

Development of a Web Application for Audio Communication using EEG Signals

Joel Mathew¹, Dr.E.Grace Mary Kanaga², M.Bhuvaneshwari³, Caleb Stephen⁴ and Prashant Srinivasan Sarkar⁵

¹⁻⁵Karunya Institute of Technology and Sciences, Coimbatore, Tamilnadu, India

joelmathew20@karunya.edu.in, grace@karunya.edu, bhvnshwari@gmail.com, calebstephen@karunya.edu.in, prashantsrinivasan@karunya.edu.in

Abstract— An accurate analysis of electroencephalogram (EEG) data was achieved to identify the needs of the patients who are unable to coherently articulate their needs to caretakers. An audio output via Bluetooth speaker was designed to articulate on behalf of the patient to communicate his/her needs to the caretaker. The analysis done in this project was on the cognitive stimuli-based EEG data. The project goes on in three main stages, Firstly, making of the web application with user friendly interface; Secondly, building of the machine learning model to identify the need of the patients and thirdly, provide an audio output suited for the identified need, so that the caretaker will know what exactly the patient need. The User interface is made using html and CSS and bootstrap is used to make the application more responsive and dynamic. The main pages contain the links that can be used to access the different parts of the project. Each page is supported by a python backend and it is applied using the flask python framework. The machine learning models are pre-trained and stored using pickle in python. The models were trained with the cognitive dataset that was provided and the performances of the models are recorded. A training section is also provided in the web application that can be used to demonstrate the performance of the model which will display the accuracy score of the trained data. Each time a prediction is made in the algorithm page the python backend is programmed so that the output of the prediction is given back to the caretaker in audio form.

Index Terms— Brain Computer Interface, Electroencephalography, Python, Web based tool

I. INTRODUCTION

Over the past decade, huge investments in Brain Computer Interface (BCI) research have been supported by various companies, universities and research centers that have seen a goldmine of ideas for further development in this area. Even the famous Elon Musk and Mark Zuckerberg have invested heavily in this research area with the aim of merging biological and digital intelligence [1]. A brain-computer interface (BCI), sometimes called a brain-machine interface (BMI), is a direct communication pathway between the brain's electrical activity and an external device, most commonly a computer or robotic limb. BCIs are often directed at researching, mapping, assisting, augmenting, or repairing human cognitive or sensory-motor functions [3]. Implementation of BCIs range from non-invasive (EEG, MEG, EOG, MRI) and partially invasive (ECoG and endovascular) to invasive (microelectrode array), based on how close electrodes get to brain tissue [3].

This project builds a web application that will help the caretaker to use the brain wave to analyze the wave and identify the need of the patient. Different classes that map the need of the patient are classified to determine the need of the patient. The next step is to use an algorithm to classify the data set and to identify the need. Using the

trained models, we have to implement it in the web application where the user can upload the dataset and get the need as the output. And the final step will give a voice output that will notify the caretaker about the need of the patient. The current communication system comprised of an input data entry which was fed into the web application and to the machine learning algorithm using the python web application framework, which in turn produces a classification as the output. This output was then fed into a predetermined program that alert the caretaker via a voice message played over a Bluetooth speaker.

Various machine learning algorithms can be used to classify the data into various classes which can be used to obtain various outputs and classifications. And some of the algorithms will be explained in the coming pages. To develop the web application we used HTML, CSS, Flask. The algorithms used for this application are:

Logistic Regression: Logistic regression is a statistical model that estimates the probability of a binary dependent variable based on one or more independent variables. It is widely used in various fields to analyze binary outcomes, such as disease diagnosis or customer behavior prediction. The model uses a logistic function to relate the independent variables and the log-odds of the dependent variable. Logistic regression is a popular and powerful tool in statistics and machine learning due to its simplicity, interpretability, and flexibility [2], [3].

