PROJECT REPORT

<u>To develop a multimodal deep learning architecture for an EEG-EMG based Brain-Computer Interface (BCI) controlled robotic hand exoskeleton for neuro-rehabilitation of stroke patients.</u>

-Shatakshi Saxena
Computer Science Engineering
Thapar Institute of Engineering and Technology

TABLE OF CONTENTS

To develop a multimodal deep learning architecture for an EEG-EMG based Brain-Computer Interface (BCI) controlled robotic hand exoskeleton for neuro-rehabilitation of stroke patients.

Abstract

Every year, a significant portion of India's population is impacted by stroke, a leading cause of disability. Traditionally, Brain-Computer Interface (BCI) systems have relied on Electroencephalogram (EEG) signals for communication and control tasks. However, to address the challenge of lower accuracy rates in BCIs, recent advancements have seen the integration of various biological signals with EEG, resulting in hybrid BCI devices. By combining EEG and Electromyogram (EMG) signals through correlation factors, a hybrid BCI device can be developed to control a hand exoskeleton, offering new possibilities in neuro-rehabilitation. Brain-computer interfaces (BCIs) have the potential to provide neurofeedback for stroke patients to improve motor rehabilitation. However, current BCIs often only detect general motor intentions and lack the precise information needed for complex movement execution, mainly due to insufficient movement execution features in EEG signals. Therefore, leveraging a corticomuscular coactivation-based hybrid BCI approach, we aim to introduce a Multimodal Deep Learning Approach for EEG-EMG Correlation-Based Brain-Computer Interface Development.

Introduction

Stroke is a leading cause of long-term disability globally, often resulting in motor impairments that significantly impact the quality of life for survivors. With up to 80% of stroke survivors experiencing residual deficits in arm and hand movement, effective rehabilitation is crucial, particularly within the first six months post-stroke when brain plasticity mechanisms are most active. Traditional rehabilitation methods, however, often lack efficiency and patient cooperation, necessitating innovative approaches to address these challenges. Robotic-assisted therapy has emerged as a promising avenue, offering precise, consistent rehabilitation training tailored to individual needs. Soft hand robots, in particular, present advantages over conventional rigid devices, providing a comfortable, adaptable interface for stroke patients. Yet, the optimal control strategy for these robots remains an ongoing challenge. Brain-Computer Interfaces (BCIs) offer a potential solution by translating brain activity into commands for controlling robotic systems. However, existing BCI systems, whether based on EEG, EOG, or EMG signals, possess limitations that hinder their clinical application, including inadequate spatial resolution, signal attenuation, and user discomfort.

In response, Multimodal Deep Learning Architectures have emerged as a promising approach to address these challenges. Techniques such as Generative Adversarial Networks (GANs), Graph Neural Networks (GNNs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs) offer powerful tools for analyzing and interpreting multimodal EEG-EMG data. By integrating these advanced algorithms, we aim to enhance the accuracy, reliability, and adaptability of BCI-controlled robotic systems, ultimately aiming to revolutionize stroke rehabilitation.Multi-modal Human-Machine Interfaces (mHMI) have been proposed to overcome these limitations by combining different signal acquisition modalities, such as EEG, and EMG, to enhance classification accuracy and increase the number of controllable commands. Despite advancements, a practical, mature mHMI method integrating

EEG, and EMG for real-time control of robot-assisted systems in stroke rehabilitation remains elusive. Building on recent research in mHMI, this study introduces a novel two-mode interface allowing stroke survivors to control soft hand robots through motor imagery, hand gestures, and eye movements. By leveraging high-performance mHMI, this research represents a critical step towards developing an effective, practical training system for motor functional recovery in stroke rehabilitation.

Objectives:

To address the challenges in traditional Brain-Computer Interfaces (BCIs) and enhance stroke rehabilitation outcomes, this research aims to achieve several objectives. First, the development of a novel BCI system integrating Electroencephalogram (EEG) and Electromyogram (EMG) signals through correlation factors is pursued. Utilizing a multimodal deep learning approach, the research seeks to analyze and interpret EEG-EMG correlation to provide accurate and real-time neurofeedback to stroke patients, thereby improving neuro-rehabilitation outcomes. Additionally, the research aims to address the limitations of current BCIs by enhancing the detection and execution of complex movement intentions using corticomuscular coactivation-based techniques. Furthermore, the research endeavors to explore the potential of synergistic integration of diverse neural and muscular signals to revolutionize rehabilitative technology and enhance the quality of life for stroke survivors.

In achieving these objectives, the proposed solutions offer several benefits. By leveraging both EEG and EMG signals, the multimodal approach enhances the accuracy and reliability of BCIs through the effective learning of complex patterns using deep learning algorithms. This will result in a BCI system that is adaptable and personalized, capable of adjusting to individual user characteristics and changes over time. Moreover, the non-invasive nature of EEG and EMG signals ensures the suitability of the multimodal approach for widespread use in neuro-rehabilitation settings while preserving user privacy and data security. Additionally, the EEG-EMG correlation-based BCI seamlessly integrates with existing rehabilitation protocols and therapeutic interventions, facilitating its adoption in clinical practice while addressing ethical and privacy concerns through encryption techniques and adherence to ethical guidelines.

