Generalized EEG Data Acquisition and Processing System

1st Vinh D. Le
Computer Science and Engineering
University of Nevada, Reno
Reno, NV, U.S.A
vle@unr.edu

2nd Chase D. Carthen
Computer Science and Engineering
University of Nevada, Reno
Reno, NV, U.S.A
ccarthen@unr.edu

3rd Norhaslinda Kamaruddin
Big Data Analytics and Artificial Intelligence
Universiti Teknologi MARA
Shah Alam, Selangor, Malaysia
norhaslinda@tmsk.uitm.edu.my

4th Alireza Tavakkoli
Computer Science and Engineering
University of Nevada, Reno
Reno, NV, U.S.A
tavakkol@unr.edu

5th Sergiu M. Dascalu
Computer Science and Engineering
University of Nevada, Reno
Reno, NV, U.S.A
dascalus@cse.unr.edu

6th Frederick C. Harris, Jr.
Computer Science and Engineering
University of Nevada, Reno
Reno, NV, U.S.A
fred.harris@cse.unr.edu

***Abstract*—Data acquisition is an integral part in any intelligent system to ensure the data captured can be processed to a meaningful deduction. It is common for the researchers to use the third-party hardware to collect raw data but integrating the processes into a research workflow is always a challenge. This is especially so for individuals working with sensors, such as electroencephalogram (EEG) headsets, as the amount of consumer support that these devices receive from their vendors does not cover the rigors of human-centered research. Researchers are forced to utilize services and functions offered by vendors that may not be tailored to their specific need. In this paper, we present a proposed methodology that is supported by a prototype to show the feasibility of consolidating the processes included in EEG-based user studies, as well as the data analysis that follows. The system presented utilizes a web application in order to facilitate the experimental data collection, record timings, and execute device calibrations. This interface is tied to an institution service-based pipeline that is not only capable of EEG data capturing, but able to produce data products for later analysis. It is envisaged that such an approach can be the first step in automating EEG data acquisition and its subsequent analytics.**

***Index Terms*—data acquisition, electroencephalogram (EEG), automated data processing, software development, affective computing, user study software, data stream, lab streaming layer, self-assessment manikin (SAM), emotiv**

I. Introduction

In an institutional setting, it is not unusual for researchers to outsource hardware for usage within a research project. Data acquisition within research project generally involve using various sensors and devices that are available off the shelf. However, in fields dealing with EEG-based user studies, such as Affective Computing or Brain-Computer Interfaces (BCI), the integration of hardware alongside the study procedures can be rather challenging. To elaborate, the rigors of conducting an IRB-certified user study and publishing the results require a decent variety of dependent variables and tasks to keep track of. Meanwhile, the base consumer support of these hardware devices does not normally provide any aid outside of operating their proprietary device(s)/software. Moreover, many of these vendors will purposely hide additional services behind paywalls, making it unsustainable for many research projects.

In lieu of this, many researchers often adopt a stance of improvisation when it comes to conducting their studies. It is uncommon within academia to prioritize the development of generalized tools to support research workflow, over conducting the actual research itself. It is not unusual for these researchers to either create a temporary script, manually track these variables, or adopt even more tools in order to provide additional coverage of their tasks. As a result, a good number of user study methodologies tend to be inconsistent and no small amount of confusion exists within their research community.

To address these issues, we developed the Generalized EEG Data Acquisition and Processing System (GEDAPS), a software prototype developed from the proposed methodology to facilitate, execute, and support EEG-based user studies within an institutional environment. GEDAPS provides an all-in-one platform in which user studies can be be hosted and also analyzed for validation purposes. GEDAPS utilizes a responsive web interface in order for researchers to setup and conduct their study, while it autonomously tracks configurable dependent variables, such as task time, overall time, and error rate. In the backend, the proposed solution GEDAPS utilizes a service-based backend, coupled with the Lab Streaming Layer (LSL), in order to autonomously label brain activity, transform data, and even provide machine learning solutions in order to generate data products from the EEG headset.

The remainder of this paper is structured as follows: Section II introduces a brief background of the topics covered and some related works, Section III goes into the software design of the GEDAPS system, Section IV contains the details behind the GEDAPS prototype, Section V includes a discussion about the GEDAPS' benefits within an institutional environment, and finally, Section VI finishes the paper off with the conclusion and future work.

