MediGraph

a healthcare application

Ioan Croitor Catargiu

Alexios Vasileiou

Department of Computer
and Systems Sciences

IoT Project

Autumn term 2023

Supervisor: Rahim Rahmani

Laboratory Assistants: Fahim Shahariar, Ismail Hossain

Stockholm
University

Abstract

Designed for both Android and iOS, this project amalgamates IoT capabilities, neuroscience insights, and mobile technology to develop a novel healthcare application. Utilizing EEG and pulse sensors as IoT 'things,' the app collects and analyzes brainwave and pulse data to assess potential schizophrenia indicators and monitor patient health remotely. This project aims to transform healthcare by enabling remote patient assessments, potentially reducing the need for physical presence during healthcare evaluations, providing at the same time a user-friendly experience.

Keywords: EGG, Graphs, Healthcare, Internet of Things, React Native, Disease Detection, Data Monitoring, Schizophrenia

Acknowledgement

We would like to thank Professor Rahim Rahmani, as well as his fellow laboratory assistants Fahim Shahariar and Ismail Hossain, DSV, Stockholm University and several anonymous reviewers for their helpful comments and time without which this project would be impossible.

Table of Contents

1 Introduction 0
2 Background 1
2.1 Theoretical Grounding 1
2.2 Usage Description 2
3 Method 3
4 Design and Development 5
4.1 Data Acquisition 5
4.2 Python Scripts 5
4.3 Application Interface 5
4.4 Flask Python 6
4.5 Firebase and MongoDB Databases 6
5 Evaluation & Discussion 7
5.1 Evaluation 7
5.2 Discussion 7
6 Limitations & Future Work 8
6.1 Limitations 8
6.2 Future Work 8
References 9
Appendix A – Python Scripts 11
Appendix B – EEG 14

1 Introduction

The four main concepts, which Internet of Things (IoT) is connecting together, are *data, people, process, things*. This constitutes the determinant factor for which its applications are widely spread and cover very diverse areas of interest, from an individual's house to healthcare environments. Of course, to support such a wide range of application, the devices used in each case should be able to support the IoT enabled capabilities, which refer to *monitoring, measuring, controlling, automating, optimizing, learning*.

Speaking of this project, nowadays, given the emerging advancements within the field of neuroscience, experts are able to extract certain information about the psychological, emotional (and other) states, of an individual based on its brain activity. Methods such as EEG (Electroencephalography) can track and record brainwaves in specific brain regions, making plausible for researchers to process data and extract useful information about patterns and features.

So what this project aspires, is to acknowledge the healthcare provider about the patient brain activity at a certain moment, diagnose schizophrenia, while it will also include useful information about the patient's pulse, using different sensors. The data can be later aggregated to produce more accurate and complex predictions about the patient's condition and (potentially) prevent the diseases. Thus the project is essentially a healthcare application, for both Android and iOS phones, with its primary focus being the ability to inform about the probability of a patient having schizophrenia and provide graphical visualization of brainwaves and pulse data

Breaking down the aforementioned scenario, the *data* is the patient's brainwave and pulse data, which will be monitored and recorded by the EEG sensors and pulse sensors, which are the *things*. Then the data can be *processed* to conclude useful information about the patient's condition, patient who is, obviously, the *people.* What is more, the capabilities are also supported in this case, as the project consists of *monitoring* through the sensors, specifically EEG sensors and sensors form wearable devices for pulse values, which collect and transmit the data, performing at the same time the *measuring* quantifying the collected data, while *controlling* is achieved by checking which data will be taken into consideration. What is more the task are conducted without the human innervation, as everything is achieved through the application, thus *automating* and also *optimizing* and *learning* by making use of machine learning algorithms which improve over time.

2 Background

2.1 Theoretical Grounding

The materialization of such a project requires rigorous research on the topics that is related to, with the main ones being brainwaves and EEG. We will try to provide a short but comprehensive summary on how brainwaves theory and how the brain activity is recorded using electroencephalography (EEG).

Starting with the brainwaves, they are nothing more than *'oscillating electrical voltages in the brain measuring just a few millionths of a volt' (Abhang et al., 2016).* So, it is a measurement of electrical voltages which the neurons in our brains produce when communicating with each other. Those brainwaves can be recorded through EEG, which is a method where microscopical electrodes (consisting of small metal discs with thin wires) are pasted on the scalp, in order to capture brain activity. It is obvious that, the measurement is just an overview of the brain activity, as for more targeted measurements other methods can be used, while its accuracy can be questioned *(see figure 3 Appendix B).* The result of the EEG are graphs representing the oscillations of electrical voltages, and the brainwaves within them can be recognized by their amplitude or frequency. *'There are five widely recognized brainwaves' (Abhang et al., 2016),* each of them representing different states, as observed below.