Datasets for Research

EEG Motor Movement/Imagery Dataset:
The EEG Motor Movement/Imagery Dataset encompasses over 1500 EEG recordings obtained from 109 volunteers. These recordings include both one- and two-minute sessions and were collected as part of experiments focusing on motor and imagery tasks. The experimental protocol involved subjects performing various motor/imagery tasks while their brain activity was recorded using the BCI2000 system with 64-channel EEG. Each subject completed 14 experimental runs, consisting of baseline runs with eyes open and closed, as well as runs for tasks involving hand and foot movements and their corresponding mental imagery.

BCI Competition IV:
BCI Competition IV presents challenges aimed at validating signal processing and classification methods for Brain-Computer Interfaces (BCIs). It addresses several pertinent issues crucial for practical BCI systems, including:

Classification of continuous EEG without trial structure.
Classification of EEG signals affected by eye movement artifacts.
Classification of the direction of wrist movements from MEG.
Discrimination requiring fine-grained spatial resolution in ECoG.
These datasets provide diverse scenarios to test the efficacy of signal processing and classification algorithms in real-world BCI applications.

Motor-Imagery Left/Right Hand MI Dataset:
This dataset includes recordings from 52 subjects, with 38 subjects validated for containing discriminative features. It provides not only the EEG data but also results of physiological and psychological questionnaires, EMG datasets, the location of 3D EEG electrodes, and EEGs for non-task-related states. The dataset aims to facilitate research into motor imagery tasks, particularly focusing on left/right hand motor imagery.

EMG Data for Gestures:
For EMG recordings, the dataset includes raw data from 36 subjects performing static hand gestures. The data was collected using a MYO Thalmic bracelet equipped with eight sensors placed around the forearm. Each subject performed two series of gestures, each consisting of six to seven basic gestures, with each gesture lasting 3 seconds and a pause of 3 seconds between gestures. This dataset provides valuable insights into muscle activity during hand gestures, which is crucial for developing effective control strategies for the robotic hand exoskeleton.

These datasets offer a rich resource for developing and evaluating the multimodal deep learning architecture for the brain-computer interface controlled robotic hand exoskeleton. They cover a wide range of scenarios and modalities, allowing for comprehensive analysis and development of effective rehabilitation techniques for stroke patients.

Literature Review:

S.No	Paper	Summary	Limitation
[1]	An EEG-EMG correlation-based brain-computer interface for hand orthosis supported neuro-rehabilitation	Introduces a new method based on EEG-EMG correlation for BCI control, showing improved performance for hand orthosis triggering tasks in healthy individuals and hemiplegic patients.	Limited sample size and lack of long-term evaluation in clinical settings may restrict generalizability to broader neuro-rehabilitation populations and contexts.
[2]	Investigating the feasibility of combining EEG and EMG for controlling a hybrid human computer interface in patients with spinal cord injury	Examines the feasibility of combining EEG and EMG for controlling a hybrid HCI in individuals with spinal cord injury, showing superior performance compared to EEG alone.	The study primarily focuses on offline classification and does not address real-time control challenges or long-term usability in real-world rehabilitation scenarios.
[3]	Corticomuscular Co-Activation Based Hybrid Brain-Computer Interface for Motor Recovery Monitoring	Evaluates a corticomuscular co-activation based BCI for motor recovery monitoring in stroke patients, demonstrating its potential as a biomarker for neurorehabilitation.	The study lacks a control group and larger sample size, limiting the ability to draw definitive conclusions about the clinical effectiveness and generalizability of the proposed BCI approach.
[4]	A sequential learning model with GNN for EEG-EMG-based stroke rehabilitation BCI	Proposes a sequential learning model incorporating Graph Isomorphic Networks for EEG-EMG-based stroke rehabilitation BCIs, achieving more accurate prediction results and execution quality scores for complex movements.	The study primarily focuses on model performance and does not address real-world implementation challenges or user feedback, limiting its applicability to practical neurorehabilitation settings

Methodology:

An EEG-EMG cortivascular-based Brain Computer Interface (BCI) enables people to communicate with the outside world by interpreting the EEG-EMG signals of their brains to interact with devices such as wheelchairs and intelligent robots. More specifically, motor imagery EEG (MI-EEG), which reflects a subjects active intent, is attracting increasing attention for a variety of BCI applications. Accurate classification of MI-EEG signals while essential for effective operation of BCI systems, is challenging due to the significant noise inherent in the signals and the lack of informative correlation between the signals and brain activities. A novel deep neural network based learning framework that affords perceptive insights into the relationship between the MI-EEG data and brain activities.

Data Preprocessing and Augmentation:

Data preprocessing and augmentation are crucial steps in developing a multimodal deep learning architecture for EEG-EMG based BCI systems. These steps aim to ensure that the input signals are of high quality, free from artifacts, and diverse enough to train robust models. In this section, we will discuss in detail the preprocessing and augmentation techniques applied to EEG and EMG data.