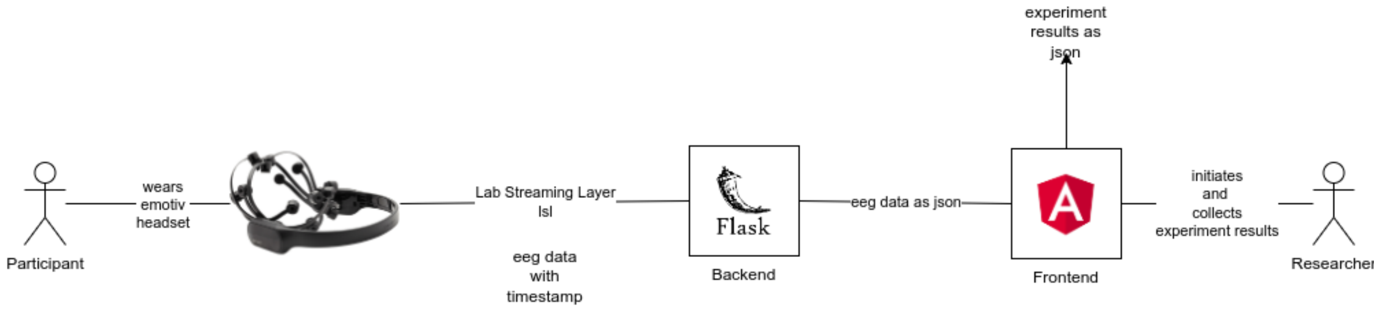

Fig. 1. A diagram showing the whole system regarding the data flow and interactions with the research and participant.

II. Background & Related Work

A. Infrastructure for EEG-Based Studies

The major contributions that this paper provides is two-fold. The first contribution is the software prototype and methodology that automates the delivery of user study specific routines, autonomously gathers the EEG data from a headset, and gathers dependent variable data normally handled by a facilitator. Secondly, researchers are provided an analytical support module that allows for creation of data products through machine learning by utilizing the already recorded experimental data. Looking briefly into the background of similar software at an institutional level will yield a rather light load, as there has not been much movement in that direction. However, the idea of creating systemic improvements to EEG-based studies is not an original idea.

Over the recent Covid-19 pandemic lockdown, the fields that utilized EEGs in order to conduct research were deeply affected. Considering that personnel were required to quarantine, this made research logistics a nightmare. The EEG data collection requires intimate contact between researchers and the test subjects. Such a situation may raise the risk of Covid-19 virus transmission. Hence, many researchers have stopped collecting EEG data or have dramatically changed data gathering protocols during the past year[1]. In response to these severe conditions, research conducted by Demazure et. al. sought to remedy this issue by creating an online framework and methodology so that EEG experimentation could continue during the pandemic lockdown[2]. Similarly, the work conducted by Desai et. al. introduced another solution to the problem of the pandemic lockdown, Sans Tracas, a web application that allowed users to connect their affective devices and create or execute an experiment[3]. The San Tracas software aligns the closest with one solution presented within this paper. GEDAPS differs from these two solutions in that it provides infrastructure for in-person EEG-based studies and that it also provides post-processing services after the study itself.

In a more common research setting, the approach that many EEG researchers adopt consists of re-purposing similar software and/or using scripting languages to achieve the needs of their user study at the time [4], [5]. These setups could consist of form gathering through web applications, such as Google Forms[6], and manual inputing of data through spreadsheeting software, such as Excel[7]. Depending on the complexity of the experiment, researchers will often use scripting languages or analytic toolkits, such as Python[8] or MATLAB[9]. While not ineffective in the slightest, the lack of standards does introduces an element of inconsistency and confusion within the EEG research community.

B. Lab Streaming Layer

A key component utilized as part of the solution presented in this paper would be the Lab Streaming Layer. To elaborate, the Lab Streaming Layer network protocol is a standard of data collection over networked instances that has been established since 2013[10]. For this early prototype, we targeted the Emotiv Epoch X, but will not be limited to just this platform in the near future. Emotiv has since incorporated lsl into their software in order to make it easier to acquire data from the Emotiv headset and make it more portable for other platforms. This network protocol allows for data to be collected for BCI or brain-machine interface (BMI) applications. This network protocol is open source and uses tcp to publish data to other devices that may be listening for a stream.