Figure 1. Characteristics of the Five Basic Brain Waves (Abhang et al., 2016)

Frequency band	Frequency	Brain states
Gamma (γ)	>35 Hz	Concentration
Beta (β)	12–35 Hz	Anxiety dominant, active, external attention, relaxed
Alpha (α)	8–12 Hz	Very relaxed, passive attention
Theta (θ)	4–8 Hz	Deeply relaxed, inward focused
Delta (δ)	0.5–4 Hz	Sleep

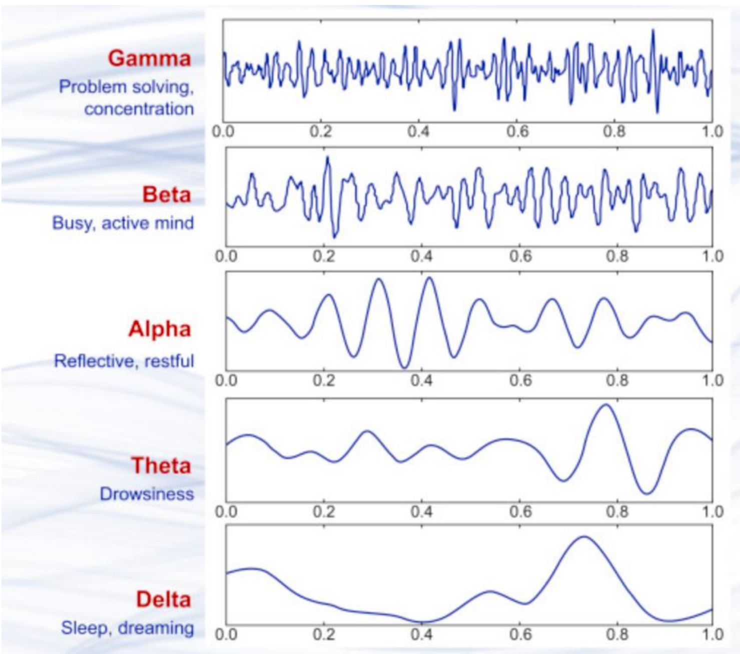


Figure 2. Brain wave samples with dominant frequencies belonging to beta, alpha, theta, and delta bands and gamma waves (Abhang et al., 2016)

Regarding the pulse, its rate is representing the heart rate, meaning how many heartbeats occur at a given time. Their recording can be achieved via multiple methods, such as through clinical recording, using dedicated equipment (high accuracy), or through wearable devices (Apple Watch etc.), or even by placing our fingers on our wrist or neck feeling the beats (not that accurate due to the human factor).

2.2 Usage Description

Having established the core concepts and theories around our project, we have to provide a more detailed and specific application/scenario, in which this project can be proven useful, as well as describing and highlining what makes it stand out.

Therefore, as previously mentioned, this project caters to healthcare professionals and their patients, introducing a pioneering approach by integrating brainwave and comprehensive data analysis into a user-friendly application. Unlike existing health software, often complicated and unwieldy, this app aims to empower every personnel member and patient with a convenient tool accessible via their phones. The novelty lies in combining brainwave analysis and other data processing into a single, easy-to-use application. Its significance is evident as it addresses the current challenges faced by healthcare software – complexity and lack of user-friendliness. By implementing algorithms for neural activity analysis, it offers a crucial feature: enabling professionals to access pertinent patient information remotely, reducing the need for constant physical presence for both the professional and the patient, and at the same time unlocking new horizons for the healthcare industry, as the patient, in the near future, won’t be required to be physically present to get his/her examination done, because that will be possible via remote sessions with the doctor.

3 Method

Given the complex nature of the project, we had to carefully plan the timetable, to schedule each step, so that we manage to finish everything within the desired timeframe. So first, we performed research, regarding the theoretical grounding, which was time consuming, since neither of us had previous knowledge about neuroscience, apart from the knowledge gained from the course Human Cognition in Human Computer Interaction of the former fall period, where an introduction in the realms of neuroscience and neuropsychology was made to us. Moving on, completely understanding the concepts of IoT and how they will be connected and implemented in our project (as described in the Introduction), constituted a new challenge, which we nevertheless overcame, with the help of the lectures, laboratories, and of course our own effort, including dedicating extra time to searching and learning.