K-Neighbors Classifier: The k-nearest neighbor (k-NN) classifier is a non-parametric machine learning algorithm used for classification and regression tasks. In k-NN classification, the class of an unlabeled data point is predicted based on the majority class of its k nearest neighbors in the feature space. The distance between two data points is typically measured by some distance metric, such as Euclidean distance or Manhattan distance. The optimal value of k depends on the complexity of the problem and can be determined through cross-validation or other model selection techniques. K-NN is a simple and intuitive algorithm that does not require training or any assumptions about the underlying distribution of the data. However, it suffers from some drawbacks such as the need to store and search the entire training set for each prediction, which can be computationally expensive for large datasets or high-dimensional feature spaces. k-NN is also sensitive to the choice of distance metric and the presence of irrelevant or noisy features [4].

Naive Bayes Algorithm: The Naive Bayes algorithm is a probabilistic classification algorithm based on Bayes' theorem, which states that the probability of a hypothesis (such as a class label) is equal to the prior probability of the hypothesis multiplied by the likelihood of the evidence (such as the feature values) given the hypothesis, normalized by the marginal likelihood of the evidence. The algorithm is called "naive" because it makes the simplifying assumption that the features are conditionally independent given the class label [6]. The algorithm can handle high-dimensional datasets and is relatively fast and easy to implement. However, the assumption of feature independence may not hold in many real-world scenarios and the algorithm can suffer from the problem of "zero-frequency", where a feature that appears in the training set with only one class label may cause the algorithm to assign zero probability to that feature for the other class labels [7].

II. LITERATURE SURVEY

As of now there is no communication tool that utilizes the brain computer interface technology which helps the users for communication. The BCI enables the user to control the outcomes using their thoughts if we're able to collect the right dataset and by using the right algorithms. Being able to control things using thoughts are not impossible but the problem is cost of equipment that is needed to collect the brain wave from the user and also the right application that can use the data collected and predicts the need.

Speller based: The BCI speller, which was among the first application of the idea, is a classic visual example of the BCI. Farewell and Donchin proposed the first BCI speller. It used a row and column (RC) paradigm based on event-related potential (ERP) P300. The ERP P300 in EEG signals has a positive potential and appears at around 300 ms, with variations in the precise time of its occurrence, following stimulus brought on by an unlikely event [8]. The central portion region of the brain is where this potential is most noticeable. Most BCI spellers that were offered for a few years following its inception were based on the P300. The BCI was designed with the visual P300, auditory P300, and tactile P300 as well. The P300 speller's poor pace is a drawback. Users must wait a very long time since each output must travel via every stimulus in a holonomic trial. Additionally, the P300's amplitude of potential declines with prolonged usage, lowering its classification accuracy [8].

Cursor based: One other implementation that is done using the BCI technology is using it to move a cursor from center of a screen to a target located at different positions on the screen. It is done by training subjects to use the 8–12 Hz, mu rhythm recorded from the scalp over the central sulcus. $\neg\neg$ Larger amplitudes pushed the cursor up, while lesser amplitudes moved it down. The amplitude of the mu rhythm was measured using on-line frequency analysis. Over a period of weeks, the subjects developed the ability to swiftly and precisely alter the mu rhythm's amplitude so that the cursor typically reached the destination in three seconds [9].

Evaluation of the distribution of amplitudes in response to top and bottom targets led to the derivation of the parameters that converted mu rhythm amplitudes into cursor movements. This study's utilization of these distributions was both one of its defining characteristics and the main success factor.

In this project, we are using the brain data collected from different patients and other subjects to make a model by using various machine learning algorithms and using the data to develop an application that can use the model to predict the patient need from the data that is passed through the application.

III. PROPOSED METHOD

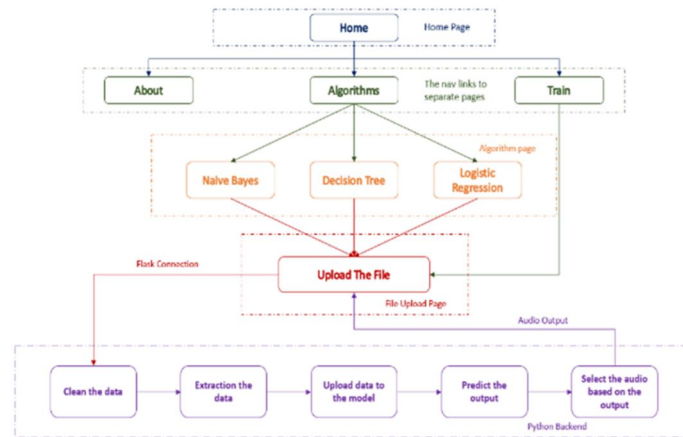


Figure 1: architecture

The architecture shown in Figure 1 describes an overall structure of the developed application. More detailed information is given below to get a clear understanding of the application.

