

## Article

# Comparative Analysis of Deep Learning Models for Optimal EEG-Based Real-Time Servo Motor Control

Dimitris Angelakis , Errikos C. Ventouras, Spiros Kostopoulos  and Pantelis Asvestas 

Department of Biomedical Engineering, University of West Attica, 12243 Athens, Greece; ericvent@uniwa.gr (E.C.V.); skostopoulos@uniwa.gr (S.K.); pasv@uniwa.gr (P.A.)

\* Correspondence: dangelakis@uniwa.gr

**Abstract:** This study harnesses EEG signals to enable the real-time control of servo motors, utilizing the OpenBCI Community Dataset to identify and assess brainwave patterns related to motor imagery tasks. Specifically, the dataset includes EEG data from 52 subjects, capturing electrical brain activity while participants imagined executing specific motor tasks. Each participant underwent multiple trials for each motor imagery task, ensuring a diverse and comprehensive dataset for model training and evaluation. A deep neural network model comprising convolutional and bidirectional long short-term memory (LSTM) layers was developed and trained using k-fold cross-validation, achieving a notable accuracy of 98%. The model's performance was further compared against recurrent neural networks (RNNs), multilayer perceptrons (MLPs), and Transformer algorithms, demonstrating that the CNN-LSTM model provided the best performance due to its effective capture of both spatial and temporal features. The model was deployed on a Python script interfacing with an Arduino board, enabling communication with two servo motors. The Python script predicts actions from preprocessed EEG data to control the servo motors in real-time. Real-time performance metrics, including classification reports and confusion matrices, demonstrate the seamless integration of the LSTM model with the Arduino board for precise and responsive control. An Arduino program was implemented to receive commands from the Python script via serial communication and control the servo motors, enabling accurate and responsive control based on EEG predictions. Overall, this study presents a comprehensive approach that combines machine learning, real-time implementation, and hardware interfacing to enable the precise and real-time control of servo motors using EEG signals, with potential applications in the human–robot interaction and assistive technology domains.

**Keywords:** EEG signals; servo; machine learning; real-time control; deep neural network; LSTM; Python; Arduino



**Citation:** Angelakis, D.; Ventouras, E.C.; Kostopoulos, S.; Asvestas, P. Comparative Analysis of Deep Learning Models for Optimal EEG-Based Real-Time Servo Motor Control. *Eng* **2024**, *5*, 1708–1736. <https://doi.org/10.3390/eng5030090>

Academic Editors: Juvenal Rodriguez-Resendiz, Akos Odry, José Manuel Álvarez-Alvarado and Marco Antonio Aceves-Fernandez

Received: 27 June 2024  
Revised: 15 July 2024  
Accepted: 24 July 2024  
Published: 2 August 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Electroencephalography (EEG) is a field dedicated to the recording and interpretation of the electroencephalogram. The electroencephalogram (EEG) represents the electrical signal produced by the coordinated activity of brain cells, specifically the temporal pattern of extracellular field potentials resulting from their synchronized activity. The term “electroencephalogram” is derived from the Greek words “enkephalo” (brain) and “graphein” (to write). An EEG can be recorded using electrodes placed on the scalp or directly on the cortex. When recorded directly on the cortex, it is sometimes referred to as an electrocorticogram (ECoG). Electric fields measured intracortically are termed Local Field Potentials (LFPs). An EEG recorded without an external stimulus is termed a spontaneous EEG, while an EEG generated in response to an external or internal stimulus is termed an event-related potential (ERP). The amplitude of EEG in a normal awake subject recorded with scalp electrodes ranges from 10–100 mV. In cases of epilepsy, EEG amplitudes can increase nearly tenfold, with cortical amplitudes ranging from 500–1500 mV. EEG rhythms include delta (0.5–4 Hz), theta (4–8 Hz), alpha (8–13 Hz), beta (13–30 Hz), and gamma (above 30 Hz).

Gamma components are challenging to record with scalp electrodes, generally not exceeding 45 Hz, but an ECoG can register frequencies up to 100 Hz or higher. The contributions of these rhythms vary with age and the behavioral state, especially alertness. EEG patterns also differ between individuals and are influenced by neuropathological conditions, metabolic disorders, and drug effects. Delta rhythm is dominant during deep sleep, with large amplitudes (75–200 mV) and strong coherence across the scalp. Theta rhythms are rare in adult humans but common in rodents, with a broader frequency range (4–12 Hz) and high amplitude. In humans, theta activity occurs in emotional or cognitive states. Alpha rhythms are prominent during wakefulness, especially in the posterior regions when the eyes are closed and the subject is relaxed. They are attenuated by visual attention and mental effort. Mu rhythms, which are similar in frequency, are related to motor cortex function and are blocked by motor activities. Beta activity is associated with increased alertness and focused attention. Gamma activity relates to information processing and the onset of voluntary movements. Generally, slower cortical rhythms are linked to an idle brain, while faster rhythms are related to information processing.

EEGs are observed in all mammals, with primate EEG characteristics being most similar to humans. Cat, dog, and rodent EEGs also resemble human EEGs but have different spectral contents. In lower vertebrates, electric brain activity is observed, but it lacks the rhythmic behavior found in recordings from higher vertebrates [1].

The brain–computer interface (BCI) [2] utilizes EEG signals to enable direct communication between the brain and external devices, such as robotic arms or computers. The process involves the following steps: EEG signals are recorded using electrodes placed on the scalp, and the raw EEG data are then processed to extract relevant features. These features, such as the power spectral density, event-related potentials, or specific frequency bands like alpha or beta rhythms, are extracted from the EEG signals. Advanced algorithms, including machine learning or signal processing techniques, are used to decode the extracted features and translate them into control commands for the robotic arm or device. Subsequently, the decoded commands are used to control the movement or operation of the robotic arm or device.

Types of BCI systems using EEG signals include motor imagery-based BCIs, where users imagine specific motor movements, such as moving their right hand, which produce distinct EEG patterns. These patterns are then decoded to control the robotic arm. In the P300 Speller system, users focus on a specific character or item presented in a matrix, and when the desired item flashes, a P300 waveform is elicited in the EEG signal, which is used to select the item. Another type is the SSVEP (Steady-State Visually Evoked Potential) system, where users focus on visual stimuli flashing at different frequencies, and the corresponding EEG responses are used to determine the user's intended action.

Benefits of EEG-based BCIs include their non-invasive nature, as unlike invasive methods, they do not require surgical implantation, and their versatility, as they can be used by individuals with severe motor disabilities to control external devices and improve their quality of life. However, there are several challenges associated with EEG-based BCIs. These challenges include the susceptibility of EEG signals to noise and artifacts, which can affect the accuracy of the BCI; the need for users to undergo extensive training to generate consistent and distinguishable EEG patterns; and the limited bandwidth of EEG signals compared to invasive methods, which can restrict the range and complexity of tasks that can be performed.

EEG signals play a crucial role in the development of brain–computer interfaces, enabling direct communication and control of servo motors, robotic arms or devices through the user's brain activity. With advancements in signal processing techniques and machine learning algorithms, EEG-based BCIs have the potential to revolutionize the field of assistive technology and neurorehabilitation [3].

BCI technology has emerged as a transformative field bridging neuroscience and engineering, enabling direct communication between the human brain and external devices. BCI systems hold immense potential for augmenting human capabilities, particularly in

assisting individuals with disabilities and enhancing human–machine interactions. One promising application of BCI technology is the control of robotic systems using neural signals extracted from the brain [4].

#### *Related Work*

Recently, there has been a growing interest in utilizing machine learning (ML) algorithms, specifically utilizing long short-term memory (LSTM) neural network models [5], to control robotic devices using electroencephalography (EEG) signals extracted from the brain. For instance, in ref. [6], researchers introduced a classification framework utilizing LSTM neural network models for the classification of motor imagery electroencephalograph signals in BCI systems by employing a one-dimension-aggregate approximation (1d-AX) for feature extraction and incorporating a channel weighting technique inspired by the classical common spatial pattern and achieved enhanced classification performance.