After, we started watching tutorials and searching for techniques, for both manipulating brainwave and pulse data, and for developing machine learning algorithms which will be able to discern patterns for whether a patient has schizophrenia. After several failed attempts, we found a very helpful tutorial on YouTube which guided as to create the machine learning model. The other scripts are in detailed explained in the Appendix A.

Coming to the application, we decided to make it healthcare-professionals-oriented, meaning that it will serve healthcare professionals' needs, initially implementing the following functions:

- Personalized profile
- Data visualization (brainwaves and pulse graphs of patients)
- Notification/Information about the possibility that a patient has schizophrenia and indices that a patient might have epilepsy (based on the brainwaves)
- Notification/Information about the pulse of a patient (if it is critical or normal)
- History

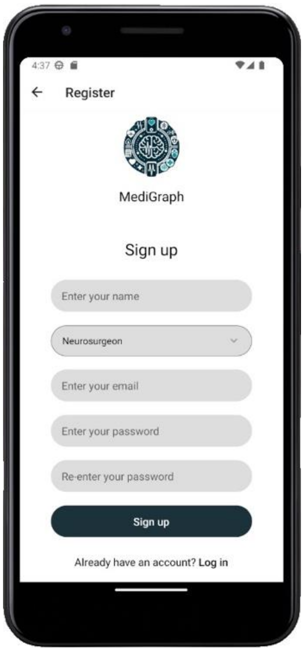
(*in the end, however we did not manage to implement the epilepsy)

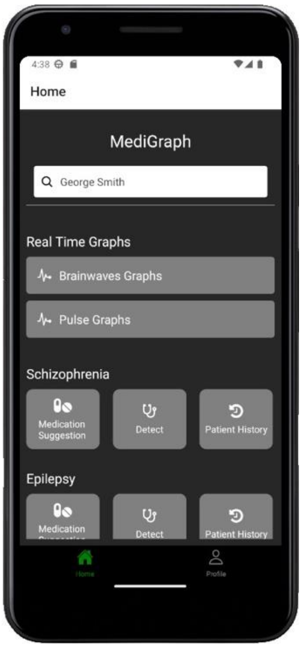
To design an application which such functionality we used React Native, a widely used JavaScript based frameworkj for app development through Visual Studio, as well as Flask Python, which is a micro web framework written in Python, which helped us transferring the data returned from the Python scripts, to the React application. Moreover, as a database, Firebase was used to retrieve the files for schizophrenia detection, and MongoDB for implementing the history functionality.

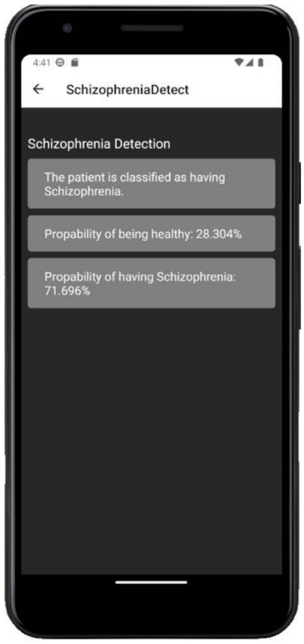
React Native was chosen over our initial option, Android Studio, because one of us had more experience with it and because it will make it available for iOS devices as well. Flask, on the other hand, was a handy way to serve as a server from which the application will receive the needed data from the scripts. Similarly, Firebase and MongoDB, were opted for as both of us had previous experience using them. Initially, the application did not have a database and the files were hardcoded, but we thought that it is better to retrieve them from a centralized place, after discussion with the laboratory assistants.

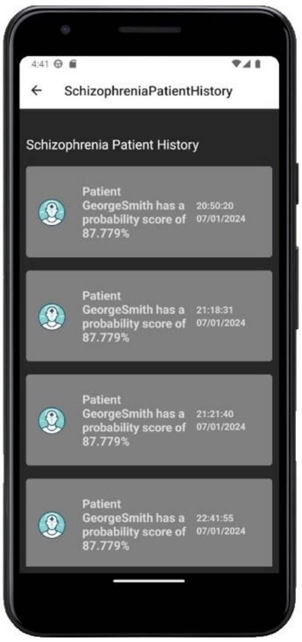
Regarding the login, signup, and profile page of the application, we should clarify that they are not a real authentication system, so the values are not stored in a database, but they are hardcoded. This is due to the fact that the time limit did not permit us to properly implement it, so we decided to focus our attention on more important functionalities, a decision which was made, also after discussing with the assistants.