Data Preprocessing:

1. Artifact Removal: EEG and EMG signals are often contaminated with artifacts such as noise from electrical equipment, muscle artifacts, and eye blinks. Various artifact removal techniques, such as Independent Component Analysis (ICA) and template matching, are employed to identify and remove these artifacts from the raw data.

2. Filtering: Filtering is performed to remove unwanted frequencies from the signals. Low-pass, high-pass, band-pass, and notch filters are applied to isolate specific frequency bands of interest and remove noise from the signals.

3. Baseline Correction: Baseline correction involves adjusting the signal baseline to zero or a predefined reference level. This correction helps in standardizing the signals and removing any DC offset.

4. Normalization: Normalization is applied to scale the signals to a common range, typically between 0 and 1 or -1 and 1. This ensures that all input features have a similar scale, which is essential for the effective training of deep learning models.

Data Augmentation:

Data augmentation techniques are employed to increase the diversity of the training dataset, thereby improving the generalization ability of the model. In the context of EEG and EMG data, the following augmentation techniques are commonly used:

1. Adding Noise: Random noise is added to the signals to simulate real-world variability. This helps the model to become more robust to noisy input data.

2. Signal Shifting: EEG and EMG signals are shifted in time or frequency to introduce variations in the data. This allows the model to learn from different temporal and spectral characteristics of the signals.

3. Generating Synthetic Samples: Generative Adversarial Networks (GANs) are used to generate synthetic EEG-EMG samples that closely resemble real data. These synthetic samples are generated by training a generator network to produce realistic samples while simultaneously training a discriminator network to distinguish between real and synthetic samples.

Mathematically, the preprocessing and augmentation steps can be represented as follows:

$$X_{\text{processed}} = \text{Preprocessing}(X_{\text{raw}})$$

$$X_{\text{augmented}} = \text{Augmentation}(X_{\text{processed}})$$

where Xraw represents the raw EEG and EMG data, Xprocessed represents the preprocessed data, and Xaugmented represents the augmented data.

These steps ensure that the input signals are of high quality, diverse, and representative of real-world scenarios, ultimately leading to more accurate and robust BCI systems.

Autoencoder Training:

Autoencoders are fundamental neural network architectures used for unsupervised feature learning and dimensionality reduction of EEG and EMG data in the context of BCI systems. They consist of an encoder network that compresses input signals into a lower-dimensional representation, and a decoder network that reconstructs the original signals from this representation. In this section, we will delve into the details of autoencoder training and its mathematical representation.

Architecture Overview:

An autoencoder architecture typically consists of two main components:

1. Encoder: The encoder network takes the input signals (augmented EEG and EMG data) and maps them to a lower-dimensional latent space representation. This is achieved through a series of hidden layers with progressively fewer neurons, resulting in a compressed representation of the input signals.

2. Decoder: The decoder network takes the encoded representation from the encoder and reconstructs the original input signals. It mirrors the architecture of the encoder, with layers progressively expanding in size to decode the compressed representation back to the original signal space.

Training Process:

The training process of an autoencoder involves optimizing the model parameters to minimize the reconstruction error between the input and output signals. This is typically achieved using gradient descent optimization techniques, such as stochastic gradient descent (SGD) or Adam optimization.

1. Encoding:

$$X_{\text{encoded}} = \text{Encoder}(X_{\text{augmented}})$$

Here, X_encoded represents the encoded representation of the input signals obtained from the encoder network.

2. Decoding:

$$X_{\text{decoded}} = \text{Decoder}(X_{\text{encoded}})$$

X_decoded represents the reconstructed signals obtained from the decoder network using the encoded representation.

3. Autoencoder Loss:

$$L_{\text{autoencoder}} = \text{Loss}(X_{\text{augmented}}, X_{\text{decoded}})$$

The autoencoder loss L_autoencoder measures the discrepancy between the original input signals and their reconstructions. Common loss functions used include mean squared error (MSE) or binary cross-entropy, depending on the nature of the data and the reconstruction task.
Autoencoder training plays a crucial role in learning meaningful representations of EEG and EMG data for subsequent tasks in BCI systems. By compressing the input signals into a lower-dimensional space and reconstructing them accurately, autoencoders enable efficient representation learning and dimensionality reduction, contributing to the overall effectiveness of the multimodal deep learning architecture.

GAN Training for Synthetic Sample Generation:

Generative Adversarial Networks (GANs) are a class of deep learning architectures used for generating synthetic data that closely resembles real data. In the context of EEG-EMG based Brain-Computer Interface (BCI) systems, GANs play a crucial role in augmenting the training dataset with realistic synthetic samples. Let's explore the training process of GANs and its mathematical representation in detail.

Architecture Overview:

A GAN consists of two main components:

1. Generator: The generator network takes random noise as input and generates synthetic samples that mimic real data. It learns to map the input noise vector to the data space of EEG-EMG signals.

2. Discriminator: The discriminator network acts as a binary classifier that distinguishes between real and synthetic samples. It learns to differentiate between samples generated by the generator and real samples from the training dataset.

