data formats and is not able to interact with other tools,
   - Medium - the tool supports a limited set of EEG data formats and is integrated with a small set of tools (up to 3),
   - High - the tool supports all or almost all commonly used EEG data formats and provides possibilities of working with other sw tools,
6. community size based on the number of forks of the project on Github,
   - Low - number of forks is < 200,
   - Medium - number of forks is between 200 and 1000,
   - High - number of forks is > 1000,
7. user friendliness,

- Low - deep knowledge of the tool is required for convenient use, some entities are incomprehensibly implemented (it is often related to the lower level of documentation),
- Medium - implementation is difficult to understand, convenient use of the tool requires good knowledge of the tool,
- High - implementation is easy to understand, all functionalities of the tool are easily accessible.

When evaluating the results given in Table 1, the MNE library and the related ecosystem (the BrainVision data format, scikit-learn library and Keras library) were selected as the candidates for an open and convenient workflow suitable for processing of standardized EEG/ERP data.

4 USE CASE EXAMPLE

The use case further presented verifies the feasibility of the selected workflow; integration of appropriate ML/DL classification methods into the chosen tool for EEG/ERP data pre-/processing is practically evaluated. This is demonstrated by replicating a processing workflow of EEG/ERP data described in (Vařeka, 2020).

The related experiment called 'Guess the number' included EEG/ERP data collected from 250 primary and secondary school children; the underlying data are also available in (Mouček et al., 2017). Before the start of the experiment, each participant was asked to select a number from 1 to 9 arbitrarily and to concentrate on it. Then, the recording of the EEG data was started (EEG data from the electrodes Fz, Cz, and Pz and event markers were collected) and the participant was projected with visual stimuli (numbers between 1 and 9) in random orderNecessary experimental metadata from each participant were collected.

The experiment aimed to classify EEG/ERP epochs into the target (thought number) and non-target (another number) classes. The division of epochs into training, validation and testing sets was random. The success of the classification using a convolutional neural network was tested and compared with the traditional LDA and SVM classifiers. The MATLAB tools for data loading, epoch extraction and filtering, and the scikit-learn and Keras libraries for classification purposes were used in the original processing workflow.

The goal of the presented use case is to replicate the original processing workflow described in (Vařeka, 2020) entirely in the Python ecosystem selected above and thus prove that this processing workflow can be completed (when achieving the similar results) only with the use of open resources supporting current standardization initiatives.

The data in the BrainVision format were read and preprocessed (filtering, epochs extraction) using the same methods as described in (Vařeka, 2020), but utilizing the MNE library. The data were further classified with the LDA and SVM ML methods from the scikit-learn library as well as convolutional and recurrent neural networks from the Keras library (the same convolutional neural network as in (Vařeka, 2020) was used). The success of the classifiers was evaluated using the Monte-Carlo cross-validation. The results were compared with the results obtained in (Vařeka, 2020). The resulting implementation and the detailed manual are publicly available in (Kupilík, 2020).

4.1 Testing

Due to the nature of the EEG/ERP data processing and also because the original experiment is not fully reproducible (the artefact rejection procedure is not described in sufficient detail, selection of non-target epochs is random, and generally, neural networks are used), the functionalities of the entire processing workflow and its methods were both inspected visually by plotting the state of the data before and after the application of the methods and by comparing the achieved classification results.

4.2 Results

The entire processing workflow was found functional. The filtering method was applied to the raw data of all 250 subjects, and the method for removing artefacts removed approximately 30% of epochs from both target and non-target sets of epochs. Table 2 shows the averaged results obtained after ten iterations of the classification method using standard metrics. The standard deviations are given in parentheses. The CNN classifier achieved the best AUC (area under the ROC curve), and the SVM classifier achieved the best accuracy and precision. The values closely correspond to the results obtained in (Vařeka, 2020) for the case when the core CNN architecture was used; a positive change of 1-3 percentage points depending on the metric is observable. The RNN classifier that was tested and evaluated only in this processing workflow achieved decent results in terms of AUC and precision compared to other classifiers, but significantly worse values when recall is considered.

The MNE library was found to be a comprehen-

Table 1: Evaluation of software tools for EEG/ERP data processing according to the given criteria.

