Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

MUQUN YU

Detecting Arm Movements from EEG-signal with Machine Learning

DEGREE PROGRAMME IN DATA ENGINEERING
2023

ABSTRACT

Muqun, Yu: Detecting Arm Movements from EEG signal with Machine Learning
Bachelor's thesis
Degree programme in data engineering
November 2023
Number of pages: 26

The integration of brain-computer interface (BCI) in human-computer interaction has started in a new time full of innovation and possibilities. BCI has overcome the limitations of traditional input devices and has provided direct communication between the human brain and external systems. Especially electroencephalogram (EEG)-based biometrics (BCI) have become a promising method for seamless data exchange between the human brain and applications. This paper explores how to use convolutional neural networks (CNN) and long short-term memory (LSTM) as a basic mechanism for EEG signal description, preprocessing of EEG data using ICA and MNE, and analysis to explore subtle differences in EEG signals of left and right arm movements and hypothesize the application of this technology to the steering system of the future self-propelled wheelchair.

Keywords: EEG, BCI, ICA, MNE, machine learning, neural Network

CONTENTS

LIST OF SYMBOLS AND TERMS

BCI – Brain Computer Interfaces

CNN – Convolutional Neural Network

EEG – Electroencephalography

EMG – Electromyography

ECG – Electrocardiography

ICA – Independent Component Analysis

OpenBCI – Open-Source Brain-Computer Interface

LSTM – Long short-term memory

AC – Alternating current

MRCP – Movement-Related Cortical Potentials

1 INTRODUCTION

With the growth of industrial society and increased human activities, more and more people are facing various mobility-related challenges. These challenges can be caused by several factors including accidents, rare diseases, aging, inherited diseases and heart diseases, these limitations lead to loss of mobility. Examples of this include stroke(Feigin et al., 2022) and spinal cord injury (SCI)(Albayar et al., 2019), both of which can result in paralysis and loss of lower limb mobility. The integration of EEG and BCI technology into wheelchairs provides a new solution. The EEG-BCI system allows patients to regain mobility and independence by controlling their wheelchairs using brain signals. EEG-BCI holds great promise for improving patients' quality of life with severe movement disorders. Advanced wheelchairs have been developed with this technology to meet the special needs of disabled and old people, allowing them to act independently.

This study aims to identify and classify the changes in the left and right arms on EEG using the CNN-LSTM neural network framework as an analytical tool, supplemented by MNE and ICA as a library of data preprocessing tools, based on the EEG signals. In Chapter 2, the details related to experimental data collection are described. In Chapter 3, the pre-processing phase of the experimental data is described. In Chapter 4, the core of the article is analysed in depth regarding the framework and processing of the neural network. Chapter 5 then presents and analyses the results of the experiment. In the end, Chapter 7 contains a plan and vision for the future improvement of the experiment.

2 COLLECTION AND PLANNING OF EXPERIMENTAL DATA

2.1 Description of the equipment used for experimental data collection

The Neoprene Headcap is a comfortable, reliable and flexible solution to place electrodes for EEG monitoring and stimulation. Its positioning grid with 39 predefined and clearly annotated positions is based on a subset of the international 10-20 EEG system.

Figure [1] Neoprene Headcap from https://www.neuroelectrics.com/solution/spareparts-consumables/cap

The pair of OpenBCI ear clip electrode cables with electrode material of silver chloride (Ag-AgCl). This electrode provides a stable ground signal while recording EEG data.

Figure [2] OpenBCI ear clip electrode cables from
https://shop.openbci.com/products/earclip-electrode

Boards and Dongles: The OpenBCI Cyton board is an 8-channel neural interface with a 32-bit processor. It has a large amount of local memory and fast processing speeds. Each of the eight channels samples data at 250Hz and acquires data and connects electrodes via the OpenBCI EMG/ECG Snap electrode cable. The Cyton dongle is used to connect the Cyton board to a computer via Bluetooth, allowing the data collected by the Cyton board to be transmitted and displayed on the OpenBCI program.