In [7], the authors combined the use of discrete wavelet transform (DWT) for frequency feature extraction with a bidirectional long short-term memory (BiLSTM) neural network to improve the classification of MI-based EEG signals in a BCI system, achieving satisfactory results.

In [8], Garcia-Moreno et al. explored the feasibility of motor imagery classification using a low-cost and low-invasive BCI headband, achieving a validation accuracy of 96.5% with their CNN-LSTM deep learning model. The study emphasized the necessity of analyzing all five EEG wave types (alpha, beta, theta, delta, and gamma) through specific channels (TP9, TP10, AF7, and AF8) for accurate classification. The authors noted that raw data alone were insufficient for achieving high accuracy, highlighting the importance of feature extraction. While the Muse headband used in the study offers low invasiveness compared to similar devices, its intrusiveness in outdoor activities and the small sample size were identified as limitations. Future work aims to expand the sample size, integrate the model with existing e-health systems, extend the detection capabilities to other types of motor imagery, and explore real-time predictions to enhance user experience. The study underscores the potential of such wearable technologies to democratize BCI adoption in healthcare, enabling interactions with computer systems through thought alone and integrating seamlessly with comprehensive health monitoring systems.

In [9], Li et al. proposed a sophisticated EEG classification algorithm for motor imagery tasks that combines convolutional neural networks (CNN) and long short-term memory (LSTM) networks. This hybrid CNN-LSTM feature fusion network leverages the strengths of both architectures: the CNN is adept at capturing spatial features from the EEG signals, while the LSTM excels at extracting temporal dependencies. The parallel integration of these networks allows for a comprehensive feature extraction process, leading to a significant improvement in classification accuracy. The study demonstrated that this method outperformed traditional approaches, providing a promising avenue for enhancing brain–computer interface (BCI) systems with more accurate and reliable EEG signal interpretation.

In their 2023 study [10], Martín-Chinea et al. examined the impact of time windows on the performance of LSTM networks in EEG-based brain–computer interfaces (BCIs). They highlight the essential role of high-quality data in ensuring the efficient performance of both classical machine learning and deep learning algorithms like LSTMs. Although deep learning techniques typically require large datasets and incur higher computational costs, the sequential processing capabilities of LSTMs provide significant advantages. The study underscores that while LSTM networks have been extensively applied in the EEG-based BCI literature, there is a notable gap in research focusing on the time window parameter, which is critical for the success of such models.

The research demonstrates that LSTMs can effectively capture temporal patterns in EEG signals, particularly in distinguishing between eyes open (EO) and eyes closed (EC) states. This task is challenging due to the overlapping characteristics of alpha-band power in both states. Using LSTM networks, the researchers show that it is possible

to elegantly resolve these challenges without relying on the assumptions necessary for threshold-based methods.

Their findings emphasize that the choice of the time window significantly affects the accuracy of LSTM classifiers, with better performance compared to classical algorithms that require extensive preprocessing. Unlike RNNs, LSTMs can retain information over long periods, which enhances the classification results. The study concludes that, despite the efficacy of the proposed LSTM-based approach, other modalities like EMG or video eye tracking might yield better results for certain tasks. Nonetheless, the versatility of the LSTM approach makes it applicable to various domains, including motor disease feedback, neuromarketing, and complex brain task classification.

In the authors' previous study [11], Angelakis et al. created an efficient model divided into two parts. In Part I, the constant features approach was employed, which encompassed data loading, feature extraction, preprocessing, model selection, and tuning to identify the best-performing model. The performance of classification algorithms (support vector machine (SVM), decision tree classifier, and random forest classifier) was evaluated using root-mean-squared error metrics. In Part II, a multivariate time series approach was utilized to enhance the accuracy and robustness of the model. A neural network architecture consisting of convolutional filters followed by a LSTM was employed for EEG classification. The convolutional layer's purpose was to extract high-level features from the EEG data. This approach was particularly well-suited for handling the sequential nature of EEG signals. The results obtained were impressive, with the model achieving an accuracy of 98% in predicting the chosen action based on EEG signals.

This study introduces several advancements to the model. First, the inclusion of k-fold cross-validation [12] aims to enhance the model's generalization and robustness. This technique partitions the dataset into 'k' subsets and iteratively trains the model on 'k - 1' of these subsets while validating on the remaining subset. Second, the integration of Dropout Regularization was incorporated to prevent overfitting and boost the model's performance. This technique randomly sets a fraction of input units to zero during training, helping to reduce overfitting. Third, the model's training was optimized using the Adam optimizer, an adaptive learning rate optimization algorithm well-suited for deep learning. Lastly, to minimize any dataset's imbalance, class weights were computed and employed to give more weight to under-represented classes, thereby enhancing the model's generalization. To evaluate the model's performance, a classification report and confusion matrix were computed for each fold during the cross-validation process, with the average validation loss and accuracy printed at the training process's conclusion.

Building upon these enhancements, this study extends the comparison by incorporating recurrent neural networks (RNNs), multilayer perceptrons (MLPs), and Transformer algorithms alongside the CNN-LSTM deep neural network. In addition to the aforementioned advancements, the primary goal was to implement the model in hardware to test its practical application. The trained LSTM model was integrated into a hardware setup to control two servo motors based on predictions. Initially, preprocessed EEG data were loaded from a validation data folder, and the trained LSTM model was loaded using the TensorFlow Keras API. A serial connection was then established between the computer and an Arduino board to facilitate the control of the servo motors. The preloaded validation data were utilized to predict the movement labels using the LSTM model. A control function was developed to send commands to the Arduino based on the following predicted labels: 'L' for left arm, 'R' for right arm, and 'N' for no action. The model's predictions were continuously tested in real-time using the preloaded data, with the confusion matrix and classification report printed for each iteration to evaluate the model's performance. Based on the LSTM's predictions, the Arduino was then instructed to move the servo motors accordingly.

The subsequent sections of this paper will elucidate the methodology employed, detailing the process of EEG data acquisition, model development, real-time implementation, and performance evaluation. By elucidating the intricacies of the proposed approach, this

study seeks to contribute to the burgeoning field of BCI research and foster advancements in human–machine collaboration.

## 2. Materials and Methods

### 2.1. Data Collection

The data collection process involved acquiring EEG recordings from individuals performing motor imagery tasks associated with left hand movement, right hand movement, and no action scenarios. EEG data were obtained from the publicly available OpenBCI Community Dataset, specifically targeting recordings collected in February 2021. This dataset encompasses EEG data from 52 subjects, including 38 validated subjects with discriminative features (33 males, 19 females, mean age  $\pm$  SD =  $24.8 \pm 3.86$  years), and the experiment was approved by the Institutional Review Board of Gwangju Institute of Science and Technology. Each subject participated in the same experiment, and subject IDs were denoted and indexed as s1, s2, . . . , s52. Subjects s20 and s33 were both-handed, and the other 50 subjects were right-handed [13]. When exploring EEG datasets for research and development in the realm of BCI systems and motor imagery classification, several datasets are available for consideration in addition to the OpenBCI Community Dataset. Notable alternatives include the PhysioNet EEG Motor Movement/Imagery Dataset [14] and the EEG Motor Movement/Imagery Datasets available on Kaggle [15].

In OpenBCI, community participants imagined executing specific motor tasks while wearing EEG headsets to capture electrical signals from their brain activity. Each participant underwent multiple trials for each motor imagery task, creating a diverse and comprehensive dataset for analysis and model training.

#### 2.1.1. Non-Task-Related States

The dataset includes six types of non-task-related data: eye blinking, eyeball movement up/down, eyeball movement left/right, head movement, jaw clenching, and the resting state. Each type of noise was recorded twice for 5 s, except the resting state, which was recorded for 60 s.