Sw tool/Criteria	Brainstorm	FieldTrip	EEGLab	Neo	Pandas	MNE
License required	N (Y for ML methods)	Y (Matlab)	Y (Matlab)	N	N	N
Documentation level	High	High	High	Middle	High	High
ML/DL methods utilization	Low	Middle	Middle	Low	High	High
Number of EEG/ERP data processing methods	High	High	High	Low	Low	High
Level of integration	High	High	Middle	High	High	High
Size of community	Low	Middle	Low	Low	High	Middle
User friendliness	Middle	Middle	Middle	Low	High	High

Table 2: Average cross-validation classification results.

	AUC	accuracy	precision	recall
CNN	69.62% (0.79)	64.42% (0.74)	64.89% (1.17)	62.59% (2.52)
RNN	66.2% (0.87)	63.43% (0.83)	65.05% (1.77)	58.58% (4.51)
SVM	65.21% (0.44)	65.22% (0.45)	66.11% (0.64)	62.47% (1.04)
LDA	62.87% (0.38)	62.86% (0.38)	61.94% (0.43)	66.14% (0.8)

sive and convenient software tool for processing and analyzing EEG/ERP data. Its structure is easy to understand, and its resources are logically and hierarchically well arranged. The level of documentation, instructions provided, and commented use cases are very high, and all the above support is continuously updated. For ERP classification, the great advantage of the tool is the possibility to export the analyzed epochs to the format of the NumPy library, which is used by both libraries (scikit-learn and Keras) providing classification methods.

5 CONCLUSIONS

The paper introduced an important step in defining and implementing a processing workflow for EEG/ERP data when open-source software ecosystem and standardized EEG data format are used. In parallel, this step contributes to the completion and maturity of the whole lifecycle of electrophysiology data. The EEG/ERP data lifecycle, standardization initiatives, related EEG data formats, and current view on the use of ML/DL approaches and libraries for EEG/ERP data processing were introduced. Furthermore, software tools for processing EEG/ERP data were analyzed (not only) concerning the utilization of ML/DL methods contained in widespread ML/DL libraries.

Based on the results of the analysis, the MNE, scikit-learn and Keras libraries were chosen for the processing and classification of standardized EEG/ERP data. The proposed and implemented processing workflow was evaluated over publicly available EEG/ERP datasets by replicating the processing workflow described in (Vařeka, 2020). The achieved classification results were compared with similar results. If we take into account positive user experience, the proposed workflow was recommended for open and convenient processing of standardized EEG/ERP data.

ACKNOWLEDGEMENTS

This work was supported by the University specific research project SGS-2019-018 Processing of heterogeneous data and its specialized applications (project SGS-2019-018).

REFERENCES

BCILAB (2020). Matlab toolbox for brain-computer interface research. https://github.com/sccn/BCILAB, last accessed on 2020-09-29.

Brain Products (2020). Bids adopts brainvision core data format 1.0 as one of the recommended official eeg/ieeg data formats. https://pressrelease.brainproducts.com/bids/, last accessed on 2020-08-18.

Brain Pruducts (2019). Description of the brainvision core data format 1.0. https://www.brainproducts.com/files/public/products/more/BrainVisionCoreDataFormat_1-0.pdf, last accessed on 2020-08-17.

Craik, A., He, Y., and Contreras-Vidal, J. L. (2019). Deep learning for electroencephalogram (EEG) classification tasks:a review. *Journal of Neural Engineering*, 16(3):031001.

Elephant authors and contributors (2020). Elephant - electrophysiology analysis toolkit. https://elephant.readthedocs.io/en/latest/, last accessed on 2020-04-19.

Garcia, S. and Gill, J.(2019). ephyviewer 1.3.2.dev. https://ephyviewer.readthedocs.io/en/latest/, last accessed on 2020-04-19.

Garcia, S., Guarino, D., Jaillet, F., Jennings, T., Pröpper, R., Rautenberg, P., Rodgers, C., Sobolev, A., Wachtler, T., Yger, P., and Davison, A. (2014). Neo: an object model for handling electrophysiology data in multiple formats. *Frontiers in Neuroinformatics*, 8:10.

Gorgolewski, K., Auer, T., Calhoun, V., Craddock, R., Das, S., Duff, E., et al.(2016). The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. *Scientific Data*, 3.

Gramfort, A., Luessi, M., Larson, E., Engemann, D., Strohmeier, D., Brodbeck, C., Goj, R., Jas, M., Brooks, T., Parkkonen, L., and Hämäläinen, M.(2013). Meg and eeg data analysis with mne-python. *Frontiers in Neuroscience*, 7:267.