Training Process:

GANs are trained through an adversarial process where the generator and discriminator networks compete against each other The generator takes random noise vectors Z as input and generates synthetic samples synthetic X. The discriminator evaluates the generated samples and assigns a probability score to each sample being real or synthetic. The generator aims to produce samples that are indistinguishable from real samples, i.e., it tries to maximize the probability of the discriminator being fooled (producing a high probability score for synthetic samples).The discriminator is trained with both real samples real X from the training dataset and synthetic samples generated by the generator. It learns to correctly classify real samples as real and synthetic samples as synthetic, i.e., it aims to minimize the classification error. The discriminator loss consists of two components: one for synthetic samples and one for real samples.

Generating Synthetic Samples:
synthetic_X = Generator(Z) where, Z represents the input noise vector, and synthetic_X represents the synthetic samples generated by the generator.

Generator Loss:
generator = Loss(synthetic, real). The generator loss measures how well the generator fools the discriminator into classifying synthetic samples as real.

Discriminator Loss:
discriminator = Loss(synthetic, synthetic) + Loss(real, real). The discriminator loss consists of two components: one for synthetic samples and one for real samples. It aims to minimize the classification error between real and synthetic samples.

GAN training for synthetic sample generation is a crucial component of the multimodal deep learning architecture for EEG-EMG based BCIs. By generating realistic synthetic samples, GANs augment the training dataset, improving the model's robustness and generalization ability, ultimately enhancing the performance of the BCI system for neuro-rehabilitation of stroke patients.

Attention Mechanisms Integration:

Attention mechanisms play a crucial role in enhancing the interpretability and performance of deep learning architectures, especially in tasks involving sequential data like EEG and EMG signals. In the context of developing a multimodal deep learning architecture for an EEG-EMG based Brain-Computer Interface (BCI) controlled robotic hand exoskeleton for neuro-rehabilitation of stroke patients, attention mechanisms are integrated to selectively focus on relevant segments of EEG and EMG signals. Let's delve into the details of how attention mechanisms are incorporated and their impact on the model. Attention mechanisms allow the model to dynamically weight different parts of the input data, enabling it to focus more on relevant segments while disregarding irrelevant information. This selective attention mechanism mimics human cognitive processes, where attention is directed towards salient features or regions of interest.

Integration into Deep Learning Architecture:
Encoder Output Processing: The outputs of the encoder networks, which capture the representations of EEG and EMG signals, are passed through attention mechanisms. This allows the model to assign importance scores to different parts of the encoded representations.

Context Vector Calculation: Based on the attention scores assigned to different parts of the encoded representations, a context vector is computed. This context vector represents a weighted combination of the encoder outputs, where the weights are determined by the attention scores.

Integration with Downstream Modules: The context vector is then integrated with downstream modules of the architecture, such as classification or regression layers. This allows the model to utilize the selective attention information during the final decision-making process.

Attention Scores = Attention(Encoder Outputs)
Context Vector = Weighted Sum(Encoder Outputs, Attention Scores)
Final Output = Decision Module(Context Vector)

<u>Impact on Model Performance:</u>

Incorporating attention mechanisms into the deep learning architecture enhances the model's interpretability and performance in several ways:

1. Selective Focus: The model learns to selectively focus on relevant segments of EEG and EMG signals, improving its ability to capture salient features associated with motor intentions.
2. Robustness to Noise: By disregarding irrelevant information, attention mechanisms help the model become more robust to noise and artifacts present in the input signals.
3. Improved Performance: The selective attention information provides valuable context for downstream modules, leading to improved classification or regression performance in tasks such as gesture recognition or motor intention decoding.

Integrating attention mechanisms into the multimodal deep learning architecture enhances the model's ability to extract meaningful information from EEG and EMG signals, ultimately improving its performance in controlling the robotic hand exoskeleton for neuro-rehabilitation of stroke patients. By selectively focusing on relevant segments of the input data, attention mechanisms contribute to the interpretability and efficacy of the BCI system, facilitating more precise and responsive control of the robotic exoskeleton during neuro-rehabilitation sessions.

Graph Neural Networks (GNNs) Integration:

In the context of developing a multimodal deep learning architecture for an EEG-EMG based Brain-Computer Interface (BCI) controlled robotic hand exoskeleton for neuro-rehabilitation of stroke patients, Graph Neural Networks (GNNs) are integrated to capture spatial dependencies between EEG and EMG channels. Let's explore in detail the procedure of integrating GNNs into the deep learning architecture and how they facilitate effective information integration and decision-making processes. GNNs are a class of deep learning models designed to operate on graphs or structured data. In the context of EEG and EMG signals, the channels can be represented as nodes in a graph, and the connections between channels can be represented as edges. GNNs allow the model to propagate information across the graph, enabling it to capture spatial dependencies and correlations between different signal sources.

<u>Integration into Deep Learning Architecture:</u>

Graph Construction: EEG and EMG channels are represented as nodes in a graph, and the connections between channels are defined as edges. The adjacency matrix of the graph is constructed based on the spatial relationships between channels, such as physical proximity or functional connectivity.

Node Embedding: Each node (channel) in the graph is associated with an initial feature vector representing the corresponding EEG or EMG signal. These initial feature vectors serve as node embeddings and are input to the GNN model.