The web application is a user-friendly platform that consists of four pages - the home page (Figure 3), about page, algorithm page and train page. The home page serves as the landing page of the application, containing only the project front page. The page is designed in such a way that it is visually appealing, leaving a good first impression to the end-user. Each page inside the web application has links to every other page in the application, with the Navigation panel (Figure 2) containing the home, about, algorithm, and train page link. The user can easily navigate to each page in minimal clicks as possible.

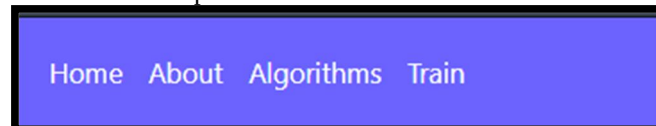


Figure 2 Navigation bar

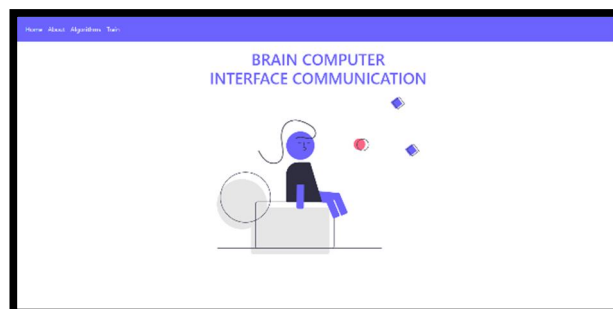


Figure 3 Home Page

The about page (Figure 4) provides a brief overall description of the whole Brain Computer Interface and the page design ensures that the user won't be bored while browsing through it. The algorithm page is the page where the actual action takes place, providing links to the file upload page and allowing the user to choose between three algorithmic models offered by the application. These algorithms are Decision Tree, Logistic Regression and Naïve Bayes and the user/care-taker can select either of the algorithms, which will direct them straight to the page where

they can upload the file with the data that needs to be classified. Each algorithm button will take the user to the particular page route for the file upload, ensuring the correct algorithm model is being run in the background. The train page (Figure 5) is provided to demonstrate the training operation that is being run, and it is similar to the file upload page as they use the same template. The difference between the two pages is that the output returned by the train page will be the model accuracy whose graph is shown in Figure 6. The machine learning algorithms used are pre-trained, and the models are stored inside a pickle file. As the user selects a particular model, the model is being run to get the output. The process follows the common steps while creating the models, where the dataset is selected and cleaned to remove any outliers and other impurities. The data set is then split into the dependent and independent

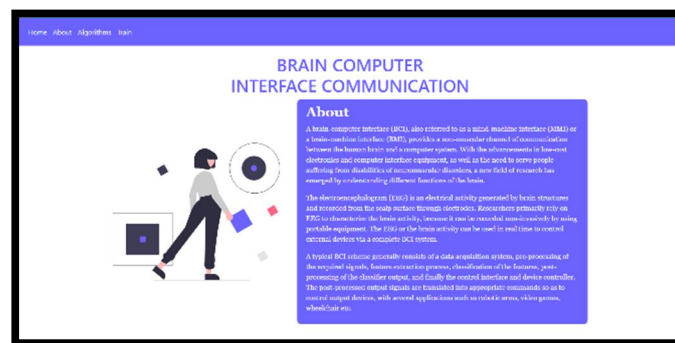


Figure 4 About Page

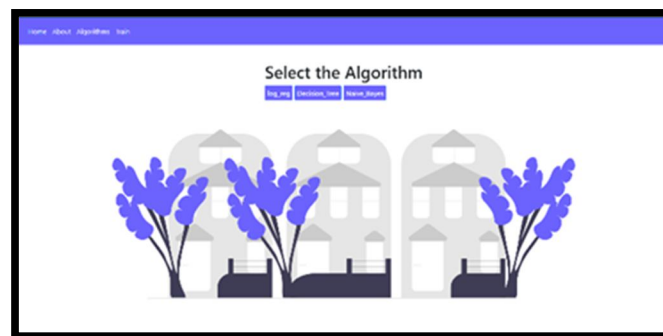


Figure 5 Training Page

Variables and the models are trained using these pre-processed data sets.