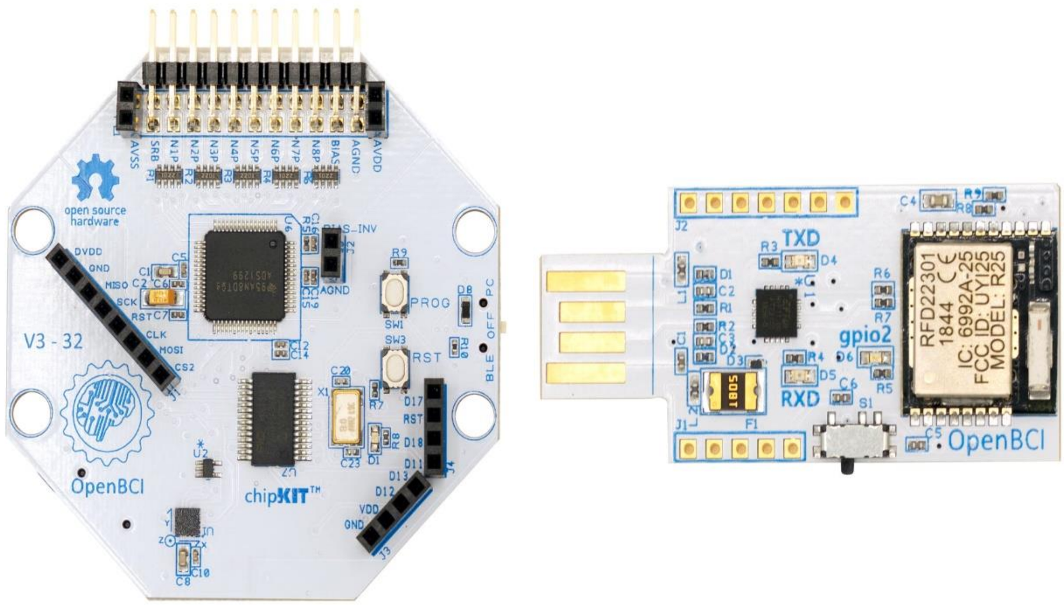


Figure [3] OpenBCI Cyton board(left) from
https://shop.openbci.com/products/cyton-biosensing-board-8-channel

Figure [4] OpenBCI Cyton dongle(right) from
https://shop.openbci.com/products/dongle

OpenBCI EMG/ECG Snap-In Electrode Cables are ribbon cables that can be attached to electrodes for use with any OpenBCI board.

Figure [5] OpenBCI EMG/ECG Snap-In Electrode Cables from
https://shop.openbci.com/products/emg-ecg-snap-electrode-cables

EEG Comb Electrodes and OpenBCI EMG/ECG Snap-In Electrode Cables need to work together.

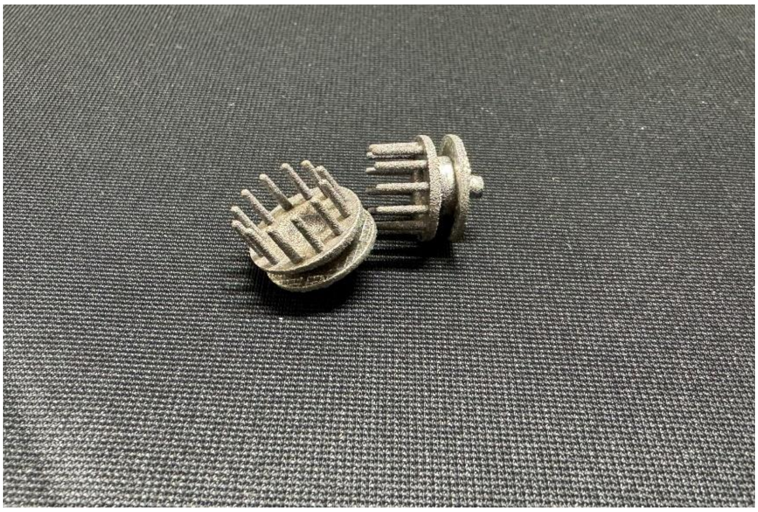


Figure [6] EEG Comb Electrodes

2.2 Experimental Data Collection Profile and Conditions

In this research, EEG data from different individuals were collected to ensure a decent level of generalization of the trained model and to strictly protect the privacy of the participants. The data will be used exclusively for research purposes and not be used for commercial applications. The foundation of our EEG-based biometric identification (BCI) system lies in the collection of high-quality EEG data. 8-single EEG electrodes was used to record the brain activity of the participants. The EEG signals are sampled at a rate of 250 Hz to capture real-time brain activity. Access to high-quality EEG data, participants were placed in a quiet, and strong light-free environment to make sure they stayed focused and undisturbed mental states.