Message Passing: The GNN model iteratively updates the node embeddings by aggregating information from neighboring nodes. This process, known as message passing, enables the model to capture spatial dependencies between EEG and EMG channels.

Graph Pooling: After several iterations of message passing, the node embeddings are aggregated to obtain a graph-level representation. Graph pooling techniques, such as max pooling or mean pooling, can be used to summarize the information across all nodes in the graph.

Integration with Downstream Modules: The graph-level representation obtained from the GNN is then integrated with downstream modules of the architecture, such as classification or regression layers. This allows the model to utilize the spatial dependencies captured by the GNN during the final decision-making process.

Graph Construction: A = Adjacency Matrix(EEG, EMG)
Node Embedding: H_0 = Initial Node Embeddings(EEG, EMG)
Message Passing: H_t = Message Passing(H_{t-1}, A)
Graph Pooling: h = Graph Pooling(H_t)
Integration with Downstream Modules: Final Output} = Decision Module(h)

<u>Impact on Model Performance:</u>

Integrating GNNs into the deep learning architecture enables the model to capture spatial dependencies between EEG and EMG channels, leading to several benefits:

1. Spatial Context: GNNs provide the model with spatial context by considering the relationships between different signal sources, improving its understanding of the overall structure of the input data.
2. Effective Information Integration: By propagating information across the graph, GNNs facilitate effective integration of information from multiple channels, enhancing the model's ability to make informed decisions.
3. Improved Decision-Making: The spatial dependencies captured by GNNs contribute to more accurate and robust decision-making processes, leading to improved performance in tasks such as gesture recognition or motor intention decoding.

EEG Data Processing and Generation

1. **Data Pre-processing:**

- Subsampling and Cropping : Subsampled and cropped the EEG data to increase the dataset size and focus on relevant time windows. Subsampling reduced the frequency, while cropping allowed focusing on specific segments of the signal.

- Continuous Wavelet Transform (CWT) : Applied CWT to the EEG data to extract time-frequency features, useful for analyzing brain activity.

2. CNN Architectures:

- Shallow CNN : Implemented a shallow CNN based on time-based EEG data to classify motor imagery tasks. This model focused on extracting temporal features.

- CNN for CWT : Designed a CNN specifically for CWT data, treating the transformed EEG signals like images to extract both temporal and frequency features.

3. Results Analysis

- Evaluated the performance of different data augmentation techniques and architectures. Found that sequential cropping improved accuracy while random cropping and subsampling didn't.

Artificial Data Generation with GAN and VAE

4. GAN Implementation:

- Architectures : Created three GAN variants (DCGAN, WGAN, WGAN-GP) with a focus on generating synthetic CWT EEG signals.
  - DCGAN : Used a standard GAN approach with binary cross-entropy loss but suffered from mode collapse.
  - WGAN : Used Wasserstein loss for better convergence but still faced issues.
  - WGAN-GP : Introduced gradient penalty to stabilize training, which successfully generated plausible EEG data.

5. VAE Implementation:

- CNN VAE : Developed a convolutional VAE to encode and decode EEG signals. The VAE faced mode collapse, leading to all channels converging to the same signal, thus failing to generate diverse artificial data.

Classification with Artificial Data

6. Augmenting Data with Synthetic Signals

- Training with Synthetic Data : Integrated synthetic EEG signals generated by WGAN-GP into the training set. Found that adding 25% artificial data improved classification accuracy, while adding more started to deteriorate performance due to overfitting on synthetic features.