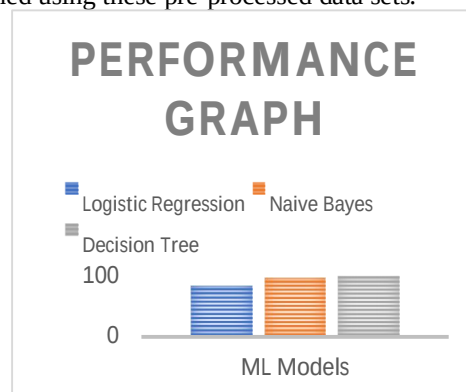


Figure 6 Performance Graph

Each machine learning algorithm is imported from the sklearn library and trained with the dataset. The accuracy score and the confusion matrix are obtained to compare the model performances (refer Figure 6 for performance details). The model is then stored into a pickle file so that the model is no longer needed to be trained each time it is called, reducing time requirements. The audio outputs are created by converting the text messages to audio and saved in the directory. The audio output is played using pre-built Python library functions. The audio is selected by checking the result returned by the classifier developed for the application.

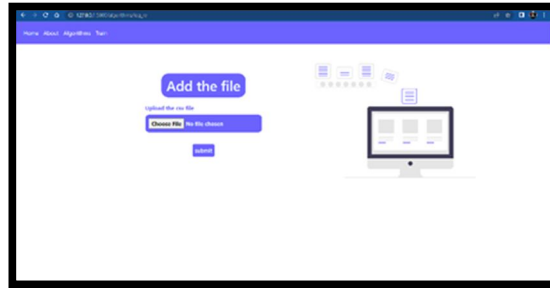


Figure 7 File Upload

The output data can be classified into six classes - Recreation (switch on the Television), Feeling thirsty, Feeling Hungry, handling natural call, need to take medicine, and Switch on the fan. The block diagram (Figure 1) displays the working flow of the process, where the dataset is cleaned, split, and used to train the models. The models are tested for accuracy and confusion matrix, and the output is stored in a pickle file. This model is then used to classify the data taken from the file upload page (Figure 7) and select the appropriate audio output. The web application offers an intuitive and user-friendly platform that provides users with easy access to machine learning algorithms and audio output classifications.

IV RESULTS

The model performances are as demonstrated by confusion matrix is given below. The models are developed just for the demonstration purpose for the application to show the workability. The model was trained using the cognitive data of one subject and the test results are as given below.