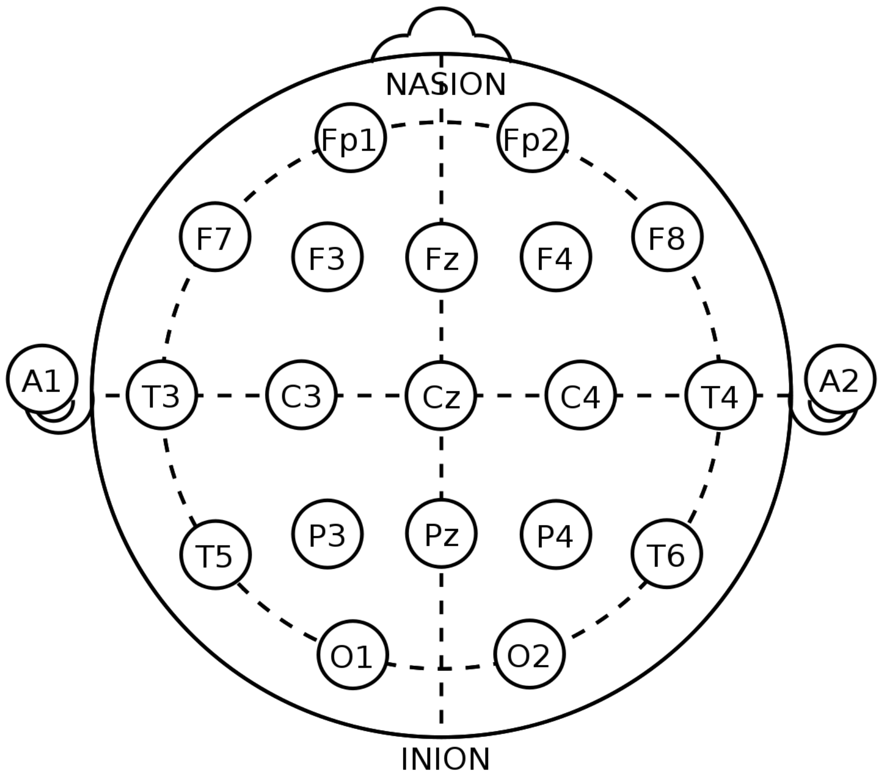


Figure [7] Electrode locations of International 10-20 system for EEG

The 10-20 system is a standardized method for placing EEG electrodes on the scalp. Electrodes are positioned at specific locations based on distances of 10% or 20% of the scalp's total front-to-back or right-to-left distance. The system is widely used in both clinical and research settings, ensuring consistency in electrode placement and allowing for comparisons across studies.

8-single EEG electrodes was used for the analysis (P3, P4, C3, C4, F3, F4, Cz, Pz), C3, C4 and Cz electrodes are typically used to extract MRCP signals (refers to specific electrical potentials in the brain that are associated with movement preparation and execution. These potentials can be recorded using EEG) for hand movement tasks(Yahya et al., 2019).

In addition, five other electrodes (F3, F4, P3, P4 and Pz) were explored in this experiment. In exploring other examples of EEG and Movement aspects, studies were found that used similar EEG electrode locations((FC3, FC4, CP3, CP4, and CPz) with the aim of investigating the effects of rehabilitation training on the EEG cortex in patients with movement disorders after stroke(Butt et al., 2020), which shows that the choice of electrode locations for the present experiment was reliable.

2.3 Procedure for experimental data collection

The collection plan consisted of the following steps: participants closed their eyes, sat comfortably in a quiet room without bright light, and the staff issued a clear but subtle voice command to start the task and start the timer. Upon hearing the instruction from the staff, the participant began to wave his right arm. They were instructed to maintain the wave motion until they heard a stop command, also delivered in voice, and the practice task lasted for 2 minutes. After completing the task of waving the right arm, participants rested for one minute and waited for the staff to issue the command to wave the left arm again. Similar to the previous task, participants continued to perform the action until the staff gave the command again, also for 2 minutes. A round starts with the right arm and ends with the left arm, and the total collection plan contains two rounds. The arm waving motion was specifically designed as the participant lifting their arm 90° to the side at a constant speed from its natural sagging position and dropping it. During this carefully designed process, we continuously recorded EEG data to capture the changing brain activity of participants as they engaged in movement tasks.

Figure [8] Collection process (this figure is only an example of collection, the real collection environment without bright light)

This approach allowed us to induce EEG activity related to left and right arm movements while keeping experimental controllability and consistency. Alternating between waving and resting allowed us to collect EEG data that was precisely locked to the time of the movement event, which laid a good foundation for our subsequent processing.

3 DATA PRE-PROCESSING

3.1 Use of independent component analysis

In this study, independent component analysis (ICA) (Rejer & Górski, 2015) played a key role in the pre-processing and analysis of EEG data. ICA is a powerful computing technique that decomposes mixed signals into statistically independent signal sources. ICA can identify and separates invalid data, such as blinking and muscle activity, from raw nerve signals. ICA is also able to remove unnecessary noise while preserving basic brain activity, all functions that improve the quality of neural data. ICA can also handle the visualization of neural data, clearly presenting potential neural sources and their temporal dynamics. These visualizations help to observe neural data and identification.