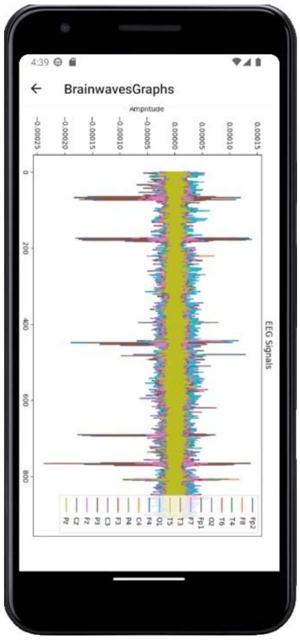
Further below, we provide snapshots of our application.

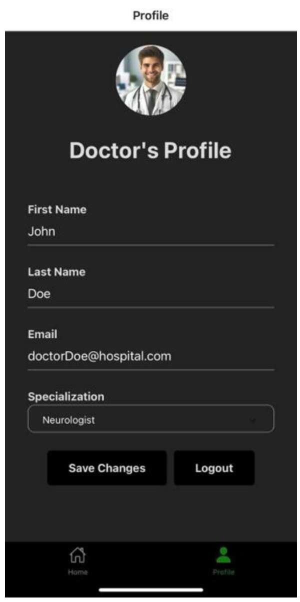












4 Design and Development

4.1 Data Acquisition

The data was gathered from the internet, mainly from research papers, through Google Scholar, references of which are provide in the References section. The data for used for the schizophrenia machine learning algorithm was gathered from a paper about '*Graph-based analysis of brain connectivity in schizophrenia' (Olejarczyk E et al., 2017)* and consists of two sets, the first set containing 14 brainwave samples from healthy patients and the second set containing brainwave samples from schizophrenically diagnosed patients. In addition, the data used for the pulse graphs, was gathered from a paper dealing with *'Motion and heart rate from a wrist worn wearable and labeled sleep from polysomnography' (Walch, O. (2019))* and consists of multiple files with entries representing time and heart rate data. So, the sensors from where the data was gathered are EEG sensors (brainwave data) and Apple Watch (pulse data).

4.2 Python Scripts

As already mentioned, Python was used for manipulating and processing the data, which are explain in detail in Appendix A.

4.3 Application Interface

The application was developed using React Native, which enables the usage of it on both Android and iOS devices. Having previous experience and researching the web, we managed to produce the basic template, based on our mockups. The main application page is in *App.js* screen, from where all the other screens are created and called.

Concerning the screens, they are 12, each of which serve a different functionality, First, we should mention that the styling of each page is the same along the app (except login and signup page), which is a minimal, dark-themed template, which can be observed in the snapshots provided above. Starting with *LoginScreen.js* and *RegisterScreen.js*, these are the two pages for login and signup respectively, but as we mentioned before they do not represent and actual authentication system, as the values entered are hardcoded and they are not connected to a database but are there just for completeness reasons. After entering the account, we land on the *HomeScreen.js* which is empty initially, having only the search bar needed to search a patient. After searching a patient, the functionalities are 'unlocked', and the grid of functions is revealed, making possible for the person to select from a variety of options. These are to see real-time brainwave or pulse graphs and a triad of functions for each of schizophrenia and epilepsy detection: medication, detect, history. The search is performed by making the application search into an array with users, and a dropdown option menu is shown when a user, that is containing the same latter as the one entered in the search box, is found. The patient will be saved and passed through each page (through the route parameters-url), so that the identity of the patient is preserved, ensuring that the data used in next screens is indeed bind to that specific person. Pressing the brainwave graph screen, will redirect to the *BrainwavesGraphsScreen.js* screen. This is the page where a graphical visualization of the brainwave activity is provided, meaning an image which is passed to the application from the processing the script *data_extract.py* does. Analyzing how the data is passed to the application, which is the same for the other screens, first a query is made using the asynchronous 'fetch' operation making a POST request to the flask server, requesting the data, and 'fetchData' performs the retrieval of the data through a 'GET' request, which in this case is a blob-format image containing the brainwave graph of the patient. Going back, if the pulse graph button is pressed, then it will show a real-time visualization of the pulse of the patient, over time, which is what the page *PulseGraphsScreen.js* does. The data is passed from the *heart_rate_values.py* script through flask, and then the values are plotted on a graph which was made, similarly to the graph showed by the *heartrate.py*. Moreover, when pressing either of the medication proposal buttons, then medication will be proposed for the selected disease, according to the phase in which the patient resides in, redirecting to *SchizophreniaMedicationScreen.js* or *EpilepsyMedicationScreen.js*, respectively. Furthermore, when pressing the button detect in the schizophrenia section, we land in the *SchizophreniaDetectScreen.js*, and the most recent data of that specific patient will be downloaded and passed to *predict.py* which will spit the probability of the patient having schizophrenia. The data then is saved in MongoDB database so it can later be accessed via the history button, which redirects to *SchizophreniaPatientHistory.js* or *EpilepsyPatientHistoryScreen.js*, accordingly. Lastly, the profile button down in the screen redirects to the *ProfileScreen.js* from where the professional can edit his profile details.