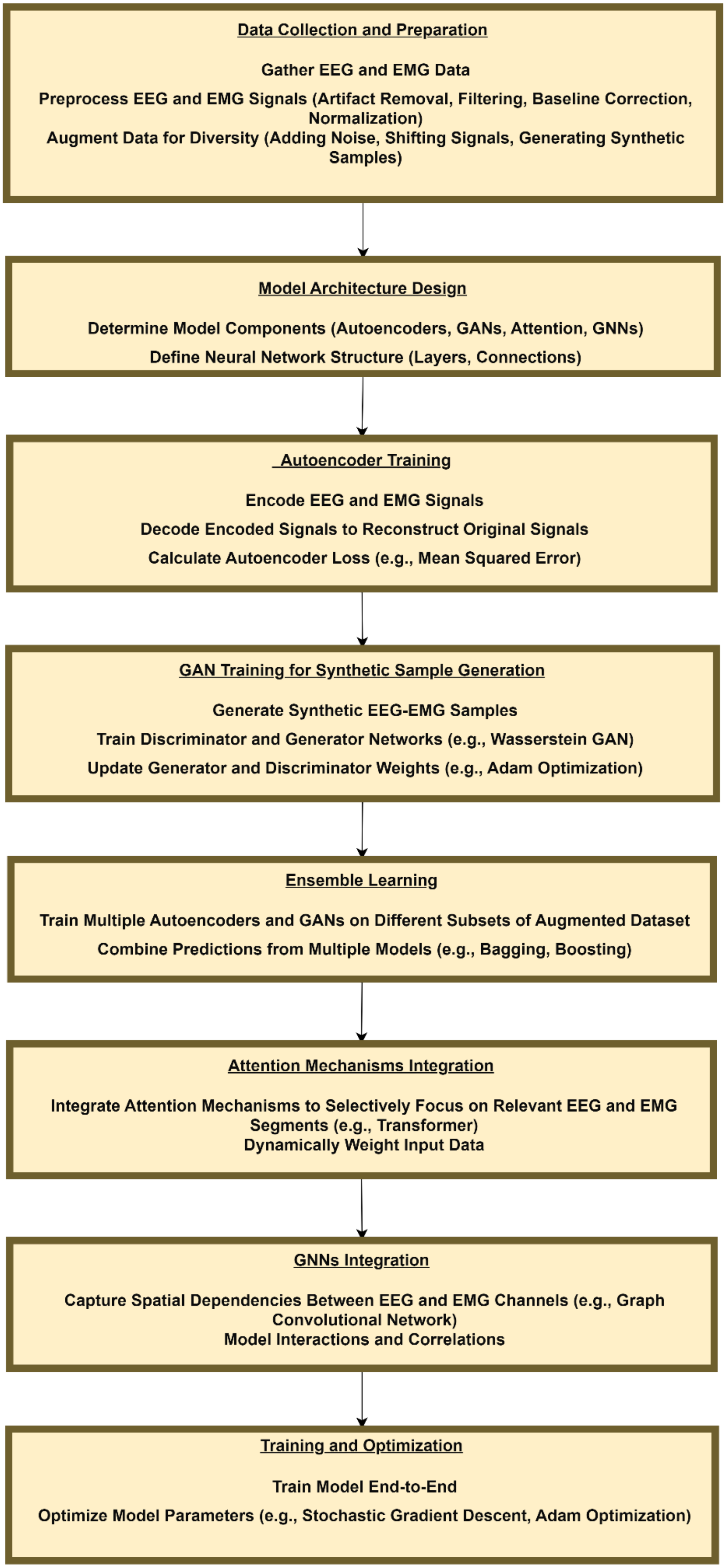
EMG Data Processing and Generation

7. EMG-GAN Implementation

- DCGAN for EMG : Focused on generating synthetic EMG signals using a DCGAN. Configured to handle specific window sizes for EMG signals.

1. **Pre-processing Techniques** : Enhanced dataset diversity and focused on relevant signal segments through subsampling, cropping, and CWT.
2. **Model Architectures** : Designed CNN architectures tailored to raw and transformed EEG data, improving classification accuracy.
3. **Generative Models** : Implemented GANs and VAEs to generate artificial EEG and EMG data, with WGAN-GP showing success in producing realistic signals.
4. **Data Augmentation** : Demonstrated that augmenting training data with synthetic signals can improve classification performance but needs careful balancing to avoid overfitting.

This theoretical understanding outlines the steps taken to generate and utilize artificial EEG and EMG data for improving motor imagery classification, highlighting the successes and challenges encountered along the way.


Data Collection and Preparation
Gather EEG and EMG Data
Preprocess EEG and EMG Signals (Artifact Removal, Filtering, Baseline Correction, Normalization)
Augment Data for Diversity (Adding Noise, Shifting Signals, Generating Synthetic Samples)
Model Architecture Design
Determine Model Components (Autoencoders, GANs, Attention, GNNs)
Define Neural Network Structure (Layers, Connections)
Autoencoder Training
Encode EEG and EMG Signals
Decode Encoded Signals to Reconstruct Original Signals
Calculate Autoencoder Loss (e.g., Mean Squared Error)
GAN Training for Synthetic Sample Generation
Generate Synthetic EEG-EMG Samples
Train Discriminator and Generator Networks (e.g., Wasserstein GAN)
Update Generator and Discriminator Weights (e.g., Adam Optimization)
Ensemble Learning
Train Multiple Autoencoders and GANs on Different Subsets of Augmented Dataset
Combine Predictions from Multiple Models (e.g., Bagging, Boosting)
Attention Mechanisms Integration
Integrate Attention Mechanisms to Selectively Focus on Relevant EEG and EMG Segments (e.g., Transformer)
Dynamically Weight Input Data
GNNs Integration
Capture Spatial Dependencies Between EEG and EMG Channels (e.g., Graph Convolutional Network)
Model Interactions and Correlations
Training and Optimization
Train Model End-to-End
Optimize Model Parameters (e.g., Stochastic Gradient Descent, Adam Optimization)

Explainable AI Models for Interpretation-:

Layerwise Relevance Propagation (LRP)

Layerwise Relevance Propagation (LRP) is a technique in deep learning designed for interpretability by assigning relevance scores to individual neurons in a neural network. LRP aids in understanding model decisions, making it particularly valuable in applications where transparency and interpretability are crucial, such as developing a multimodal deep learning architecture for an EEG-EMG based Brain-Computer Interface (BCI) controlled robotic hand exoskeleton for neuro-rehabilitation of stroke patients.

1. Initialization of Relevance: Relevance scores are initialized at the output layer of the neural network, representing the relevance of the network's prediction to each output neuron.