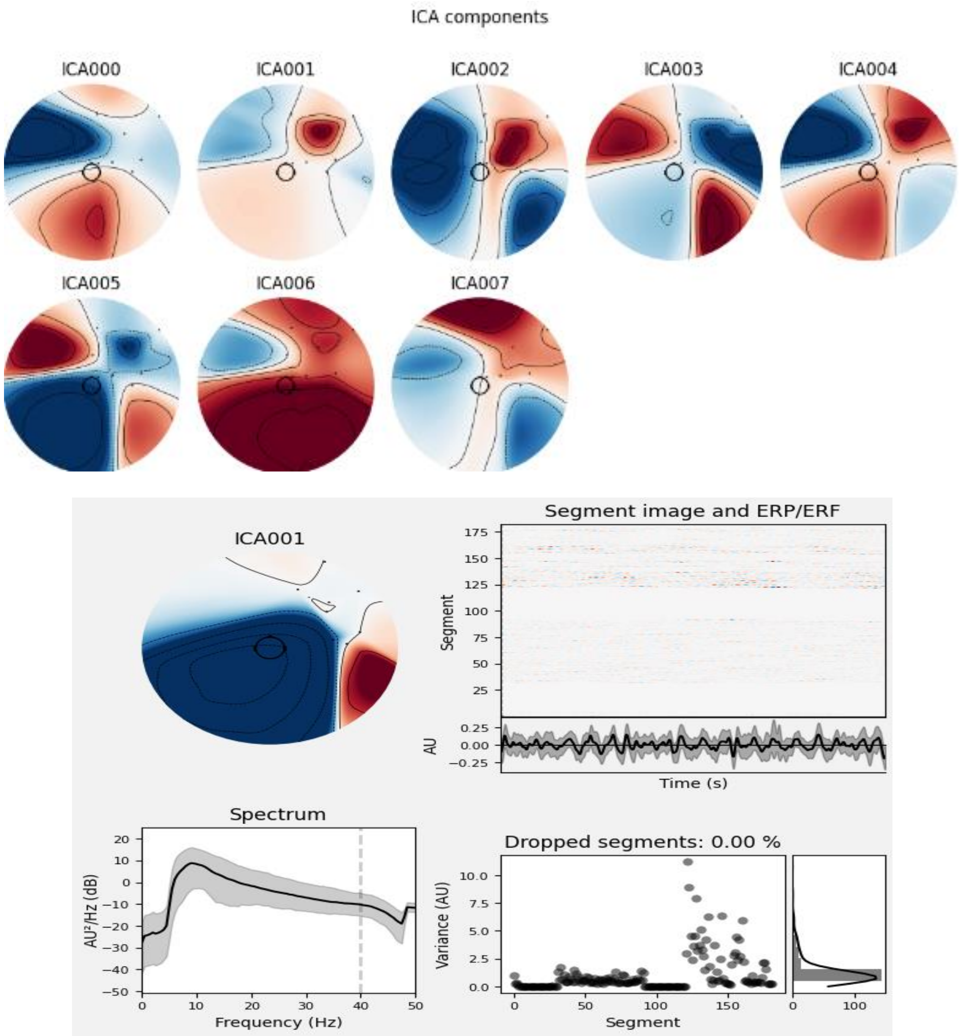


Figure [9&10] ICA's data visualization display

3.2 Filtering and noise reduction

In neuroscience research, it is common practice to select a filter with a frequency range of 1-30 Hz for EEG data analysis. To ensure the consistency of various studies, a frequency of 1-30 Hz will be used in this experiment. The spectrum of EEG signals ranges from 0.1 Hz to 100 Hz and is divided into five frequency bands: Delta (0.1-3 Hz), Theta (4-7 Hz), Alpha (8-13 Hz), Beta (14-30 Hz), and Gamma (31-100 Hz)(C.r et al., 2011). It was found that the α, β, and γ frequency bands (frequencies 8 Hz to 40 Hz) play an important role in recognizing motion signals recorded in areas of the sensory-motor cortex, with an average classification accuracy of 92%, and the highest classification accuracy using the α, β, and γ bands being 98.7%. The frequencies chosen for this experiment were approximately the same. EEG recordings can be contaminated by a variety of noise sources such as muscle activity, blinks, and environmental disturbances (e.g., alternating current (AC) noise at 50/60 Hz). By limiting the analysed frequency range to 1-30 Hz, noise components outside this range can be effectively attenuated, resulting in a purer signal.

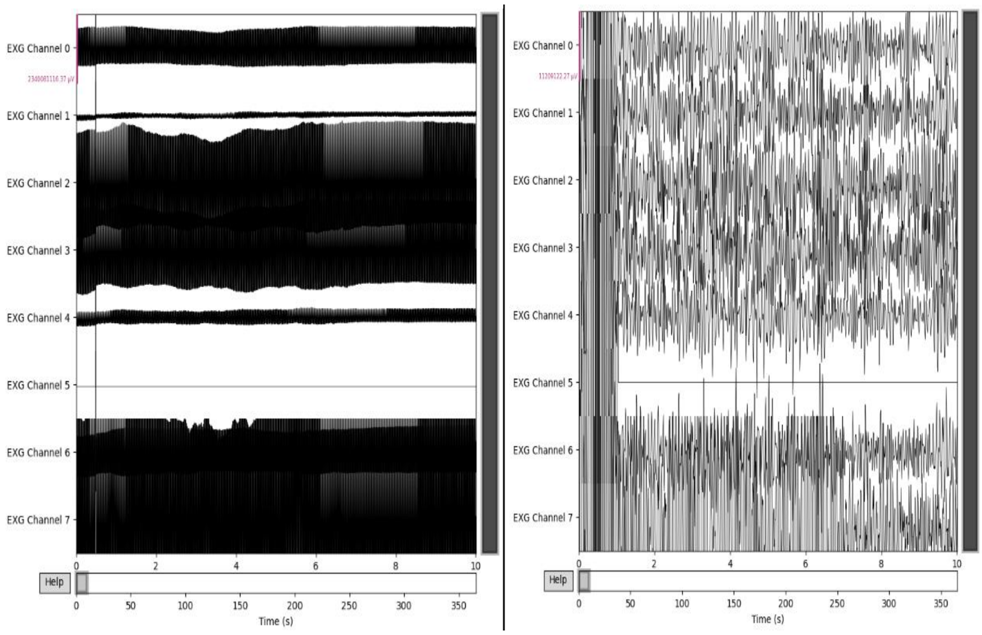


Figure [11&12] Comparison before and after filtration

3.3 Data labelling

To facilitate the precise labelling of EEG data, we employed the Annotations function from the MNE-Python library(Gramfort et al., 2014). EEG measures weak electromagnetic signals originating from neural currents in the brain.