#### 2.1.2. Real Hand Movement

Subjects sat in a chair with armrests and watched a monitor. Each trial began with a black screen displaying a fixation cross for 2 s, followed by a random instruction (“left hand” or “right hand”) on the screen for 3 s. Subjects moved the appropriate hand based on the instruction. After the hand movement, a blank screen appeared, providing a break for 4.1 to 4.8 s. This process was repeated 20 times for one run, with one run performed.

#### 2.1.3. Motor Imagery (MI) Experiment

The MI experiment followed the same setup as the real hand movement experiment. Subjects imagined the hand movement according to the given instruction. Five or six runs were performed, with each run consisting of the same sequence as the real hand movement trials. After each run, classification accuracy was calculated and feedback was provided to motivate the subject. A maximum 4 min break was given between runs, depending on the subject’s needs.

The EEG recordings were obtained using 64 Ag/AgCl active electrodes configured in a 64-channel montage based on the international 10–10 system, ensuring comprehensive coverage of the scalp to capture neural activity from relevant brain regions. The EEG signals were recorded at a high sampling rate of 512 Hz, which provides the detailed temporal resolution necessary for analyzing fast neural dynamics.

Data acquisition was performed using the Biosemi ActiveTwo system [16], known for its high-quality signal capture and low noise levels. The BCI2000 system 3.0.2 was utilized not only to collect the EEG data but also to present motor imagery instructions to participants. This integration ensured synchronized data collection and task presentation.

Built-in preprocessing features included the simultaneous recording of EMG signals with the same system and sampling rate to monitor actual hand movements. Two EMG electrodes were attached to the flexor digitorum profundus and extensor digitorum on each arm, allowing for the differentiation between imagined and real hand movements.

The EEG datasets from the OpenBCI Community Dataset were meticulously selected over these other similar datasets based on several pivotal criteria to ensure the quality, reliability, and relevance of the data for motor imagery (MI)-based brain–computer interface (BCI) research. Firstly, participants were chosen if they successfully completed all motor imagery tasks without significant artifacts or disruptions. This criterion aimed to guarantee the integrity of the EEG recordings, aligning with the general belief among BCI investigators that a BCI can be achieved through induced neuronal activity from the cortex, rather than evoked neuronal activity.

Secondly, a strong emphasis was placed on datasets that included comprehensive metadata, encompassing a psychological and physiological questionnaire, EEG coordinates, and EEGs for non-task related states. This holistic metadataset provided valuable insights into the participants' cognitive and physiological states during the motor imagery tasks, facilitating a deeper understanding of the variability in BCI performance across different sessions and subjects.

Additionally, EEG recordings with a low percentage of bad trials were prioritized to ensure data reliability and consistency. This validation criterion was corroborated by the study's approach to validating the EEG datasets using the percentage of bad trials, event-related desynchronization/synchronization (ERD/ERS) analysis, and classification analysis. Notably, the selected EEG datasets exhibited clear contralateral ERD and ipsilateral ERS patterns in the somatosensory area, which are well-recognized patterns of MI. This consistency in EEG patterns reinforced the robustness and reliability of the chosen datasets.

Furthermore, the study's findings highlighted that 73.08% of the datasets (38 subjects) included reasonably discriminative information. The inclusion of both well-discriminated and less-discriminated datasets in the EEG datasets provided researchers with opportunities to investigate human factors related to MI BCI performance variation and to potentially achieve subject-to-subject transfer by utilizing the comprehensive metadata.

In summary, the criteria for selecting the EEG datasets from the OpenBCI Community Dataset were designed to align with the study's objectives and methodologies, ensuring that the chosen datasets were of high quality, reliable, and suitable for in-depth analysis and model training in the context of MI-based BCI research

## 2.2. The Hardware Setup

The hardware setup, a cornerstone of the experimental framework, meticulously entailed connecting the Arduino UNO R4 Minima (manufacturer: Arduino, based in Turin, Italy) [Figure 1] to two Servo (SG90 Micro Servo (manufacturer: Tower Pro, based in Shenzhen, Guangdong, China) [Figure 2]) using digital pins for control and power. Before experimentation, meticulous calibration and synchronization of hardware components were conducted to ensure optimal performance and data integrity. The personal computer used for the research was an HP ProBook 450 G9 Notebook PC (manufacturer: HP Inc., based in Palo Alto, CA, USA). It featured an Intel Core i5-1235U processor (manufacturer: Intel Corporation, based in Santa Clara, CA, USA) from the 12th generation, which includes 10 cores and 12 logical processors and operates at a base clock speed of approximately 1.3 GHz. This processor is designed to handle multitasking efficiently and is well-suited for moderate computational workloads. The system was equipped with 16 GB of RAM Double Data Rate 4, which is adequate for running multiple applications simultaneously and performing memory-intensive tasks. This amount of memory ensures that the laptop can handle complex data processing and software applications required for research purposes.

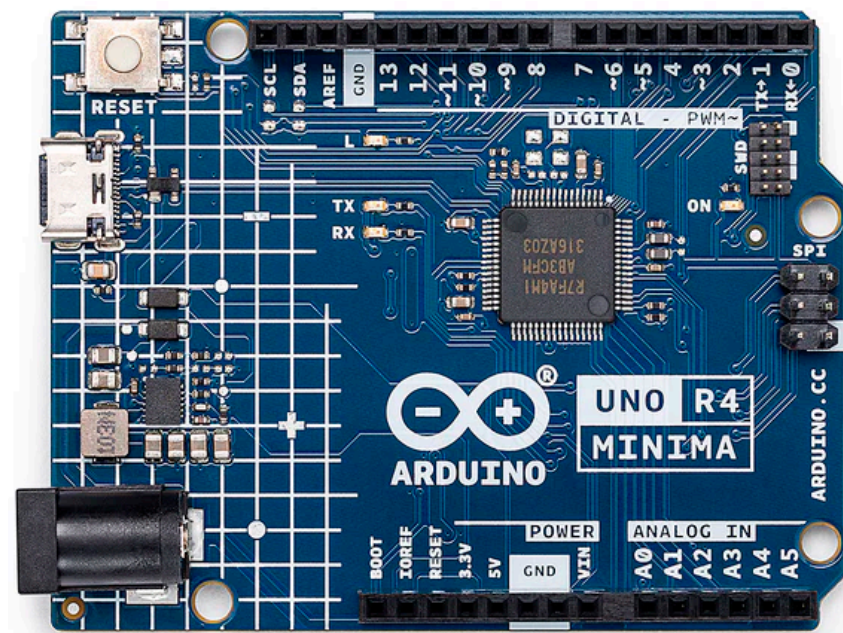


Figure 1. Arduino® UNO R4 Minima.



Figure 2. Waveshare SG90 Micro Servo.

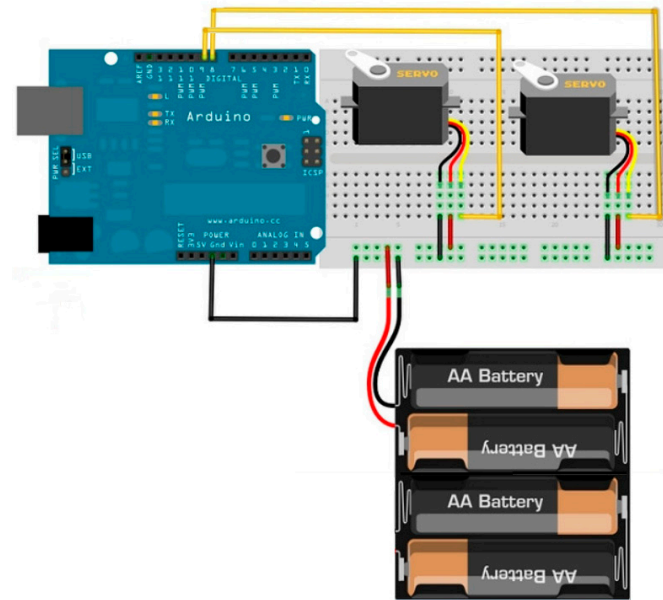
In the experimental setup, the Arduino UNO R4 Minima microcontroller was central to the integration with the servo motors.