4.4 Flask Python

The *server.py* contains all the python scripts inside functions, which value is returned then to React, serving, thus, as a server. Inside it each function has a name and a route from which it can be accessed, route which is passed in the form of urls within the queries. The python scripts remain almost unaltered.

The *firebase.py* deals with downloading the *.edf* files (containing the brainwave measurement) of a given patient. The files are downloaded inside the download folder and after the detection is performed, then the file is deleted. Each time all the files concerning a patient are downloaded (given that the patient name is the same as the name of the folder containing its data), and then the most recent one is selected (given that the naming of the files is based on date & time of creation).

The *mongoDB.py* is concerned with pushing and pulling the data of a patient, which is used for the history page, storing the patient's name, date & time, as well as the probability score of having schizophrenia.

4.5 Firebase and MongoDB Databases

The data stored in the Firebase database are the *.edf* files (containing the brainwave measurement) of the patient, and they are saved in folders, each named with the name of the patient. The data is added manually, but in a real-life scenario the EEG sensors would place it there automatically. Each folder contains multiple files, each named by date & time of creation.

The data stored in the MongoDB database, is in BSON (Binary JSON) format, with each entry having the format *_id, patient_name, probability_scores, detection_date*.

5 Evaluation & Discussion

5.1 Evaluation

The application consists of a working interface through which healthcare providers are able to perform various monitoring functions as well as a detecting one (schizophrenia). Indeed, it is an easy-to-use and user-friendly application, which is easy to understand even for non-healthcare related persons, proving thus one of the main points of the application. Additionally, the application supports the detection of schizophrenia, but only for certain types of files, specifically *.edf* files, and of certain size. The accuracy of the predicting model is one of ~66%, which is not a high one due to several reasons analyzed below, but as mentioned in the introduction, this application does not aim to replace the doctor's contributions, but to help the professional, enabling new horizons for the future. Furthermore, for the epilepsy detection, although a template algorithm was created, in the end we did not manage to implement it properly within the given time limit, so we removed this functionality, and consequently the functionality of the epilepsy history. However, both medication proposal screens, work properly, as they simply provide information to the doctor about the potential medication for the patient.

In terms of testing the application, within the evaluation, we asked several individuals, mainly from our residence, what they do they think of the interface application, and we received positive feedback, which means validating the user-friendliness of the application. Moreover, a person who studied biology assured us that she thinks that our application would be really useful in healthcare environments.

5.2 Discussion

Our aspiration with this project, was to explore our capabilities, trying to combine groundbreaking state-of-the-art advancements from the fascinating field of neuroscience with artificial intelligence, and connect it to the IoT. Moreover, we wanted to achieve simplicity and to manage to communicate the concepts in a manner that will be easy for everybody to understand and show that it is possible to embed all this functionality in a single application, combining knowledge from different fields of science, fact which was achieved to a certain extent. Furthermore, the knowledge and information gained throughout this whole process is of very significant value and worth.

6 Limitations & Future Work

6.1 Limitations

As every project, our projects had certain limitations, mainly due to data and time insufficiency. First the accuracy of the trained model is not very high (~66%), because of the small number of training data (28 samples). We tried to search for more data in several websites, and even reached a professor from DSV, who communicated with a clinic, but both attempts were unsuccessful. What is more, due to the inability of finding data we did not get to test our model on unseen data to validate its performance, one reason being that the model was trained on a certain set of data, with a precise number of features, making it implausible to test data with different sizes and recorded differently, engendering in poor scalability. Worth mentioning, however, is that we tried to perform matrix reduction using methods, such as PCA, when passing the data to the serialized model, but it turned out to be in vain. Another reason, not as important as the previous one, is the nature of the machine learning model, which is Linear Regression, because if a CNN was used instead then the accuracy could have been slightly higher.

Continuing, the authentication system was not properly and completely setup, using an actual database, because of the time limit and because we decided it was not of great importance as it can be added, relatively easy, any time after.