MNE provides comprehensive analysis tools and workflows for EEG, including preprocessing capabilities. MNE is an academic software package designed to provide a data analysis pipeline covering all stages of EEG data processing. The comprehensive feature set provided by the MNE-Python package has been implemented by dedicated contributors working closely together from multiple institutions in multiple countries and facilitated through the use of a fully open-source software development process available for contribution by anyone. The comprehensive feature set provided by the MNE-Python package is enabled by the close collaboration of multiple organizations from several countries and through a software development process that is fully open source and guaranteed to be modifiable by anyone.

3.3.1 Event Generation

Using the "Annotations" function in MNE, events can be generated utilizing the data collected with the assurance that the arm movements have precise timing. Declare "event_id" as "left" for left arm movement and "right" for right arm movement. Finally, use the "events_from_annotations" function to generate the events.

```
event_start_time_right1 = 10
event_end_time_right1 = 130
event_start_time_left1 = 190
event_end_time_left1 = 310
event_start_time_right2 = 370
event_end_time_right2 = 490
event_start_time_left2 = 550
event_end_time_left2 = 670



# Create annotations for each segment
annotation_right1 = mne.Annotations(onset=event_start_time_right1, duration=event_end_time_right1 - event_start_time_right1,
                                   description='right')
annotation_left1 = mne.Annotations(onset=event_start_time_left1, duration=event_end_time_left1 - event_start_time_left1,
                                  description='left')

annotation_right2 = mne.Annotations(onset=event_start_time_right2, duration=event_end_time_right2 - event_start_time_right2,
                                   description='right')
annotation_left2 = mne.Annotations(onset=event_start_time_left2, duration=event_end_time_left2 - event_start_time_left2,
                                  description='left')
```

Figure [13] Dividing left and right arm movements according to time

```python
event_id = {'right': 1, 'left': 2}

# Generate events based on the combined annotations
events, _ = mne.events_from_annotations(raw, event_id=event_id, chunk_duration=1.0)
```

Figure [14] Generating events

The resulting events were marked at the onset of each arm movement task, allowing us to accurately divide the EEG data into classes that represent the movements of the left and right arms to facilitate the training of the model.

3.3.2 Epochs Generation

For the model to learn the subtle differences in the electroencephalogram (EEG) signals in the direction of the left and right arm movements, a key pre-processing step is used to convert the generated events into epochs. Subsequently, these epochs will be used as input data for the machine learning model. When transforming from events to epochs, the "Epochs" function in the MNE library is called to generate them. These event tokens act as timestamps that split continuous EEG recordings into discrete epochs. Each epoch contains EEG activity that precisely corresponds to the participant's arm.

```python
picks = ch_names
tmin = -0.5
tmax = 10
# epoch generation
epochs = mne.Epochs(raw, events, event_id, tmin, tmax, proj=True, picks=picks, baseline=(None, 0), preload=True)
```

Figure [15] Generating epochs

3.3.3 Summarization and subsequent pre-processing

Notably, this annotation process maintains the temporal accuracy of the EEG data, enabling the model to learn the changing of brain activity associated with left-arm and right-arm motion. By combining the annotation capabilities of the MNE library with our well-defined experimental protocols, we achieve accurate automatic annotation of EEG data segments. This rigorous annotation process lays the foundation for training our CNN model to accurately interpret the EEG signals associated with left-arm and right-arm motion.

The pre-processed and labelled EEG data is divided into a training set (70%) and a test set (30%). The training CNN divides the EEG time into two categories: left-arm and right-arm. Standard loss functions (CrossEntropyLoss) and optimization algorithms (Adam) were used to train the network. The performance of the model was continuously monitored on the validation set to prevent overfitting and ensure generalization. In the following sections, we delve into the architectural, training, and evaluation details of our CNN model.

4 NETWORK ARCHITECTURE

4.1 Previous Related Research

Convolutional Neural Networks (CNN) have previous precedents of application regarding EEG data analysis Motor Imagery. For example, the study(Li et al., 2022) utilized a convolutional neural network (CNN) to classify EEG data and explore the relationship between motor imagery and age-related fatigue. The study compared younger and older participants and analysed energy changes during motor imagery using time-frequency plots and event-related desynchronization (ERD) values. Fatigue from motor imagery was assessed using two metrics: $(\theta+\alpha)/\beta$ and θ/β, and fatigue-sensitive channels were identified in parietal regions of the brain. The study also introduced rhythmic entropy to analyse the complexity of cognitive activity and calculated phase-locked values associated with the parietal and frontal lobes. The motor imagery EEG data was then classified using CNNs and the accuracy of the classification is discussed. The results of the study showed that ERDs were observed in both young and old people, but the fatigue-sensitive channels in the parietal region were slightly different in the two groups. The accuracy was higher in young people than in older people, with a peak of 82.81%, while older people were generally around 70%. The study concludes that older adults are less affected by cognitive fatigue during motor imagery than younger adults, but the classification accuracy of motor imagery data may be slightly lower in older adults.