2. Propagation of Relevance: Relevance is propagated backward through the network based on neuron contributions. This involves distributing relevance from each neuron to its input neurons, guided by the weights of the connections between them.

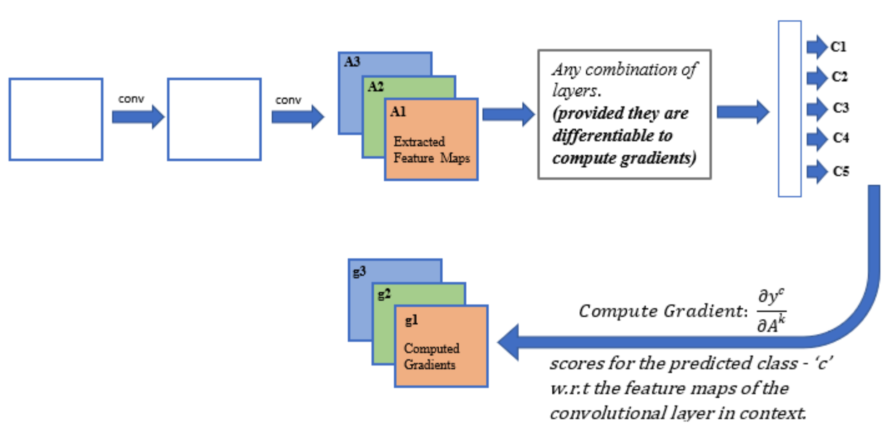
3. Conservation of Relevance: Throughout the propagation process, various redistribution rules are applied to ensure the conservation of relevance. These rules prevent vanishing or exploding relevance, allowing for meaningful interpretation of the model's decisions.

Grad-CAM (Gradient-weighted Class Activation Mapping)

Grad-CAM is another technique used in deep learning for visualizing and understanding the decision-making process of a neural network, particularly in the context of convolutional neural networks (CNNs). Like LRP, Grad-CAM is valuable for enhancing transparency and interpretability in the development of BCI systems for neuro-rehabilitation of stroke patients.

1. Gradient Computation: Grad-CAM leverages the gradients of the predicted class score with respect to the feature maps of the last convolutional layer in the network. These gradients quantify the importance of each feature map to the final prediction.

2. Weighted Combination: By computing the gradients and combining them with the feature maps using element-wise multiplication, Grad-CAM produces a heatmap that highlights the importance of different spatial locations in the feature maps. This heatmap visually represents the regions that strongly influence the final prediction.



RESULTS:

The analysis involves examining the relationships between EEG (electroencephalogram), EMG (electromyogram), and ECG (electrocardiogram) data. Below is a breakdown of the results for the various correlations and regressions performed:

EEG Data Analysis

The EEG data consists of 16 columns of measurements. While specific correlation results within the EEG dataset aren't provided in the summary, the focus is on how EEG data might relate to EMG data, particularly in the context of facial muscles.

EMG Data Analysis

The EMG data consists of measurements for facial and leg muscles. The correlation between the third and fourth columns (presumably representing different muscle activities) is as follows:

```python
EMG.corr(method='pearson')
```

	3	4
3	1.000000	0.034805
4	0.034805	1.000000

This Pearson correlation coefficient of approximately 0.0348 indicates a very weak linear relationship between these two muscle activities.

Regression Analysis for EMG

A linear regression was performed to predict one muscle activity based on another. The R-squared value is quite low, indicating that the model explains only a small fraction of the variance in the dependent variable.

```python
regr.fit(X_train, y_train)
regr.score(X_test, y_test)
# Output: 0.0012755907170908243
```

This low R-squared value (approximately 0.0013) suggests that the linear model does not fit the data well, meaning there is little to no linear relationship between the two muscle activities being studied.

ECG Data Analysis

The ECG data consists of heart measurements, and its relationship with facial muscle activities was examined. The Pearson correlation coefficient is as follows:

```python
ECG.corr(method='pearson')
```

	3	4
3	1.000000	-0.284513
4	-0.284513	1.000000

This negative correlation coefficient of approximately -0.2845 indicates a weak inverse relationship between the two variables.

Regression Analysis for ECG
A linear regression was performed to predict facial muscle activity based on heart activity. The R-squared value indicates a modest fit:

```python
regr.fit(X_train, y_train)
regr.score(X_test, y_test)
# Output: 0.08646439411447815
```

An R-squared value of approximately 0.0865 indicates that the model explains about 8.65% of the variance in the dependent variable, which is relatively low but suggests a slightly better fit than the EMG regression.

Combined EEG and EMG Analysis
Finally, the relationship between EEG data and EMG facial muscle activity was examined using multiple linear regression.

Multiple Linear Regression Results
The regression coefficients and summary statistics are as follows:

```python
regr.fit(X, Y)
print('Intercept: \n', regr.intercept_)
print('Coefficients: \n', regr.coef_)
```

- **Intercept: 34.4929663902061
- **Coefficients: Various small values indicating the relationship strength and direction between EEG channels and EMG facial muscle activity.