In regards to EEGs, there has been a strong presence of lsl usage alongside EEG-based research over the past few years. In 2017, the work conducted by Alvarado-Díaz et. al. utilized pylsl, the python-adapted version of lsl, and an EEG headset so that a user suffering from Amotrophy Lateral Sclerosis may control an automatic wheelchair[11]. In 2021, Pieper et. al. conducted a hearing study, utilizing a specialized setup consisting of headphones, an EEG headset, and pylsl coordinating between the two[12]. In more recent times, a researcher by the name of Tim De Boer had created a link between EEGs and video games through the usage of pylsl, and was able to play the game Space Invaders[13].

III. Software Design

At the highest level, GEDAPS is designed as a service-based platform run by institutional technicians to support affiliated researchers conducting EEG-based research. The three major components that compose GEDAPS consists of the frontend web interface, the backend services that automates data capturing through lsl, and finally, the analytical module

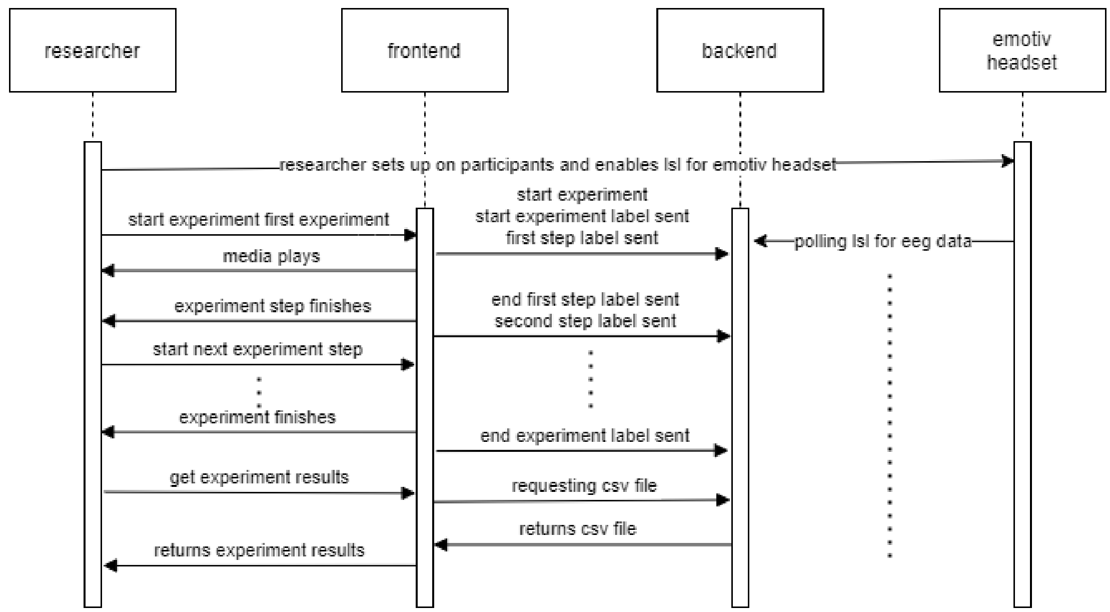

Fig. 2. Sequence diagram of the flow from the researcher starting the experiment and the experiment as a whole.

that pipes in the stored EEG data in order to perform analytical operations, such as valence and arousal classification.

The GEDAPS system communicates to each of these components through a series of HTTP requests. These requests are invoked autonomously during key events within the experimental phase, in order to reduce the workload of a researcher. The key events in question are related to the procedures found within a typical EEG user study, such as the moment that a task is started or completed, the moment when the baseline configuration phase is happening, or the moment that a resting phase has been initiated. Fig. 1 illustrates a condensed representation of the workflow through the system during an experiment. In this representation, a participant will wear an EEG headset, which in the case of the current prototype is the Emotiv Epoch X headset. The EEG headset captures the brain signals and begins to stream it towards the flask backend, with aid from the lsl integration. During this time, the key events within the experiment is inserted within the data stream. From there, a curated data format, in the form of multiple CSVs, JSON, or a multi-sheet xlsx file, is generated. After a full experiment has concluded, the researcher may use the frontend to download the data file from the backend. Otherwise, the data is stored securely in a relational database tied to the backend for potential use in the future.