The Arduino UNO R4 Minima was selected due to its hardware compatibility, enhanced performance, and extensive community support, which collectively facilitated the seamless integration and control of servo motors in our setup. The microcontroller's form factor, pinout, and 5 V operating voltage are consistent with its predecessor, the UNO R3, ensuring compatibility with existing shields and leveraging the extensive ecosystem already established for the Arduino UNO. This compatibility was essential for integrating additional hardware components without significant modifications. The UNO R4 Minima boasts increased memory and a faster clock speed, which are crucial for handling the complex computations and real-time data processing required in our motor imagery classification system. The enhanced processing capabilities enable precise calculations, ensuring the accurate and responsive control of the servo motors based on EEG signal interpretation.

The Arduino was connected to two servo motors, which were powered by an external 4 AA battery supply. The entire assembly, including the Arduino, servo motors, and the battery, was neatly integrated onto a breadboard for a compact and organized configuration. For the electrical connections, pins 9 and 8 on the Arduino were utilized. These pins were specifically chosen to interface with the control inputs of the servo motors, allowing the Arduino to send precise control signals to the motors [Figure 3].

Integrating with Arduino was crucial to the study's methodology. This integration allowed for seamless communication and control of the servo motors. Within this setup,

Arduino facilitated the precise execution of desired actions by the servo motors, operating based on commands relayed from the Python script. This real-time implementation effectively showcased the system's capability to accurately interpret EEG signals and translate them into actionable commands with minimal latency. As a result, it ensured the smooth and responsive movement of the servo motors, enhancing the overall efficacy and performance of the BCI system.



**Figure 3.** The setup of the Arduino UNO R4 Minima microcontroller interfaced with two servo motors, powered by an external 4 AA battery supply, all integrated onto a breadboard.

The process encompassed several key steps, starting from data preprocessing and feature extraction to model development and real-time implementation.

### 2.3. Data Preprocessing

In the preprocessing of the EEG data phase, a meticulous approach was adopted to enhance the quality and reliability by addressing common challenges such as unwanted noise, artifacts, and baseline drift commonly encountered in EEG recordings. The next step involved the application of a fourth-order Butterworth bandpass filter with a specific frequency range set between 1 Hz and 50 Hz. This bandpass filtering was pivotal in isolating the desired frequency components of the EEG signals while effectively filtering out unwanted low and high-frequency components that could distort the genuine brainwave signals. A lower cut-off frequency of 1 Hz was chosen to capture the very low-frequency components, such as delta waves, providing valuable insights into the brainwave patterns during motor imagery tasks. Conversely, an upper cut-off frequency of 50 Hz was selected to remove high-frequency noise and muscle artifacts, enhancing the overall quality and reliability of the EEG signals.

The choice of the Butterworth filter was driven by several key advantages that make it particularly suitable for this specific application. The Butterworth filter is characterized by its maximally flat frequency response within the passband, ensuring minimal signal distortion. This property is crucial for maintaining the integrity of the EEG signals and accurately capturing the brainwave patterns associated with motor imagery tasks. Additionally, the Butterworth filter provides a smooth and gradual transition from the passband to the stopband, which helps in minimizing signal distortion and effectively attenuating unwanted noise. The design of the Butterworth filter offers stability and robustness, which is essential when dealing with the high dimensionality and inherent noise present in EEG data, ensuring consistent filtering performance across different EEG recordings and subjects.



Furthermore, the Butterworth filter is relatively straightforward to design and implement using standard signal processing software libraries. This facilitates reproducibility and comparability in research, allowing other researchers to replicate the filtering process with ease. By selecting an appropriate order (fourth-order) and frequency range (1 Hz to 50 Hz), the Butterworth filter effectively reduces low-frequency drifts and high-frequency muscle artifacts. This targeted noise reduction is critical for isolating the relevant EEG components corresponding to motor imagery, thereby improving the accuracy of the subsequent analysis and classification. These advantages make the Butterworth filter particularly well-suited for EEG signal processing in the context of motor imagery tasks, ensuring that the key frequency components of interest are preserved while unwanted noise is minimized.

Following the bandpass filtering, fast Fourier transform (FFT) was used to derive frequency-domain details from EEG signals. This method enabled the conversion of EEG time-domain data into a frequency-domain format, allowing the recognition of specific spectral elements. FFT is a mathematical approach employed in this study to transform the time-domain signal into the frequency domain, providing insights into the number of active signals at each frequency. The FFT algorithm dissected the time-domain signal into its constituent frequencies, displaying the amplitude (or power) of each frequency. This aided in pinpointing the dominant frequencies in the signal and tracking their variations over time.

FFT is widely used to analyze brainwave signal frequencies. Brainwaves, which are electrical impulses produced by the brain, can be intricate and encompass signals at various frequencies. By applying FFT to the brainwave signal in this study, it was feasible to decompose it into its individual frequencies and evaluate the power at each frequency. In BCIs, FFT can be employed to extract features from EEG signals associated with specific mental states or actions, like variations in the power of the alpha frequency range (8–12 Hz) linked to relaxation or attention. The FFT technique was used to pinpoint and separate noise and artifact frequencies from the EEG data. By recognizing irregular frequency components that differed from the standard neural oscillations, these undesirable signals were effectively reduced. This procedure enhanced the signal quality and reduced potential disturbances from external factors, thereby improving the accuracy of subsequent analyses. In addition to the advanced filtering techniques, a specialized artifact removal algorithm was applied to identify and discard any irregularities or anomalies in the EEG data. These artifacts can arise from various sources such as muscle movements, eye blinks, or external interferences, and they can significantly distort the true EEG signals. By effectively removing these artifacts, the quality and accuracy of the EEG data were further improved. The technique that was used is the discrete wavelet transform (DWT), which provides a powerful method for analyzing and processing non-stationary signals like EEG data by offering both time and frequency localization.

The DWT involves decomposing a signal into a set of basic functions called wavelets. This decomposition provides both time and frequency localization, making it ideal for analyzing non-stationary signals like EEGs.

Compared to other methods like Independent Component Analysis (ICA), which separates EEG signals into independent components for artifact removal but requires manual identification and assumes statistical independence that might not always hold, or Principal Component Analysis (PCA), which reduces dimensionality and removes artifacts through orthogonal components but also requires manual artifact identification and may not effectively separate overlapping spectral properties, DWT offers superior time and frequency localization, allowing the precise identification and removal of artifacts without losing important temporal information. Its multiresolution analysis distinguishes between artifacts and neural activity effectively. DWT was chosen for its ability to handle non-stationary EEG signals, providing a detailed time–frequency representation and automated artifact removal, and preserving relevant neural signals for an accurate analysis.

The combination of bandpass filtering and FFT in the preprocessing steps played a significant role in ensuring that the EEG data used for subsequent analysis and model

training were of the highest quality, free from unwanted distortions and interferences. This comprehensive preprocessing approach significantly enhanced the reliability, validity, and suitability of the EEG data for further research and analysis.

The selection of cut-off frequencies for the Butterworth bandpass filter was informed by a thorough examination of the spectral characteristics of EEG signals associated with motor imagery tasks. The lower cut-off frequency was set at 1 Hz to capture very low-frequency components, including delta waves, which are crucial for understanding fundamental brainwave patterns related to motor planning and execution. The upper cut-off frequency was established at 50 Hz to effectively attenuate high-frequency noise and muscle artifacts, thereby enhancing the fidelity of the extracted brainwave signals.

These cut-off frequencies were determined based on established practices in EEG signal processing and prior empirical evidence supporting their efficacy in isolating relevant neural activity while mitigating the influence of confounding factors. The choice of a bandpass filter over other filter types was driven by its capacity to selectively target specific frequency components, aligning with the study's objective to focus on neural oscillations relevant to motor imagery tasks.

#### 2.4. Feature Extraction

Following the preprocessing phase, the next critical step was feature extraction, where relevant information indicative of motor imagery tasks was captured from the preprocessed EEG signals. To provide a comprehensive understanding of the brain activity during motor imagery, both time-domain and frequency-domain features were computed.