There is also a precedent for the use of CNN_LSTM neural networks in emotion recognition models(Ramzan & Dawn, 2023). The fusion of deep learning models (CNN and LSTM-RNN) has shown effective performance in classifying emotions based on electroencephalography (EEG). The average accuracy of the model on these five dimensions of the data High-Arousal-Low-Arousal (HALA), High-Valence-Low-Valence (HVLV), familiarity, Dominance and Liking emotions was as high as 97.39%, 97.41%, 98.21%, 97.68%, 97.89%, and 93.74% for positive and negative emotions.

4.2 Principles of CNN-LSTM

The CNN-LSTM architecture combines two powerful deep learning techniques: Convolutional neural networks (CNN)(Saxena, 2022) and Long Short-Term Memory (LSTM)(Yu et al., 2019) networks. CNNS are good at capturing spatial patterns in the data, so they are well suited for processing EEG signals with spatial and temporal features. LSTM is good at modelling temporal dependencies in continuous data. The CNNLSTM model combines the two and adopts a hierarchical feature extraction method to capture spatial and temporal patterns in EEG data.

EEG signals represent the bioelectrical activity of different brain regions over time. The CNN layer in the model processes EEG data from a temporal perspective and identifies features in the signal. These features are then passed to the LSTM layer, which handles the time aspect and captures these features over time. This combination allows the model to understand complex EEG signals. In addition, the LSTM portion of the model includes a culling layer to prevent overfitting. During training, the cull layer randomly removes some neuronal activation, which helps prevent the model from remembering the training data and encourages it to learn more generic features.

4.3 Explanation of neural network structure

The CNNLSTM model consists of several key components that are carefully organized to extract meaningful information from EEG data. The following is an in-depth discussion of its architectural elements.

The initial convolutional layer (conv1) is strategically configured to process complex EEG data and effectively reduces the number of input channels from 2626 to 1313 by means of a one-dimensional convolutional operation with a kernel size of one. Subsequent convolutional layers (conv2, conv3, conv4, conv5) build on this reduction, further refining the feature reproduction with the help of nested ReLU activation functions (relu1, relu2, relu3, relu4, relu5) in each layer to introduce fundamental nonlinearities into the model. These layers aim to capture the spatial information in the EEG signal. (Specific details of the model architecture are in Appendix 1)

After the final convolutional layer, the originality of our architecture is reflected in the inclusion of a batch normalization layer (batch_norm). This key component optimizes the stability of the network and speeds up training, finally resulting to the convergence of a more powerful and efficient model.

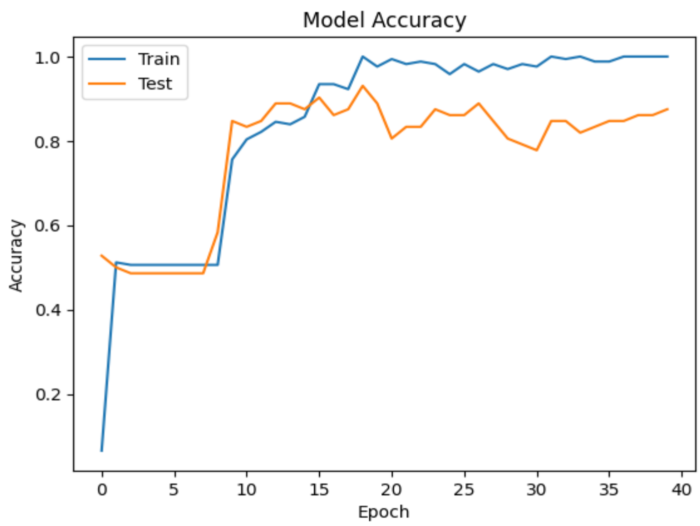
The key to the model is the integration of LSTM layers (lstm1, lstm2). These layers introduce temporal awareness into the model, making it possible to decipher complex patterns in EEG data over time. The first LSTM layer (lstm1) consists of 64 hidden units with 40% dropout rate set to enhance the generalization of the model and reduce overfitting. The second LSTM layer (lstm2) has 32 hidden units and builds on the background built by lstm1 to improve the extracted features. It also has a 40% dropout rate.

The Fully Connected Layer is placed at the end of the whole network, and it allows the integration of the knowledge extracted in the previous layers into actionable predictions. This layer provides five output units, which may include classification or continuous value predictions.

5 RESULTS FROM THE MODEL

5.1 Model Results Presentation

The training process of the model is shown in Fig. With these figures, we can see that the model converges and learns the image features during the training process. The model training step is 40 epochs and the final training accuracy is close to 1. The test accuracy is 0,87, the training loss is about 0.028 and the final validation loss is about 0.401.