In regards to the workflow between the researcher and the system, when the researcher conducts an experiment, the frontend generates an internal signal to the backend to start its autonomous recording. The backend then starts the process of streaming data and creating special labels, as mentioned above. During this time, the researcher and/or the facilitator will be guiding the participant along the user study. During EEG experiments, physical activities involving the participants are discouraged as to reduce the noise and inaccuracies within the headset. As such, a researcher and/or facilitator will either be navigating the participant through a type of media or utilizing the built-in autonomous experimentation feature that generates timed media interactions. As soon as all tasks in an experiment are completed, the frontend will signal to the backend to stop and the backend then generates and stores the file. To better illustrate these events, Fig. 2 shows the interactions between the researcher, the frontend, and the backend.

IV. Software Prototype

A. Frontend

The frontend for the GEDAPS prototype was designed and implemented with Angular[14] alongside Angular Material [15] to design custom components. For the purposes of this prototype, we targeted primarily affective computing, although the process remains largely the same for BCI. The two main types of user study supported by the current prototype consists of user perception tests to generate labelling data and EEG-based user studies whom use that labelling data. These modes were chosen primarily to represent the common types of user studies involving EEGs within affective computing and BCI. These modes are controlled based on form input and content is rendered dynamically by drawing from the backend JSON responses, as shown in Fig. 3.

For the user perception tests, we designed the frontend to display the self assessment manikin (SAM) so that users can rate their valence, arousal, and dominance [16]. We chose the self assessment manikin due to the notoriety within the affective computing field as the standard in which to gather a participants valence, arousal, and dominance. Valence represents the impact of the emotion ranging from positive to the negative effect. Arousal denotes the activation state that covers from active to passive. Finally, dominance epitomizes

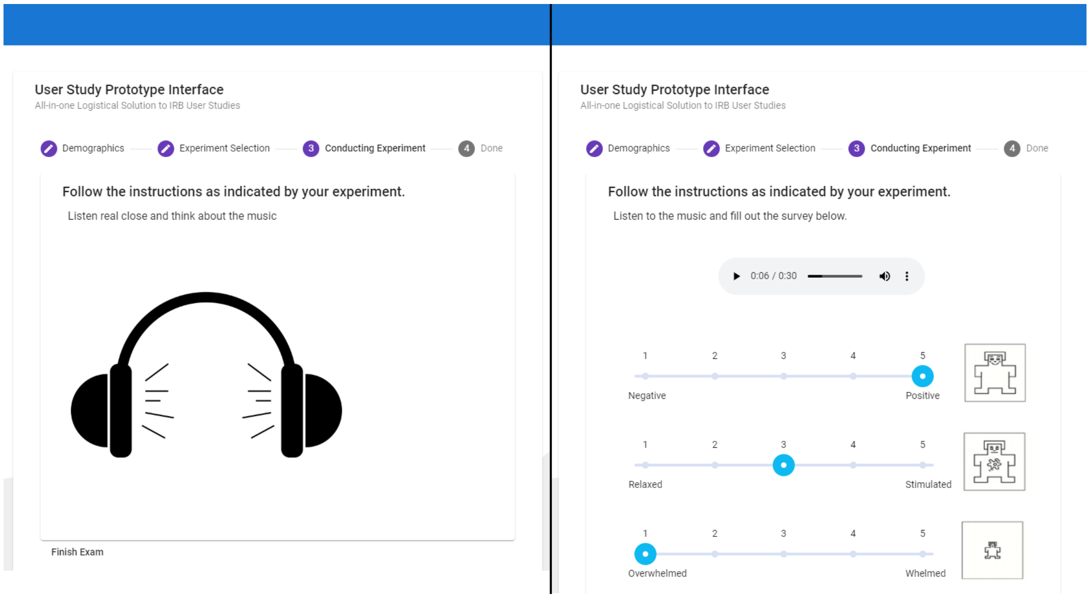

Fig. 3. The GEDAPS prototype provides support to two major types of user studies: user perception and EEG data collection.

the degree of control exerted by a stimulus, which is denoted by a range of big to small influence.

In regards to the EEG-based user studies, this mode utilizes metadata set by the researcher to generate essentially an autonomously operated and recorded media slide deck. The media currently supported by this prototype consist of audio, imagery, and video. During this time, the frontend syncs up dependent variable collection with the stream of participant EEG signals captured through the Emotiv Epoch X headset in reaction to media content being presented. The JSON configuration of this portion allows for different phases of the experiment to be configured. To elaborate, the timed content may shift between a customized video player, audio player, and dynamically loaded imagery. Timers were especially vital and were incorporated in order to allow for greater quality control among each of the assigned tasks.