In the time domain, the power spectral density (PSD) ratio was examined to delve into the distribution of signal power across different frequency bands. By analyzing the power distribution, it became possible to discern the dominance of specific frequency components, shedding light on the prevailing neural oscillatory patterns associated with different cognitive processes. The PSD ratio was computed by dividing the continuous EEG recordings into 2 s epochs. For each epoch, the fast Fourier transform (FFT) converted the time-domain signals into the frequency domain, providing the power spectrum. The PSD was calculated by squaring the FFT coefficients' magnitudes and normalizing by the number of data points, resulting in the power at each frequency. The power was integrated over specific frequency bands (delta: 1–4 Hz, theta: 4–8 Hz, alpha: 8–13 Hz, beta: 13–30 Hz, and gamma: 30–50 Hz), and the PSD ratio was obtained by dividing the power in a specific band by the total power across all bands, such as the alpha/beta ratio. The PSD ratio is particularly relevant to motor imagery tasks because different frequency bands are linked to various cognitive and motor functions. The alpha band (8–13 Hz) is associated with relaxation, while the beta band (13–30 Hz) is linked to active thinking and motor control. The theta band (4–8 Hz) and gamma band (30–50 Hz) are involved in cognitive processing.

Hjorth parameters provided insights into the EEG signal's time-domain characteristics, including its complexity, mobility, and activity. Activity measures the signal power and indicates overall brain activity, with higher activity values during motor imagery suggesting increased neural engagement. Mobility reflects the mean frequency, indicating the proportion of the standard deviation of the power spectrum, where changes in mobility can show shifts in the dominant frequency components, correlating with the transition from a resting state to active motor imagery. Complexity indicates the similarity of the signal shape to a pure sine wave, reflecting the signal's complexity, with increased complexity during motor imagery tasks signifying more intricate neural processing. By quantifying these aspects, the Hjorth parameters offered a valuable glimpse into the dynamics of cortical excitability and the overall energy distribution within the brain signals. The Petrosian fractal dimension served as a measure of signal irregularity, offering a quantification of the EEG waveform's complexity at various scales. This feature aided in detecting subtle variations and intricate patterns that might signify specific cognitive states or signal abnormalities. The Petrosian fractal dimension is a valuable measure of signal complexity, quantifying the irregularity and intricacy of EEG waveforms. This measure has been successfully used in previous

studies. For example, a study by Mohamed and Jusas (2024) [17] demonstrated that fractal dimensions, including the Petrosian fractal dimension, can effectively classify mental states from EEG signals, validating their use in distinguishing complex neural activity in motor imagery and emotion recognition. Another study by Moctezuma and Molinas (2020) [18] highlighted the efficacy of fractal dimensions in detecting epileptic seizures from EEG signals, showing high accuracy in identifying seizure states using fractal features. The Frobenius norm, representing signal variance and amplitude, holds significance in assessing the overall intensity of EEG signals. During motor imagery tasks, different cognitive states can produce varying levels of neural activity, which are reflected in the amplitude and variance of EEG signals. The Frobenius norm helps differentiate these conditions by quantifying the overall signal energy. For instance, higher Frobenius norm values might indicate increased neural activation associated with active motor imagery, while lower values could correspond to a resting or less active cognitive state. In this study, the Frobenius norm was used as one of the features for classifying motor imagery tasks. By analyzing the Frobenius norm values across different epochs, the study could identify patterns of signal intensities that correlated with specific motor imagery activities, such as imagining the movement of the right hand or left hand. By considering the Frobenius norm, the study tapped into amplitude-based characteristics, which could potentially differentiate diverse cognitive conditions.

In the frequency domain, spectral power, coherence, and additional features derived from the fast Fourier transform (FFT) were computed to capture the frequency characteristics of the EEG signals. Spectral power provides information about the intensity or magnitude of the EEG signals at different frequency bands, offering insights into the dominant brainwave frequencies associated with different motor imagery tasks. Spectral power was computed following several preprocessing steps. The raw EEG signals were first subjected to a fourth-order Butterworth bandpass filter with a frequency range of 1 Hz to 50 Hz, which isolated relevant frequency components and eliminated unwanted low-frequency drifts and high-frequency noise. DWT then used to remove artifacts, ensuring that the remaining data reflected pure neural activity related to motor imagery tasks. The continuous EEG recordings were divided into smaller, manageable epochs, allowing for a detailed analysis over time.

After preprocessing, the spectral power was computed for each epoch. The fast Fourier transform (FFT) was applied to each epoch to convert the time-domain EEG signals into the frequency domain, providing the power spectrum of the signal, which showed how the signal's power was distributed across different frequencies. Wavelet transform coefficients were analyzed to capture transient features of the EEG signals, which refer to brief, non-stationary events in the signal, such as bursts of oscillatory activity or sudden changes in amplitude. The wavelet transform coefficients were computed using the Daubechies wavelet family (specifically, db4), known for its effectiveness in capturing transient features in non-stationary signals like EEGs. The db4 wavelet was chosen for its ability to provide a good balance between time and frequency localization. These transient features are crucial for identifying specific patterns of brain activity that occur during motor imagery tasks, as they can reflect the initiation and execution phases of motor planning.

Additionally, frequency band ratios were calculated to assess the relative activity of different brainwave patterns. These ratios were calculated by dividing the spectral power in one frequency band by the power in another (e.g., alpha: 8–13 Hz, beta: 13–30 Hz, and theta: 4–8 Hz).

Frequency band ratios are significant in motor imagery tasks because they reveal the balance of cognitive and motor functions. For instance, the alpha/beta ratio distinguishes between relaxation and active cognitive processing, where a decrease may indicate a shift from a relaxed state to active motor planning. The theta/beta ratio provides insights into attentional and cognitive control processes, with higher ratios often indicating increased cognitive load and engagement.

By analyzing these ratios, researchers can identify patterns of brain activity correlating with motor imagery tasks. For example, an increased theta/beta ratio might indicate higher cognitive load during complex motor imagery, while a lower alpha/beta ratio suggests active motor planning and execution. These insights enhance the accuracy of classification models in brain–computer interface (BCI) systems, providing a deeper understanding of motor imagery processes. These frequency-domain features can be correlated with the speed, direction, and precision of servo motor movements, highlighting the brainwave frequencies and functional connectivity patterns that are most indicative of successful motor imagery tasks. A study by Barone and Rossiter (2021) [19] found that higher beta activity is related to quicker motor responses, suggesting that beta power influences motor speed. This is particularly relevant for applications requiring rapid and precise motor control. Additionally, studies have shown that theta and alpha oscillations are involved in cognitive and motor processes that influence directional decisions. For instance, theta activity is linked to attention and working memory, which are critical for planning directional movements (Klimesch, 1999) [20]. Similarly, alpha oscillations, particularly in frontal regions, are associated with motor imagery and preparation, affecting directional control. Research by Hidalgo et al. (2017) [21] on CNC servomotor tuning demonstrated that precise control parameters influenced by frequency-domain features could significantly enhance contouring accuracy and precision in motor tasks. This study highlights the role of gamma power in achieving precise movements in applications requiring high accuracy.

The rationale behind choosing these specific features lies in their ability to capture both the temporal and spectral characteristics of EEG signals, which are critical for accurate motor imagery classification. Time-domain features provide insights into the overall signal power, complexity, and temporal variations, while frequency-domain features reveal the underlying oscillatory patterns and interactions between different frequency bands. By combining both types of features, the analysis can leverage the strengths of each approach, leading to a more robust and comprehensive understanding of the neural mechanisms underlying motor imagery tasks. This dual approach ensures that both spatial and temporal dynamics of the EEG signals are captured, enhancing the performance of machine learning models in classifying motor imagery. The segmentation of EEG signals involved dividing the continuous EEG recordings into smaller, manageable segments or epochs, which was crucial for analyzing the temporal dynamics of the EEG data associated with motor imagery tasks. Each EEG recording session was divided into epochs of a fixed duration, specifically 2 s for this study. Each epoch corresponds to a specific motor imagery task (left hand movement, right hand movement, or no action) and includes all EEG channels recorded during that period.