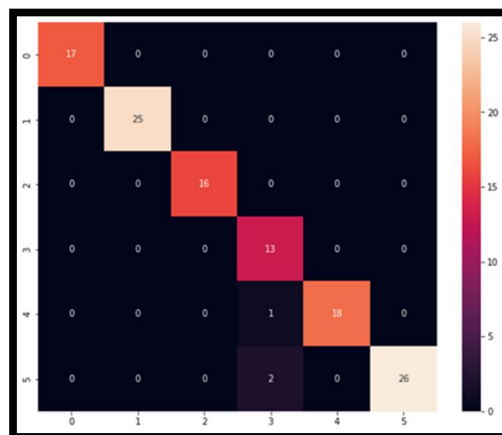


Fig 8 Naïve Bayes

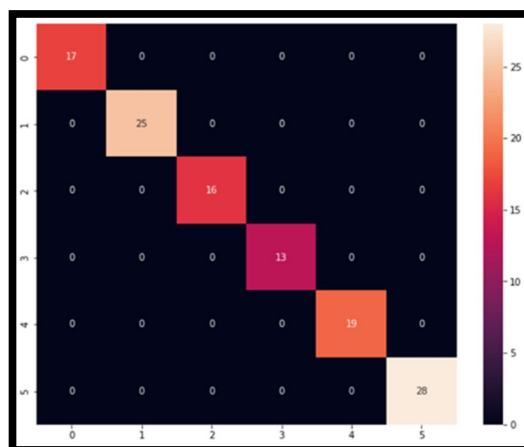


Fig 9 Decision Tree

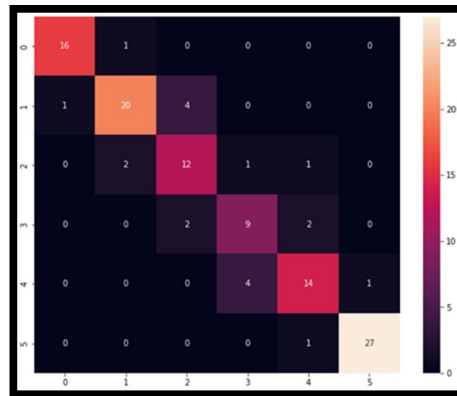


Fig 10 Logistic Regression

The application is able to be developed by adding more algorithms that utilizes deep learning techniques which will give a much general model that can be used for predicting the needs without giving the subject specific data.

V. CONCLUSIONS

As we observed in this project, the BCI technology can be used to explain and identify the need of the patients that will help the rehabilitation technologies to take a big leap into the future where even the paralyzed patients can also able to communicate. As the project showed we were able to make a communication medium that will give the audio output to the caretaker notifying/reminding about the particular need of the patient at the time. Further development can be done to make the project to do it in real time. At this moment, an offline product that reads EEG signals and classifies them has been achieved. The constraints with the current setup entail that the input data must be pre-recorded and fed manually. A device that can monitor real-time EEG signals is required to make this product more practical. The scope of brain computer Interface technology is bright and the developed web application shows that rather more studies and improvement is required to make the technology efficient and low priced.

REFERENCES

- [1] R. G. Lupu, F. Ungureanu, and C. Cimpanu, "Brain-computer interface: Challenges and research perspectives," Proc. - 2019 22nd Int. Conf. Control Syst. Comput. Sci. CSCS 2019, pp. 387–394, 2019, doi: 10.1109/CSCS.2019.00071.
- [2] Peña Yañez, "El anillo esofágico inferior.," Rev. Esp. Enferm. Apar. Dig., vol. 26, no. 4, pp. 505–516, 1967.
- [3] A. Agresti, "3Rd-Ed-Alan_Agresti_Categorical_Data_Analysis.Pdf," International encyclopedia of statistical science, vol. 47, no. 4, pp. 755–758, 2013.
- [4] P. Li, Y. gen Ding, P. peng Yao, K. min Xue, and C. ming Li, "Study on High-Temperature Flow Behavior and Substructure and Texture Evolution of TA15 Titanium Alloy," J. Mater. Eng. Perform., vol. 25, no. 8, pp. 3439–3447, 2016, doi: 10.1007/s11665-016-2173-6.
- [5] S. P. Lloyd, "Least Squares Quantization in PCM," IEEE Trans. Inf. Theory, vol. 28, no. 2, pp. 129–137, 1982, doi: 10.1109/TIT.1982.1056489.
- [6] Rish, "An Empirical Study of the Naïve Bayes Classifier An empirical study of the naive Bayes classifier," Cc.Gatech.Edu, no. January 2001, pp. 41–46, 2014, [Online]. Available: <https://www.cc.gatech.edu/~isbell/reading/papers/Rish.pdf>
- [7] R. M. Maertens, A. S. Long, and P. A. White, "Performance of the in vitro transgene mutation assay in MutaMouse FE1 cells: Evaluation of nine misleading ('False') positive chemicals," Environ. Mol. Mutagen., vol. 58, no. 8, pp. 582–591, 2017, doi: 10.1002/em.22125.
- [8] M. Li, D. He, C. Li, and S. Qi, "Brain-computer interface speller based on steady-state visual evoked potential: A review focusing on the stimulus paradigm and performance," Brain Sci., vol. 11, no. 4, 2021, doi: 10.3390/brainsci11040450.
- [9] R. Wolpaw, D. J. McFarland, G. W. Neat, and C. A. Forneris, "An EEG-based brain-computer interface for cursor control," Electroencephalogr. Clin. Neurophysiol., vol. 78, no. 3, pp. 252–259, 1991, doi: 10.1016/0013-4694(91)90040-B.