Coming down to the epilepsy, unfortunately we were not able to implement the functionality of the detector, because the data we had was insufficient or in the wrong format, resulting in an inability to carry out this requirement.

6.2 Future Work

Regarding the future work, the project provides a strong grounding for becoming an actual full working application in the market. Its potential resides in the fact that it is easy-to-use and provides complex information, in a simple and useful way.

So, complementary to the previously mentioned limitations, more data can be used so that all the functions will be available (epilepsy), while different algorithms and techniques can be used to improve the generalizations and the scalability. In addition, an authentication system could be integrated properly.

Also, more functionalities can be added later on, apart from the one proposed by us, such as detection of brain tumors or of Alzheimer, facilitating even more the healthcare industry, and possibly resulting a multi-purpose application for general usage within healthcare environments.

Last but not least, a very promising and ambitious future application will be to receive all the needed data directly from the implanted microchips, inside individual persons, such as Neuralink’s microchips, which are currently under development and experiment, and which are designed to help individuals take actions by just thinking at them, while it also has other, extended applications.

References

Information

https://choosemuse.com/blogs/news/a-deep-dive-into-brainwaves-brainwave-frequencies-explained-2 (brainwaves)

https://en.wikipedia.org/wiki/Electroencephalography (EEG)

https://en.wikipedia.org/wiki/Neural_oscillation (brainwaves)

https://link.springer.com/chapter/10.1007/978-3-030-21642-9_8 (paper on CNN schizophrenia classification)

https://www.mayoclinic.org/tests-procedures/eeg/about/pac-20393875#:~:text=An%20electroencephalogram%20(EEG)%20is%20a,lines%20on%20an%20EEG%20recording (EEG measurement)

Abhang, P. A., Gawali, B. W., & Mehrotra, S. C. (2016). Technological basics of EEG recording and operation of apparatus. In Elsevier eBooks (pp. 19–50). https://doi.org/10.1016/b978-0-12-804490-2.00002-6 (brainwaves)

https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/electroencephalogram-eeg#:~:text=An%20EEG%20is%20a%20test,activity%20of%20your%20brain%20cells (electrodes)

Biasiucci, A., Franceschiello, B., & Murray, M. M. (2019). Electroencephalography. Current Biology, 29(3), R80-R85 (EEG figures)

https://my.clevelandclinic.org/health/diagnostics/17402-pulse--heart-rate (heart rate and pulse)

https://www.medanta.org/patient-education-blog/normal-vs-dangerous-heart-rate#:~:text=Abnormal%20Heart%20Rates%20or%20Heart,doctor%27s%20intervention%20is%20a%20must. (critical pulses values)

Kickstart-EEG-research-Ebook

Tutorials

https://youtube.com/playlist?list=PLtGXgNsNHqPTgP9wyR8pmy2EuM2ZGHU5Z&si=tzZsh6euXH-2oAeU (ML for schizophrenia classification)

https://youtu.be/0-S5a0eXPoc?si=4disd-zIfyF3c2D1 (React Native tutorial)

https://www.youtube.com/watch?v=7LNl2JlZKHA (Flask tutorial)

https://www.geeksforgeeks.org/mongodb-and-python/ (Flask and MongoDB)

https://mne.tools/stable/index.html (mne library)

Databeses

https://repod.icm.edu.pl/dataset.xhtml?persistentId=doi:10.18150/repod.0107441 (schizophrenia)

https://physionet.org/content/sleep-accel/1.0.0/heart_rate/#files-panel (pulse)

https://openneuro.org/datasets/ds003555/versions/1.0.1 (epilepsy)

Appendix A – Python Scripts

*We mention that explicit explanation of the code it is provided as comments within the python files

**mne* library is a widely used open-source python package designd for visualizing, and analyzing human neurophysiological data as EEG

- *schizophrenia.ipnyb*

This notebook, contains the manipulation of the *.edf* files, extracting the data from 28 files, 14 healthy patients labeled as *hXY.edf,* as well as 14 patinets diagnosed with schizophrenia, where X, Y integers, while it also has the training for the machine learning model. The data was gathered using Electroencenphalography (EEG) and more specifically '*The dataset comprised 14 patients with paranoid schizophrenia and 14 healthy controls. Data were acquired with the sampling frequency of 250 Hz using the standard 10-20 EEG montage with 19 EEG channels: Fp1, Fp2, F7, F3, Fz, F4, F8, T3, C3, Cz, C4, T4, T5, P3, Pz, P4, T6, O1, O2. The reference electrode was placed between electrodes Fz and Cz.'*, as mentioned, in the dataset page.