Table 1 provides an overview of the number and nature of features extracted from the EEG signals for each epoch and channel.

**Table 1.** Summary of features extracted from EEG signals.

Feature Type	Feature	Number of Features per Channel	Total Number of Features for 64 Channels
<b>Time-Domain Features</b>			
	Power Spectral Density (PSD) Ratio	5	320
	Hjorth Parameters	3	192
	Petrosian Fractal Dimension	1	64
<b>Frequency-Domain Features</b>			
	Spectral Power	5	320
	Wavelet Transform Coefficients	7	448
	Frequency Band Ratios	2	128

Table 2 provides a concise summary of the total number of features extracted for each epoch by combining time-domain and frequency-domain features.

**Table 2.** Summary of total number of features per epoch.

Feature Type	Total Features
Time-Domain Features	$320 + 192 + 64 = 576$
Frequency-Domain Features	$320 + 448 + 128 = 896$
Grand Total	$576 + 896 = 1472$

### 2.5. Model Development

An arsenal of sophisticated software tools, including PySerial, Numpy, Sklearn, TensorFlow, Keras, MNE-Python, PyWavelets, SciPy, and Arduino IDE, served as indispensable companions throughout the model development, training, and real-time implementation phases. PySerial was used to establish and manage serial communication between the Python script and the Arduino board, enabling real-time control of the servo motors. Numpy was employed for numerical computations and data manipulation, providing efficient handling of large EEG datasets and facilitating various mathematical operations required for preprocessing and analysis. The feature extraction process was carried out using several libraries: MNE-Python was used for EEG data preprocessing, artifact removal, and segmentation; SciPy facilitated signal processing, particularly for implementing the FFT to extract spectral power features; PyWavelets was utilized for extracting wavelet transform coefficients; and Scikit-Learn (Sklearn) was used for feature extraction and model training, offering a range of machine learning algorithms and tools to process and classify the EEG data effectively. TensorFlow and Keras were pivotal in building and training the deep learning models, allowing the creation of sophisticated neural network architectures, including convolutional and LSTM layers, to accurately capture and interpret the complex patterns in the EEG signals. These libraries collectively enabled a comprehensive and efficient approach for processing and analyzing the EEG data, ultimately supporting the development of robust and accurate deep learning models for real-time implementation. Additionally, custom-tailored scripts, meticulously crafted in both Python and Arduino programming languages, played a pivotal role in expediting data processing, facilitating model deployment, and seamlessly interfacing with hardware components. Python version 3.9.13 was used.

In the pursuit of the accurate classification of EEG signals, a sophisticated deep neural network architecture was meticulously crafted. This architecture comprised both convolutional layers and bidirectional LSTM layers, strategically designed to extract intricate patterns inherent in EEG data. The convolutional layer played a pivotal role in feature extraction by convolving input feature maps with a meticulously crafted filter matrix, thereby abstracting high-level features essential for classification. Meanwhile, the bidirectional LSTM layers were enlisted to grapple with the temporal dependencies embedded within the EEG signals while effectively mitigating the notorious vanishing gradient problem often encountered in deep learning architectures. The choice of employing convolutional layers and bidirectional LSTM layers in this neural network model was driven by the need to effectively capture and interpret the intricate patterns and temporal dependencies inherent in EEG signals. EEG signals frequently exhibit spatial dependencies and local patterns essential for precise classification. Convolutional layers excel at capturing these spatial hierarchies. By convolving input feature maps with a meticulously designed filter matrix, these layers extract high-level spatial features from the EEG data, making them particularly apt for EEG signal feature extraction, which can be conceptualized as 2D signals (electrodes  $\times$  time).

EEG signals also manifest strong temporal dependencies, where the current state is influenced by past states. Bidirectional LSTMs are tailored to capture these long-range dependencies by processing the data in both forward and backward directions. This design

is especially effective at handling sequential data like EEG signals and at mitigating the vanishing gradient problem commonly faced in deep learning architectures. The bidirectional nature of LSTMs ensures that the model can leverage both past and future information when making predictions, enabling a comprehensive understanding of the context and temporal dynamics of EEG signals. Given the complex nature of EEG signals, which contain both spatial and temporal patterns, the combination of convolutional layers and bidirectional LSTM layers enables the model to effectively capture both the spatial hierarchies and temporal dependencies present in the EEG data. This renders the architecture well-suited for interpreting EEG signals for robotic control. It can accurately classify the user's intentions based on the EEG patterns and translate them into precise robotic movements.

The culmination of model development and training efforts culminated in a real-time deployment scenario, orchestrated through a meticulously crafted Python script interfacing seamlessly with an Arduino board. This script perpetually churned through preprocessed EEG data, leveraging the trained model to predict imminent actions and thereby orchestrating precise control of servo motors. The establishment of serial communication between the Python script and the Arduino board enabled the seamless transmission of commands dictating the servo motor movements. Additionally, a bespoke Arduino program was meticulously engineered to adeptly receive commands from the Python script, orchestrating swift and precise responses from the servo motors.

Alongside the primary deep neural network architecture that combines convolutional layers and bidirectional LSTM layers, this study also rigorously tested and compared other deep learning methods, including recurrent neural networks (RNNs), multilayer perceptrons (MLPs), and Transformer models, to classify motor imagery EEG signals. Each model was meticulously implemented and evaluated to understand its strengths and limitations in handling the complexity of EEG data. These algorithms were selected due to their suitability for handling BCI data, given their strengths in processing sequential data, capturing complex patterns, and leveraging advanced attention mechanisms. This diverse set of models ensures a thorough evaluation, providing a comprehensive assessment of different deep learning approaches for EEG signal classification and enhancing the understanding of their respective capabilities and limitations in the context of brain-computer interface systems.

### 2.5.1. Convolutional Neural Network (CNN)–Long Short-Term Memory (LSTM) Approach

For the development of a robust classification model capable of accurately distinguishing between different motor imagery tasks, a specialized deep learning architecture was devised. The model was constructed with a combination of convolutional and bidirectional LSTM layers, chosen specifically for their capabilities to capture both spatial and temporal dependencies within the EEG signals.

The bidirectional LSTM layers were incorporated to capture the temporal dynamics and dependencies in the EEG data, allowing the model to analyze and learn from both past and future time steps. The bidirectional nature of the LSTM layers provides a holistic view of the sequential data, enabling the model to understand the context and temporal patterns in the EEG signals associated with different motor imagery tasks.

The combination of Conv1D and bidirectional LSTM layers in the model architecture provides a comprehensive framework for feature representation and classification of the EEG signals. This architecture was chosen for its capacity to effectively process and interpret the complex spatial and temporal characteristics of the EEG data, thereby enabling the robust and accurate classification of motor imagery tasks.

In summary, the devised deep learning model architecture, comprising convolutional and bidirectional LSTM layers, was meticulously designed to capture the intricate spatial and temporal dependencies within the EEG signals, enabling the robust and accurate classification of motor imagery tasks.

In terms of model development, the selection criteria for the architecture components were based on their ability to effectively capture the spatial and temporal intricacies of the

EEG signals and their relevance to motor imagery tasks. The integration of both Conv1D and bidirectional LSTM layers in the model architecture was pivotal. It allowed the model to comprehensively capture the spatial and temporal intricacies of the EEG signals, thereby presenting an effective and representative feature set for the classification process. The choice of Conv1D layers with increasing filter sizes was made to enable the extraction of hierarchical and complex features from the EEG signals, enhancing the model's ability to discriminate between different motor imagery tasks. The bidirectional LSTM layers were incorporated to capture the temporal dependencies and long-range dependencies in the EEG data, thereby improving the model's ability to recognize and classify the sequential patterns and dynamics associated with different motor imagery tasks. The model comprises several layers, each with specific configurations. The first layer is a Conv1D layer with 32 filters, a kernel size of 3, and a ReLU activation function. This layer is designed to capture the initial spatial features of the EEG signals. The second layer is another Conv1D layer, but with 64 filters and a kernel size of 2, again using the ReLU activation function, which aims to capture more complex spatial features. This is followed by a MaxPooling1D layer with a pool size of 2 to reduce the spatial dimensions and highlight the most salient features. The third convolutional layer is a Conv1D layer with 128 filters and a kernel size of 2, utilizing the ReLU activation function, to further extract intricate spatial features.