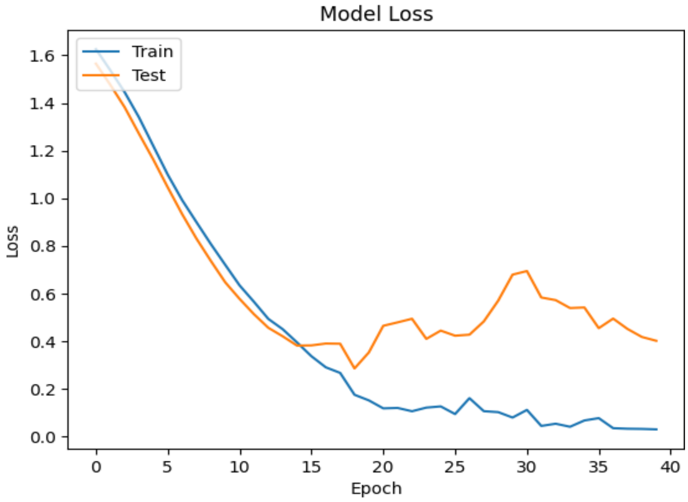


Figure [16&17] Displaying model accuracy and model loss

In addition to monitoring the model's performance through training metrics such as accuracy and loss rate, the experiment validated the model's learning degree using a new EEG signal dataset collected under the same conditions as the original dataset. This method measures the generalization ability of the model and confirms its reliability on unseen data. The new EEG data went through the same pre-processing steps as the training data, ensuring consistency in data preparation. After loading the pre-trained model, predictions are made on these new data. These predictions are then compared to the basic real labels obtained during the data collection process, so that the validity of the model in the real world is determined. By calculating the prediction accuracy of the model on the new data set, this is an important criterion to measure the generalization and learning performance of the model.

After several iterations, the structure of the model is improved according to the feedback, and it can be seen from the figure that the maximum accuracy can reach more than 50%.

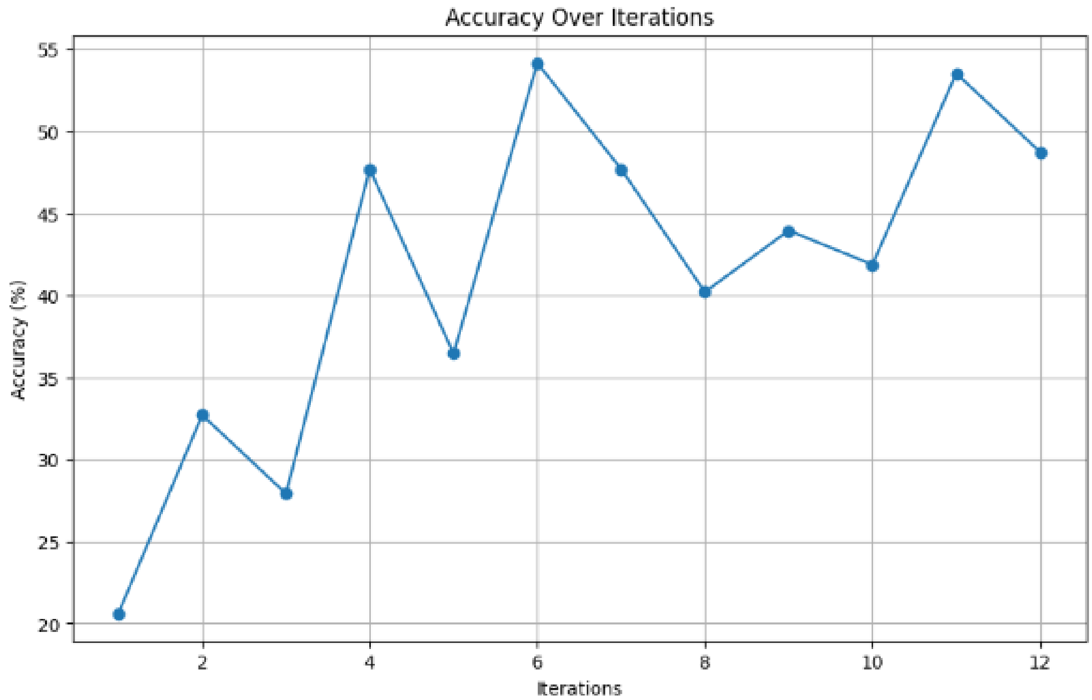


Figure [18] Statistics on the accuracy of the model on new data

5.2 Announcement

The frequency range of 1-30 Hz selected in this study is consistent with the convention of neuroscience research. While it may not be the most specific frequency range for capturing body motion-related artifacts, it represents a pragmatic option given the resources and instruments available. In addition, this frequency range is a reliable starting point for analysis. To explore narrower or more specialized frequency segments, it is best to seek the guidance and cooperation of experienced professionals and clinicians to ensure further refinement in a rigorous approach.

6 FUTURE PLANS

6.1 Online real-time forecasting

Extend the current offline prediction model to an online real-time prediction system. This requires adapting the model to continuously process incoming EEG data and provide instantaneous predictions. Real-time applications can be explored in areas such as BCI to control external devices in real-time.

6.2 Migration learning and adaptation

Migration learning techniques are investigated to improve model adaptation across different EEG datasets and subjects. This may include pre-training on large datasets and fine-tuning on subject-specific data. The goal is to improve the generalization ability of the model.

6.3 Multimodal integration

Explore the integration of other modalities, such as eye tracking or motion capture, to complement EEG data. Multimodal approaches can improve the accuracy of motion prediction and provide a richer understanding of user intent and actions.