As it stands now, our frontend is currently served with node.js, but it could be easily scaled up and ran on Kubernetes or even containerized with docker. This will be covered further in Section VI. Angular was chosen as it remains a dependable industry standard and provides many of the needed user interface features for this application. Angular is also known to be rather effective at catering to feature-rich applications and has a more complete out-of-the-box solution than many of it's competitors. Considering the feature density of our design, we found that it was appropriate to incorporate our backend with Angular.

B. Backend

The GEDAPS prototype utilizes a REST API backend written in Python, using the Flask micro framework, to generate a series of request calls to operate the lsl bridge between the backend and the EEG headset [17]. A high-level overview can be viewed in Fig. 4, where it demonstrates the capturing of data from Emotiv's lab streaming layer. Pylsl was a key component in the backend design and is used to connect with the EEG signal streaming from Emotiv's software [18]. The Flask backend handles recording the EEG signal and other signals being sent by Emotiv by streaming them to a specified data format. Starting and stopping the recording on the backend is done through HTTP requests to a start or stop URI route within this backend API. As the EEG datastream is being captured by the backend, the backend also inserts timestamps of each key event within the data file itself.

Key events are captured and registered by the backend through an event route that accepts identifiable metadata as parameters. These events can include anything, ranging from a song starting, ending, or the pre and post calibration phase of a EEG user study when baselines are being measured. When the backend invokes the event route, it splices the event into the generated data file along with a timestamp, similar to the routes described above. As previously mentioned, the backend API was designed so that it can tap into the captured datastream originating from the Emotiv headset and insert custom time-series data entries. This is done as an alternative to inconsistent timings done manually by researcher or even the latency-driven timings generated by the frontend interface. All data pertaining to the operation of the headset is unified into one source without any splicing of data from multiple sources.

The backend API unified the data streams and the labels generated by the experiment. This in turn made it so that the extraction of meaningful data from the EEG recordings significantly easier. Another benefit behind this design is that the backend API follows a more generic approach and allows for other EEG headsets outside of Emotiv to be used, assuming that lsl streaming is supported. Finally, the organized key events recorded from the backend API eliminated the tedious

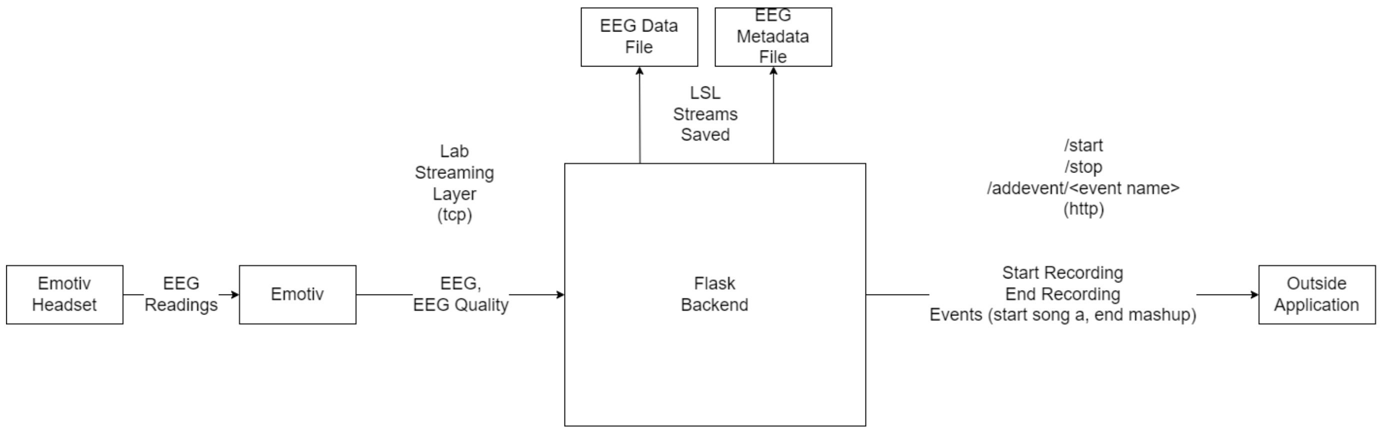

Fig. 4. High level diagram of data being recorded and capture from Emotiv's lab streaming layer in a Flask Backend.

task of data pre-processing and allowed for near 1:1 insertion into analytical processes, such as feature extraction.