Following the convolutional layers, the model incorporates a bidirectional LSTM layer with 64 units, configured to return sequences. This layer captures the temporal dependencies from both past and future time steps within the EEG data. After the bidirectional LSTM layer, a Flatten layer is used to convert the 3D output into a 2D format suitable for the dense layers. The final layers include a dense layer with 128 units and a ReLU activation function, which helps in learning complex representations, and the output layer, a dense layer with 3 units and a softmax activation function, which provides the classification output for the three motor imagery tasks.

### 2.5.2. Recurrent Neural Network (RNN) Approach

This study implemented and evaluated a recurrent neural network (RNN) [22] for classifying motor imagery EEG signals. The RNN architecture is well-suited for sequential data due to its ability to maintain a memory of previous inputs through its recurrent connections, making it particularly effective for a time-series analysis, such as EEG signals.

The RNN model consisted of two layers of SimpleRNN units with 64 neurons each. The first SimpleRNN layer was designed to return sequences, allowing the subsequent layers to process the temporal structure of the data. SimpleRNNs, due to their recurrent connections, can retain information about the sequence of data, making them ideal for capturing the temporal dependencies in EEG signals. This ability to remember and utilize past inputs provides a robust framework for understanding and predicting the patterns in the EEG data.

Following the SimpleRNN layers, the model included two dense layers. The first dense layer contained 128 neurons with ReLU activation, which helped in learning complex, non-linear relationships within the data. The second dense layer had 3 neurons with softmax activation, and was designed to classify the data into three categories: 'left', 'right', and 'none'.

The RNN's architecture, with its ability to process and learn from sequential data, proved effective at handling the temporal dynamics of EEG signals. By retaining information from previous time steps, the RNN was able to make more informed predictions about the current state, capturing the underlying temporal patterns essential for accurate classification. The implementation of the RNN model demonstrated its capability in effectively classifying motor imagery EEG signals. Its architecture is well-suited for tasks involving sequential data, providing a solid foundation for EEG signal analysis and classification.

### 2.5.3. Multilayer Perceptron (MLP) Approach

The study also implemented a multilayer perceptron (MLP) model [23] to classify motor imagery EEG signals. The MLP model architecture included an initial Flatten layer to reshape the input EEG data into a 1D array suitable for dense layers. This step is crucial for converting the 2D EEG data (electrodes  $\times$  time) into a format that can be processed by the fully connected layers. Following the Flatten layer, the model comprised two dense layers. The first dense layer contained 64 neurons with ReLU activation, which helps in learning non-linear relationships within the data by introducing activation non-linearities. The second dense layer had 128 neurons, also with ReLU activation, to further capture intricate patterns. The final dense layer consisted of 3 neurons with softmax activation, providing a probability distribution over the following three classes: 'left', 'right', and 'none'.

The MLP model was trained using the Adam optimizer, which is known for its efficiency in handling large datasets and its adaptive learning rate capabilities. The loss function used was sparse categorical cross-entropy, appropriate for multi-class classification problems. Training was conducted over 10 epochs with a batch size of 512, which balanced the need for computational efficiency and the model's ability to learn from the data.

### 2.5.4. Transformer Model Approach

The Transformer model was implemented to classify motor imagery EEG signals, leveraging its advanced capabilities in capturing long-range dependencies and complex patterns within sequential data. The architecture of the Transformer model, originally designed for natural language processing tasks [24], has been adapted here to process EEG data, showcasing its versatility and effectiveness in time-series analysis.

The model begins with an input layer that takes the EEG data and projects them into an embedding dimension suitable for processing by the Transformer layers. Each Transformer encoder layer consists of a multi-head attention mechanism and a feed-forward neural network. The multi-head attention mechanism allows the model to focus on different parts of the input sequence simultaneously, capturing intricate dependencies within the EEG signals. This mechanism enhances the model's ability to understand the relationships and patterns present in the data, which is critical for accurate classification.

In each encoder layer, the multi-head attention output is passed through a dropout layer for regularization and then added to the original input through a residual connection. This addition is followed by layer normalization, which helps stabilize and accelerate training. The output of the layer normalization is fed into a feed-forward network composed of two dense layers: the first layer with a ReLU activation function to introduce non-linearity, and the second layer to project back to the embedding dimension. Another dropout layer is applied, and the result is again added to the input of the feed-forward network through a residual connection and normalized.

The architecture includes multiple such Transformer encoder layers stacked sequentially, allowing the model to build progressively more complex representations of the EEG data. After the final Transformer layer, the output is flattened to convert the 2D data into a 1D vector. This vector is then passed through a dense layer with a softmax activation function to classify the data into one of three categories: 'left', 'right', and 'none'.

The multi-head attention mechanism within the Transformer model enables it to handle the complexity of EEG signals by capturing dependencies across different time steps, making it highly effective for this type of sequential data. Additionally, the use of residual connections and layer normalization ensures that the model can be trained efficiently, overcoming issues such as vanishing gradients that often plague deep learning models.

### 2.6. Training and Validation

The training of deep learning models was conducted using a k-fold cross-validation approach on pre-processed EEG data. This method partitions the dataset into 'k' subsets, where the model is iteratively trained on 'k - 1' subsets and validated on the remain-



ing subset. This process ensures that the models are trained and evaluated on different combinations of the data, enhancing their generalization and robustness.

K-fold cross-validation is a popular method used in machine learning for model evaluation to assess the performance of a machine learning algorithm. It is particularly useful when the dataset is limited in size. In k-fold cross-validation, the original dataset is divided into 'k' subsets. The model is trained 'k' times, with each time using a different subset as the testing set and the remaining subsets as the training set.

The value of 'k' is established at 5, leading to the division of the dataset into 5 equally sized folds. The selection of 'k' is crucial as it has a substantial impact on the evaluation of the model's performance. The rationale behind opting for 'k = 5' is as follows: A larger 'k' value would entail a greater number of training and testing iterations, escalating the computational demands of the cross-validation process. Conversely, a diminished 'k' value could result in heightened variance in the performance estimation. With 'k = 5', each fold encompasses 80% of the data for training and 20% for testing, striking a harmonious equilibrium between the two. This ensures that the model is adequately trained on a diverse dataset while also being rigorously tested on novel data. Typically, a 'k' value ranging between 5 to 10 is deemed appropriate for most datasets. By adopting 'k = 5', a balanced trade-off is achieved between the model's bias and variance, rendering the performance estimation of the model robust and dependable. The rationale for choosing k = 5 in k-fold cross-validation is based on empirical evidence and the relevant literature, which demonstrate that k = 5 offers a practical balance between computational demands and performance estimation variance. Specifically, research by Nti et al. (2021) [25] suggests that in some cases, k = 5 offered better accuracy with Bayesian network models, indicating that k = 5 can provide reliable and slightly optimistic performance estimates. Additionally, a study by Fushiki (2009) [26] found that the bias of the cross-validation estimate of the prediction error was greatly reduced when k = 5 was used. These findings support the use of k = 5 as a common and effective choice in k-fold cross-validation, balancing the need for computational efficiency with the accuracy of performance estimation.