Herger, B. (2020). keras-pandas 3.1.0. https://pypi.org/project/keras-pandas/, last accessed on 2020-09-30.

International Neuroinformatics Coordinating Facility (2020). https://www.incf.org/, last accessed on 2020-08-17.

Kemp, B. and Olivan, J. (2003). European data format 'plus' (edf+), an edf alike standard format for the exchange of physiological data. *Clinical Neurophysiology*, 114(9):1755 – 1761.

Kothe, C. and Makeig, S. (2013). Bcilab: A platform for brain–computer interface development. *Journal of neural engineering*, 10:056014.

Kupilík, F. (2020). MNE_ML - EEG data processing pipeline using the MNE, Keras and the scikit-learn libraries. https://github.com/fkupilik/MNE_ML, last accessed on 2020-10-27.

Mons, B., Neylon, C., Velterop, J., Dumontier, M., Da Silva Santos, L., and Wilkinson, M. (2017). Cloudy, increasingly fair; revisiting the fair data guiding principles for the european open science cloud. *Information Services and Use*, 37(1):49–56.

Mouček, R., Vařeka, L., Prokop, T., Štěbeták, J., and Brůha, P. (2017). Event-related potential data from a guess the number brain-computer interface experiment on school children. *Scientific Data*, 4 cited By 5.

Pernet, C., Appelhoff, S., Gorgolewski, K., Flandin, G., Phillips, C., Delorme, A., et al. (2019). Eeg-bids, an extension to the brain imaging data structure for electroencephalography. *Scientific Data*, 6(1).

Pröpper, R. and Obermayer, K. (2013). Spyke viewer: a flexible and extensible platform for electrophysiological data analysis. *Frontiers in Neuroinformatics*, 7:26.

Roy, Y., Banville, H., Albuquerque, I., Gramfort, A., Falk, T. H., and Faubert, J. (2019). Deep learning-based electroencephalography analysis: a systematic review. *Journal of Neural Engineering*, 16(5):051001.

Rübel, O., Tritt, A., Dichter, B., Braun, T., Cain, N., Clack, N., et al. (2019). Nwb:n 2.0: An accessible data standard for neurophysiology. *bioRxiv*.

Schirrmeister, R., Springenberg, J., Fiederer, L., Glasstetter, M., Eggensperger, K., Tangermann, M., Hutter, F., Burgard, W., and Ball, T. (2017). Deep learning with convolutional neural networks for eeg decoding and visualization: Convolutional neural networks in eeg analysis. *Human Brain Mapping*, 38.

Teeters, J., Godfrey, K., Young, R., Dang, C., Friedsam, C., Wark, B., et al.(2015). Neurodata without borders: Creating a common data format for neurophysiology. *Neuron*, 88(4):629–634.

The Brainstorm team (2020). Brainstorm. https:// neuroimage.usc.edu/brainstorm/, last accessed on 2020-05-02.

The NeuralEnsemble Initiative (2020a). Pynn - a python package for simulator-independent specification of neuronal network models. https://neuralensemble.org/PyNN/, last accessed on 2020-04-19.

The NeuralEnsemble Initiative (2020b). Spykeviewer. https://neuralensemble.org/SpykeViewer/, last accessed on 2020-05-03.

The pandas development team (2020). pandas - data analysis and manipulation tool. https://pandas.pydata.org/, last accessed on 2020-09-30.

Vařeka, L. (2020). Evaluation of convolutional neural networks using a large multi-subject p300 dataset. *Biomedical Signal Processing and Control*, 58:101837.

van Gerven, M., Bahramisharif, A., Farquhar, J., and Heskes, T. (2011). Donders machine learning toolbox. https://github.com/distrep/DMLT, last accessed on 2020-05-02.

Wilkinson, M., Dumontier, M., Aalbersberg, I., Appleton, G., Axton, M., Baak, A., et al.(2016). The fair guiding principles for scientific data management and stewardship. *Scientific Data*, 3.

Zehl, L., Jaillet, F., Stoewer, A., Grewe, J., Sobolev, A., Wachtler, T., et al.(2016). Handling metadata in a neurophysiology laboratory. *Frontiers in Neuroinformatics*, 10:26.