6.4 Clinical application and validation

Collaborate with healthcare professionals to validate model performance in a clinical setting. Apply the model to patients with movement disorders to validate and evaluate the feasibility and accuracy of the developed system in assistive technology and rehabilitation.

7 CONCLUSION

In this thesis, we set out to explore the complex world of EEG signal processing and machine learning for the purpose of movement prediction. The convergence of neuroscience and computer science opens a field full of possibilities, offering promising solutions for people with movement disorders, neurorehabilitation, and brain-computer interfaces. We started with the acquisition and preprocessing of EEG data to lay the foundation for subsequent analysis. Subsequently, the introduction of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) layers provides a powerful model for motor prediction. In the "Future Plans" section, multiple directions that can be improved and implemented are presented, all of which represent opportunities to further strengthen the field and increase the utility of our findings. Finally, our exploration of EEG-based motion prediction shows the unlimited possibilities of neuroscience and machine learning.

REFERENCES

Albayar, A. A., Roche, A., Swiatkowski, P., Antar, S., Ouda, N., Emara, E., Smith, D. H., Ozturk, A. K., & Awad, B. I. (2019). Biomarkers in Spinal Cord Injury: Prognostic Insights and Future Potentials. *Frontiers in Neurology*, *10*. https://www.frontiersin.org/articles/10.3389/fneur.2019.00027

Butt, M., Naghdy, G., Naghdy, F., Murray, G., & Du, H. (2020). *Effect of Post-Stroke Motor Training on EEG Movement-Related Cortical Potentials* [Preprint]. In Review. https://doi.org/10.21203/rs.3.rs-50667/v1

C.r, H., M.p, P., S, Y., A.h, A., & R, N. (2011). An Analysis of the Effect of EEG Frequency Bands on the Classification of Motor Imagery Signals. *International Journal of Biomedical Soft Computing and Human Sciences: The Official Journal of the Biomedical Fuzzy Systems Association*, *16*(1), 121–126. https://doi.org/10.24466/ijbschs.16.1_121

Feigin, V. L., Brainin, M., Norrving, B., Martins, S., Sacco, R. L., Hacke, W., Fisher, M., Pandian, J., & Lindsay, P. (2022). World Stroke Organization (WSO): Global Stroke Fact Sheet 2022. *International Journal of Stroke*, *17*(1), 18–29. https://doi.org/10.1177/17474930211065917

Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Parkkonen, L., & Hämäläinen, M. S. (2014). MNE software for processing MEG and EEG data. *NeuroImage*, *86*, 446–460. https://doi.org/10.1016/j.neuroimage.2013.10.027

Li, X., Chen, P., Yu, X., & Jiang, N. (2022). Analysis of the Relationship Between Motor Imagery and Age-Related Fatigue for CNN Classification of the EEG Data. *Frontiers in Aging Neuroscience*, *14*. https://www.frontiersin.org/articles/10.3389/fnagi.2022.909571

Ramzan, M., & Dawn, S. (2023). Fused CNN-LSTM deep learning emotion recognition model using electroencephalography signals. *International Journal of Neuroscience*, *133*(6), 587–597. https://doi.org/10.1080/00207454.2021.1941947

Rejer, I., & Górski, P. (2015). Benefits of ICA in the Case of a Few Channel EEG. *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 7434–7437. https://doi.org/10.1109/EMBC.2015.7320110

Saxena, A. (2022). An Introduction to Convolutional Neural Networks. *International Journal for Research in Applied Science and Engineering Technology*, *10*(12), 943–947. https://doi.org/10.22214/ijraset.2022.47789

Yahya, N., Musa, H., Ong, Z. Y., & Elamvazuthi, I. (2019). Classification of Motor Functions from Electroencephalogram (EEG) Signals Based on an Integrated Method Comprised of Common Spatial Pattern and Wavelet Transform Framework. *Sensors*, *19*(22), Article 22. https://doi.org/10.3390/s19224878

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation*, *31*(7), 1235–1270. https://doi.org/10.1162/neco_a_01199

APPENDIX 1

```
class CNNLSTM(nn.Module):
    def __init__(self):
        super(CNNLSTM, self).__init__()
        self.conv1 = nn.Conv1d(2626, 1313, kernel_size=1 ,bias=False)
        self.relu = nn.ReLU()
        self.conv2 = nn.Conv1d(1313, 512, kernel_size=1)
        self.relu = nn.ReLU()
        self.conv3 = nn.Conv1d(512, 256, kernel_size=1)
        self.relu = nn.ReLU()
        self.conv4 = nn.Conv1d(256, 128, kernel_size=1)
        self.relu = nn.ReLU()
        self.conv5 = nn.Conv1d(128, 64, kernel_size=1)
        self.relu = nn.ReLU()
        self.batch_norm = nn.BatchNorm1d(64)
        self.lstm1 = nn.LSTM(64, 32, dropout=0.4, batch_first=True)
        self.lstm2 = nn.LSTM(32, 16, dropout=0.4, batch_first=True)
        self.fc = nn.Linear(16, 5)
```