Starting with, using the *mne* library, we managed to read the files. Specifically, this is the purpose of the *read_data()* function which returns a *numpy* array, containing al the data which is labeled after with 0 if the patient is healthy and with 1 if not. Furthermore, several statistical features are extracted from the data, which are then concatenated into a single array, using the *concatenate_features().* After, using a Logistic Regression Classifier, we train a machine learning model based on the features of the provided data. Lastly, the trained model is serialized to the *model.joblib* file, so that it can be later used to predict the probability of a patient being schizophrenic, with new unseen data.

The whole script was materialized by making use of the guidance provided by the tutorial mentioned in the references.

- *predict.py*

This script returns the probability of a patient being classified as schizophrenic or not, based on new unseen data, and using the trained model which was serialized after its training, as described in the *schizophrenia.ipnyb* section.

Thus, first the model is deserialized and then the *read_new_data()* function is used to read the new file (essentially is the same as *read_data()* in *schizophrenia.ipnyb*) and then the same features are extracted using the *concatenate_features()* and lastly, the features are passed to the model which makes a prediction.

It should be mentioned that when implemented in the *flask* server, the files are selected from the firebase database download directory.

- *data_extract.py*

This script deals with determining within which frequency band the extracted EEG signal resides in primarily, as well as plotting the brainwave signal to a *matplot* graph.

To begin with, after loading and reding the file, the frequencies are filtered to keep the ones within the range (1, 40) Hz, while the power spectral density (PSD) is calculated using Welch's method, while at the same time the frequencies of the channel are calculated. PSD provides information about the distribution of power into frequency components present in the signal, which are further used to calculate indices corresponding to the predefined frequency bands. This is done for all the five different frequencies bands (alpha, beta, gamma, delta, theta), and after the graph with the frequency bands is plotted, using matplotlib. *(see Figure 4, Appendix B)*

- *heart_rate_values.py*

This script is concerned with extracting the values from the *.txt* files, which contain recorded pulse values of patients during sleep, along with time.

First, we need to clarify the format of the data inside the files which is in the format '*date (in seconds since PSG start), heart rate (bpm)*', according to the database page. The PSG refers to polysomnography which is a sleep study recording various physiological parameters during sleep. The values of time within the file are negative because they refer to the time before a certain event occurred (e.g., a sleep stage), but it is out of our area of interest. So, in the code the time values are stripped accordingly, removing the negative sign in front of the time values, and reversing the array so they are in an ascending order. The code results in two arrays, one containing the time values, and one containing the pulse values. These values will be later used to simulate a pulse detector machine.

- *heartrate.py*

This is the script on which the visualization of the data in the application (using React native) was based upon.

It contains the main function *update_plot()* which simulates a pulse detector, resulting in real-time graph (technically multiple graphs are stacked one upon another).