C. Analytical Module

The Analytical Module is currently a standalone module separate from the frontend and backend, as it is very much in the early stages of development. This module analyzes the EEG data captured from the backend and correlates the data to the appropriate key events thanks to labels generated during the experimentation phase. A main feature of this module involves the pre-processing of the experimental data by resolving unwanted artifacts within the recording. This could entail removing eye blinks, normalizing the data into a appropriate range, or identifying erroneous signals generated from involuntary movement. After the pre-processing of the EEG data, the module then uses AutoML in the form of TPOT to extract a data product [19]. In the case of the current prototype setting, this data product is the classification of both valence and arousal within a participant.

V. Discussion

GEDAPS, at a core concept, seeks to elevate the tedious processes involved within EEG-related user studies. By establishing an infrastructure solution for these types of studies, we wanted to enable an opportunity for past user studies to be edited and reproduced with ease. To continue along with the idea of alleviation, the usage of lsl within GEDAPS sets the foundation for a more generalized integration platform for EEG headsets or other wearable bio-metric devices. At the researcher level, GEDAPS brings to the table an institutional resource for researchers to design, facilitate, and execute experiments, especially as more features are added in the future.

In the beginning, the design of this software system was created in mind so that it can be easily ported to an institution's infrastructure or a research cluster accessible to researchers. An institution's infrastructure resources would be an ideal location to store sensitive data, such as the data gathered within a IRB user study. Especially considering that EEG readings are specific to a individual, the choice becomes even more obvious that an institution should be securing this data. Ideally, an institution's technical staff could set up an internal resource specifically meant for storing personal EEG data for it's affiliated researchers. In that situation, a researcher would only need to utilize this software platform to facilitate their experiment and store their results. The design of the GEDAPS software and the prototype described in this paper is a strong first step in this direction.

The execution of this system will not be without it's own set of challenges. Designing and implementing the GEDAPS system will require the allocation of it's own set of computing resources. This could be especially difficult, depending on the budgeting available for the institution. Additionally, the institution will have to also budget for the establishment of an administrator to oversee the daily operation and evolution of this system. On the system side, the integration of additional devices could prove challenging due to potential software or hardware incompatibilities

VI. Conclusions & Future Work

This paper presented a methodology and prototype, dubbed GEDAPS, designed to address the issue of logistics and infrastructure within EEG-related user studies. Ultimately, GEDAPS is currently tailored to support two types of user studies: user perception or labelling studies and EEG data collection. GEDAPS utilized both the lab streaming layer network protocol and a custom backend API in order capture EEG data and insert labels pertaining to user study tasks. GEDAPS' frontend is designed to be adaptive, the interface is designed so that content of page is dynamically constructed based on the metadata surrounding the experiment itself. The last major feature of GEDAPS was the creation of an analytical module that automates the generation of data products surrounding the curated EEG data.

In the near future, we plan to expand on GEDAPS by deploying it onto an internal institution Kubernetes cluster and running EEG user studies as case studies. Additionally, there are plans to incorporate other EEG headsets and bio-metric devices, such as a headset from openBCI and the Emotibit [20], [21]. With these new devices, we would work towards unifying the data coming from each data set into a data model that would represent the different types of channels or sensors coming from each device. Also, we plan to evaluate the user perception and effectiveness of our software in comparison with other existing methodologies with a user study.

The analytical module at present is separate from the system as a standalone module. A future endeavor would be to incorporate this module formally into the system to allow for machine learning models to be trained on the data within the current pipeline. Finally, the selection of available machine learning solutions were limited due to time constraints. We plan to expand this selection by incorporating more advanced solutions, such as AutoKeras or AutoPyTorch to support Deep Learning[22], [23].

References

[1] A. M. Simmons and S. J. Luck, "Protocol for reducing covid-19 transmission risk in eeg research," *Research square*, 2020.

[2] T. Demazure, A. J. Karran, J. Boasen, P.-M. Léger, and S. Sénécal, "Distributed remote eeg data collection for neurois research: A methodological framework," in *International Conference on Human-Computer Interaction*, Springer, 2021, pp. 3–22.