OLS Regression Results Summary:

```plaintext
```

```
                    OLS Regression Results
==============================================================================
Dep. Variable:                      3   R-squared:                       0.003
Model:                            OLS   Adj. R-squared:                  0.002
Method:                 Least Squares   F-statistic:                     3.753
Date:                Sun, 26 Jul 2020   Prob (F-statistic):           5.29e-07
Time:                        16:24:17   Log-Likelihood:            -1.1500e+05
No. Observations:               19029   AIC:                         2.300e+05
Df Residuals:                   19012   BIC:                         2.302e+05
Df Model:                          16
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         34.4930      6.850      5.035      0.000      21.066      47.920
1          -8.782e-05      0.000     -0.435      0.664      -0.000       0.000
2              0.0002      0.000      1.014      0.311      -0.000       0.001
3              0.0002   9.71e-05      1.548      0.122   -4.01e-05       0.000
4           4.109e-05   8.18e-05      0.502      0.615      -0.000       0.000
5           -9.11e-05   5.16e-05     -1.767      0.077      -0.000    9.96e-06
6           -8.28e-05   2.13e-05     -3.880      0.000      -0.000    -4.1e-05
7             -0.0011      0.002     -0.501      0.616      -0.005       0.003
8              0.0011      0.002      0.516      0.606      -0.003       0.005
9             -0.0023      0.002     -1.181      0.238      -0.006       0.002
10            -0.0037      0.003     -1.392      0.164      -0.009       0.001
11             0.0006      0.000      1.736      0.083   -7.77e-05       0.001
12            -0.0011      0.000     -2.431      0.015      -0.002      -0.000
13             0.0098      0.003      3.183      0.001       0.004       0.016
14            -0.0033      0.001     -2.792      0.005      -0.006      -0.001
15         -8.367e-05    3.1e-05     -2.702      0.007      -0.000         ...
```

R-squared: 0.003 indicates a very small fraction of the variance in EMG facial muscle activity is explained by the EEG data.
Significance: Most EEG channels do not significantly predict the EMG facial muscle activity, as indicated by their p-values.

The analyses reveal weak correlations and low explanatory power of linear models in predicting muscle activity from EEG or other muscle activity. This suggests that either the relationships are non-linear, the data contains significant noise, or other unmeasured factors are at play. Further investigation with more sophisticated models or additional data preprocessing might be necessary to uncover more meaningful relationships.

Conclusion:

In conclusion, the development of a multimodal deep learning architecture for an EEG-EMG based Brain-Computer Interface (BCI) controlled robotic hand exoskeleton for neuro-rehabilitation of stroke patients represents a significant advancement in the field of assistive technology and rehabilitation medicine. Through the integration of various techniques and methodologies, including autoencoders, Generative Adversarial Networks (GANs), attention mechanisms, Graph Neural Networks (GNNs), Layerwise Relevance Propagation (LRP), and Grad-CAM, the proposed architecture aims to address the challenges associated with interpreting, understanding, and improving the performance of BCI systems. The process begins with data collection and preparation, where EEG and EMG signals are acquired, preprocessed, and augmented to ensure high-quality input data for model training. Model architecture design follows, wherein the components of the neural network are determined, and the structure of the model is defined based on the task requirements. Autoencoder training and GAN training for synthetic sample generation are crucial steps in the development of the architecture. Autoencoders facilitate unsupervised feature learning and dimensionality reduction of EEG and EMG data, while GANs generate synthetic samples that closely resemble real data, thereby enhancing the diversity of the training dataset. Ensemble learning, attention mechanisms integration, and GNNs integration further contribute to improving the model's performance and interpretability. Ensemble learning combines predictions from multiple models to enhance accuracy and robustness, while attention mechanisms and GNNs capture spatial dependencies and facilitate effective information integration and decision-making processes. Additionally, techniques such as Layerwise Relevance Propagation (LRP) and Grad-CAM provide valuable insights into the model's decision-making process, enabling researchers and practitioners to interpret and understand model decisions better. These explainable AI techniques enhance model transparency and aid in building user trust, especially in critical applications such as neuro-rehabilitation for stroke patients. Overall, the multimodal deep learning architecture presented in this report holds great promise for advancing BCI technology and improving the quality of life for stroke patients undergoing neuro-rehabilitation. By leveraging state-of-the-art techniques and methodologies, the architecture aims to facilitate more accurate, reliable, and interpretable BCI systems, ultimately leading to better patient outcomes and enhanced rehabilitation effectiveness.

Ref:

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5758532/
https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9629969
https://www.mdpi.com/1424-8220/21/21/7014
https://www.sciencedirect.com/science/article/abs/pii/S1567423109701510
https://www.tandfonline.com/doi/full/10.1080/2326263X.2022.2029308
https://pubmed.ncbi.nlm.nih.gov/34891407/
https://www.frontiersin.org/articles/10.3389/fnbot.2019.00007/full
https://irjiet.com/common_src/article_file/1567692947_b3ed8079f1_3_irjiet.pdf
https://www.researchgate.net/publication/330114656_Hybrid_EEG-EMG_based_brain_computer_interface_BCI_system_for_real-time_robotic_arm_control
https://github.com/neuroidss/EEG-GAN?tab=readme-ov-file