Appendix B – EEG

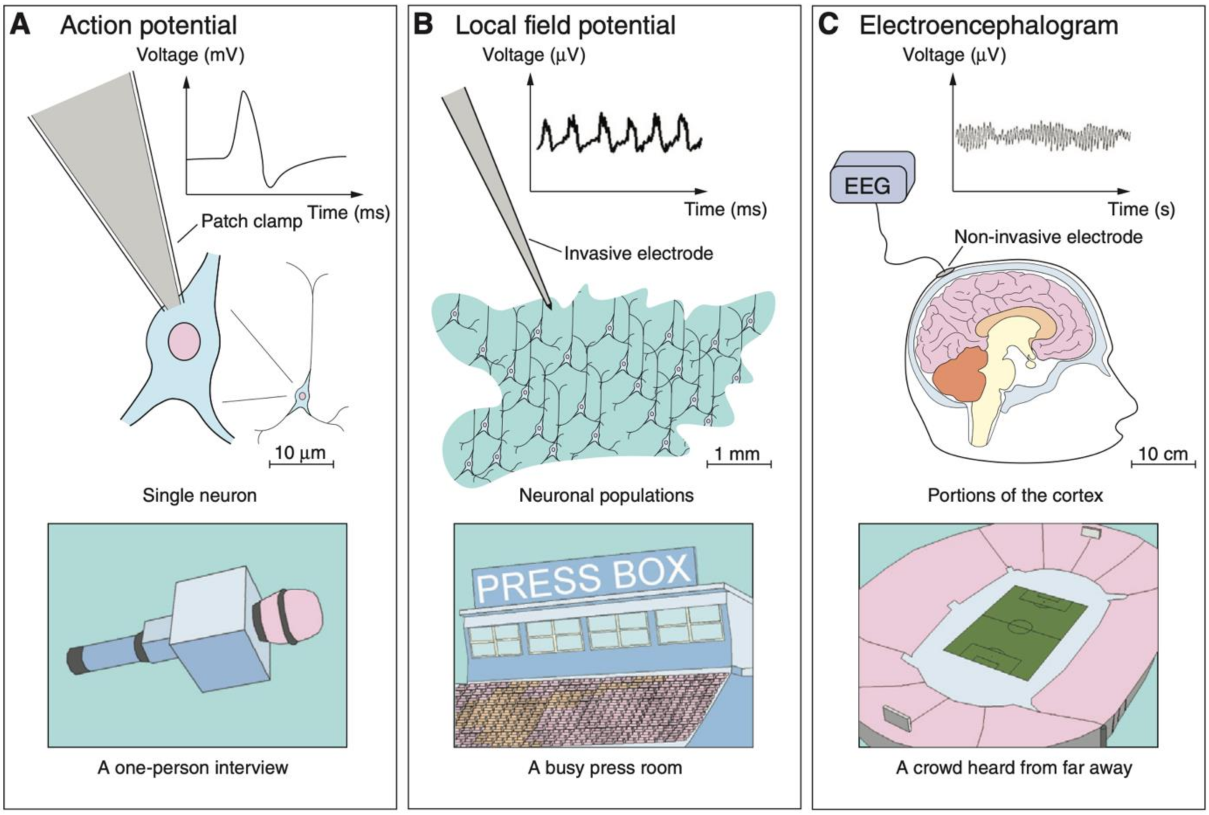


Figure 3. A good analogy of what EEG recordings are capturing (Andrea Biasiucci et al., 2019)

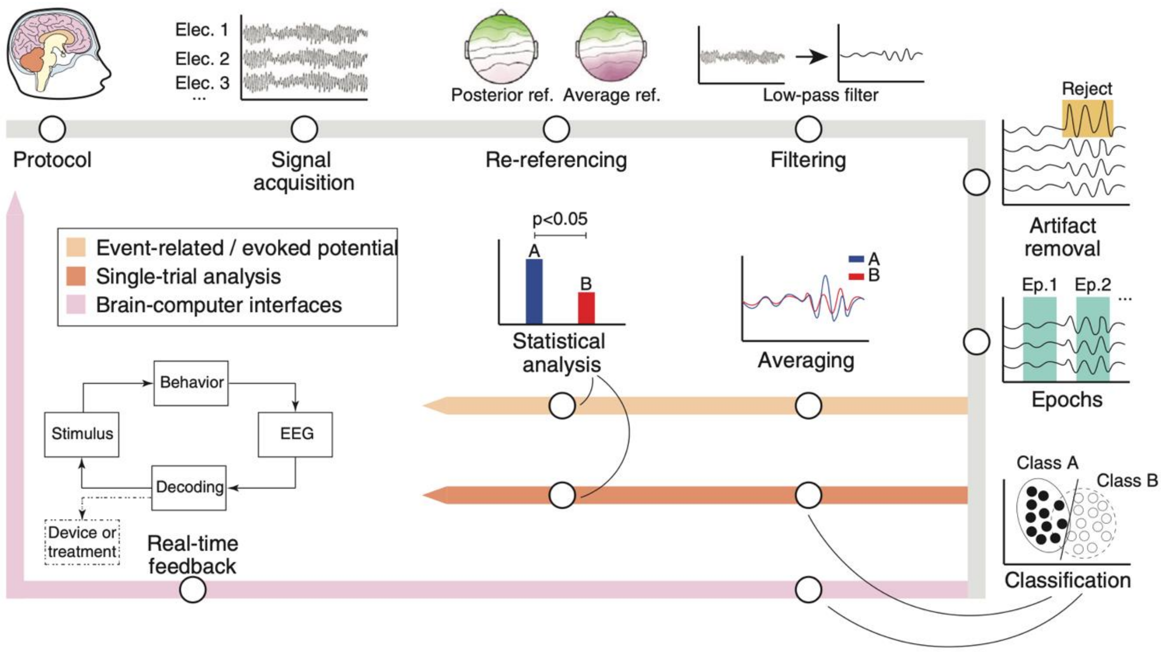


Figure 4. An overview of EEG processing and analysis steps (Andrea Biasiucci et al., 2019)