[3] R. Desai, R. Orji, E. Roudaia, and A. B. Sekuler, "Sans tracas: A cross-platform tool for online eeg experiments.," in *Companion of the 2022 ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, 2022, pp. 37–43.

[4] K. Kutt, G. J. Nalepa, B. Giżycka, P. Jemiolo, and M. Adamczyk, "Bandreader-a mobile application for data acquisition from wearable devices in affective computing experiments," in *2018 11th International Conference on Human System Interaction (HSI)*, IEEE, 2018, pp. 42–48.

[5] X. Shen, X. Hu, S. Liu, S. Song, and D. Zhang, "Exploring eeg microstates for affective computing: Decoding valence and arousal experiences during video watching," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, IEEE, 2020, pp. 841–846.

[6] Google, *Google forms: Free online form creator*. [Online]. Available: https://www.google.com/forms/about/ (visited on 10/27/2022).

[7] Microsoft, *Microsoft excel spreadsheeting software*. [Online]. Available: https://www.microsoft.com/en-us/microsoft-365/excel (visited on 10/27/2022).

[8] P. S. Foundation, *Welcome to python.org*. [Online]. Available: https://www.python.org/ (visited on 10/27/2022).

[9] MathWorks, *Matlab - mathworks*. [Online]. Available: https://www.mathworks.com/products/matlab.html (visited on 10/27/2022).

[10] EEGLAB, *EEGLAB 2018 - Hands-on Lab Streaming Layer (LSL) Workshop 2018*, en-US. [Online]. Available: https://sites.google.com/ucsd.edu/eeglab-2018/hands-on-lab-streaming-layer-lsl-workshop-2018 (visited on 05/10/2022).

[11] W. Alvarado-D'ıaz, B. Meneses-Claudio, and A. Roman-Gonzalez, "Implementation of a brain-machine interface for controlling a wheelchair," in *2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, IEEE, 2017, pp. 1–6.

[12] K. Pieper, R. P. Spang, P. Prietz, *et al.*, "Working with environmental noise and noise-cancelation: A workload assessment with eeg and subjective measures," *Frontiers in neuroscience*, vol. 15, 2021.

[13] T. de Boer, "I built a brain computer interface to play space invaders using thoughts and i will explain you how, including code examples in python," [Online]. Available: https://towardsdatascience.com/i-built-a-brain-computer-interface-to-play-space-invaders-using-thoughts-23980cb4faf7 (visited on 10/27/2022).

[14] Angular. [Online]. Available: https://angular.io/ (visited on 10/27/2022).

[15] Angular, *Angular material*. [Online]. Available: https://material.angular.io/ (visited on 10/27/2022).

[16] M. M. Bradley and P. J. Lang, "Measuring emotion: The self-assessment manikin and the semantic differential," *Journal of Behavior Therapy and Experimental Psychiatry*, vol. 25, no. 1, pp. 49–59, 1994, ISSN: 0005-7916. DOI: https://doi.org/10.1016/0005-7916(94)90063-9. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0005791694900639.

[17] Pallets, *Welcome to Flask — Flask Documentation (2.1.x)*. [Online]. Available: https://flask.palletsprojects.com/en/2.1.x/ (visited on 05/08/2022).

[18] C. Kothe, D. Medine, C. Boulay, M. Grivich, and T. Stenner, *Quick Start — Labstreaminglayer 1.13 documentation*. [Online]. Available: https://labstreaminglayer.readthedocs.io/info/getting_started.html (visited on 05/08/2022).

[19] W. Fu, R. Olson, Nathan, *et al.*, *Epistasislab/tpot: V0.11.5*, version v0.11.5, Jun. 2020. DOI: 10.5281/zenodo.3872281. [Online]. Available: https://doi.org/10.5281/zenodo.3872281.

[20] EmotiBit, *Emotibit.com*. [Online]. Available: https://www.emotibit.com/ (visited on 10/27/2022).

[21] OpenBCI, *Open source tools for neuroscience*. [Online]. Available: https://openbci.com/ (visited on 10/27/2022).

[22] AutoML, *Auto-pytorch*. [Online]. Available: https://www.automl.org/automl/autopytorch/ (visited on 10/27/2022).

[23] H. Jin, Q. Song, and X. Hu, "Auto-keras: An efficient neural architecture search system," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 1946–1956.