To further ensure the validity of the training–test split and prevent data leakage, several key practices were implemented. First, a stratified training–test split was employed to maintain the class distribution in both training and test sets. This technique ensures that each class is proportionally represented, providing a balanced evaluation of the model's performance across all classes. Stratification is critical in scenarios where class imbalances exist, as it prevents the model from being biased towards the majority class by ensuring that minority classes are adequately represented in both training and test sets. In this study, stratified sampling was employed to ensure the balance of class distribution in the dataset, which is crucial for maintaining the integrity and reliability of the model's training process. The specific dataset used in this study, the OpenBCI Community Dataset, comprises EEG data from 52 subjects performing motor imagery tasks. The dataset's class distribution includes three primary classes: left hand movement, right hand movement, and no action. The decision to use stratification was influenced by the inherent class distribution within this dataset. Each participant's EEG recordings contain multiple trials for each motor imagery task, creating a varied and comprehensive dataset. However, without stratification, there was a risk of having an imbalanced training set, which could lead to a biased model. For instance, if the majority of the data points were from one class (e.g., right hand movement), the model would become biased towards predicting that class more frequently, thereby reducing its overall accuracy and effectiveness.

Stratified sampling allowed us to maintain the proportional representation of each class in both the training and validation sets. This approach ensured that the model was exposed to a balanced mix of all classes during training, which is critical for learning the distinguishing features of each class accurately. Consequently, the model could generalize better when predicting new, unseen data. By preserving the class distribution, stratified sampling helped to mitigate the risks associated with the class imbalance, such as overfitting to the majority class or underperforming on minority classes. This method-

ological choice was pivotal in achieving the high classification accuracy reported in our study, demonstrating the model's robustness and its capability to handle real-time control tasks effectively.

Before splitting the data, they were shuffled to prevent any order-related biases, ensuring a random and unbiased distribution of samples. Shuffling disrupts any inherent order that might be present in the data collection process, such as temporal sequences or batch effects, which could otherwise introduce biases into the model training. This step is vital for enhancing the randomness and diversity of the training data, which in turn improves the model's ability to generalize to unseen data. Additionally, it was ensured that the training and test sets were completely separate, with no overlap, to avoid data leakage. Data leakage occurs when information from outside the training dataset is used to create the model, leading to overly optimistic performance estimates. By strictly partitioning the data, it is ensured that no test data were inadvertently used during the training process, thereby preserving the integrity of the evaluation metrics. This separation guarantees that the model's performance metrics reflect its true ability to generalize to new, unseen data.

For all models, the data preprocessing and splitting processes were consistent, ensuring a balanced class distribution and input dimension consistency. Each model was compiled using the Adam optimizer, known for its adaptive learning rate capabilities and efficiency in handling sparse gradients. The loss function used was sparse categorical cross-entropy, suitable for multi-class classification problems where labels are provided as integers. The Adam optimizer and sparse categorical cross-entropy loss function are justified for the multi-class classification on EEG data due to their specific alignment with the characteristics of EEG signals. EEG data are inherently noisy and exhibit high variability, which can complicate the training process. The Adam optimizer is well-suited for such data because it combines the advantages of both AdaGrad and RMSProp algorithms, providing an adaptive learning rate for each parameter. This adaptability helps in managing the noise and variability in EEG signals, ensuring more stable and efficient convergence during training. The sparse categorical cross-entropy loss function is appropriate for EEG data, which often involve multi-class classification tasks such as distinguishing between different motor imagery states, eliminating the need for one-hot encoding and thus simplifying the training process. It is particularly effective at handling the multi-class nature of EEG classification problems, ensuring accurate and computationally efficient model training.

Together, these choices enhance the model's robustness and performance, effectively addressing the noisy and complex nature of EEG data and improving the accuracy of multi-class classification tasks in brain-computer interface (BCI) applications. Accuracy was chosen as the primary metric to evaluate performance during both training and validation phases.

The training process involved using the `model.fit` function, with iterative training conducted over 10 epochs and a batch size of 512. During each fold of the cross-validation, the models' performances were evaluated on the validation data, providing insights into their generalization capabilities and monitoring potential overfitting or underfitting issues. Training and validation accuracy histories were recorded for subsequent analysis.

After the training phase, model predictions of the test set were obtained and compared to true labels. Classification reports were generated, detailing the precision, recall, and F1-score for each class and providing a comprehensive view of the models' performances in accurately classifying instances across the three categories. The neural network models used several hyperparameters, including a learning rate of 0.001 for the Adam optimizer. The loss function employed is Sparse Categorical Crossentropy. To find the optimal settings for these hyperparameters, GridSearchCV was utilized. Grid search systematically explores a specified parameter grid, evaluating each combination using cross-validation. The parameter grid included variations in batch size (32, 64, 128, and 512). The batch size affects the model's ability to generalize and the efficiency of the training process. Smaller batches can lead to more robust updates, while larger batches can speed up training. The number of epochs (10, 20, and 30) determines how many times the learning algorithm

will work through the entire training dataset. Exploring a range helps balance between underfitting and overfitting, ensuring the model is trained sufficiently without excessive training that might lead to overfitting. The learning rate (0.01, 0.001, and 0.0001) controls the step size during gradient descent. Exploring a range of values helps identify the optimal learning rate that balances convergence speed and stability. By training the model with these different combinations and evaluating their performance, GridSearchCV identifies the best combination of hyperparameters that yields the highest validation accuracy. This process ensures that the selected hyperparameters are well-suited for the specific dataset and model architecture, leading to improved model performance.

In practice, this involved wrapping the model creation function in a KerasClassifier to be compatible with GridSearchCV, defining the parameter grid, and running the grid search to determine the best settings. The final model was then trained using these optimized hyperparameters, leading to better overall performance and robustness in the model's predictions.

Confusion matrices were computed to visually represent classification results, indicating the number of true positives, false positives, true negatives, and false negatives for each class. These were visualized using the plot confusion matrix function, facilitating a clear understanding of the models' strengths and weaknesses in distinguishing between classes. Sensitivity (recall) and specificity were calculated from the confusion matrices, and these metrics were plotted to assess performance across different categories.

While the preprocessing, training, and evaluation processes were consistent across all methods, the architectural differences highlighted the unique strengths of each approach. Maintaining consistency across all methods in data preprocessing, training, and evaluation was critical for ensuring fair, reliable, and scientifically valid comparisons between models. Standardization controls extraneous variables, allowing the isolation of the effect of the model architecture on performance. This uniformity ensures that each model is evaluated on the same data points, avoiding biases and enabling fair comparisons of strengths and weaknesses. By minimizing the risk of introducing bias, standardizing parameters ensures that performance differences are due to the model architecture rather than training regime discrepancies.

### *2.7. Real-Time Implementation*

Upon successful training and validation, the trained deep learning models were deployed in real-time using Python scripts that interfaced with an Arduino board. An Arduino program was developed to facilitate the interaction between the Python script and the physical hardware, specifically the servo motors. A serial communication link was established between the Python environment and the Arduino board to transmit real-time EEG predictions from the model. This enabled the precise control of servo motors, facilitating the translation of EEG signals into actionable commands to control the system.

The implementation process involved continuously receiving EEG data in real-time, preprocessing the incoming data to remove noise and artifacts, and feeding them into the trained model for prediction. The model then classified the motor imagery tasks based on the extracted features and sent the corresponding commands to the Arduino board. The Arduino board, in turn, controlled the servo motors to execute the desired actions corresponding to the EEG predictions.

This real-time implementation demonstrated the practical application of the trained deep learning model in controlling the system based on the individual's motor imagery tasks. The integration of the model with the Arduino board and servo motors showcased the feasibility and effectiveness of using EEG signals to enable the precise and responsive control of external devices, highlighting the potential of BCIs in various applications.

### *2.8. Performance Evaluation*

The performance of the developed deep learning models was rigorously evaluated using test datasets to gauge its efficacy in classifying motor imagery tasks. Metrics including

accuracy, precision, recall, and the F1-score were computed to provide a comprehensive assessment of the model's classification capabilities. Additionally, confusion matrices and classification reports were generated to further analyze and understand the model's performance across different classes of motor imagery tasks.

The obtained results demonstrated the model's ability to accurately classify and differentiate between left hand movement, right hand movement, and no action scenarios based on EEG signals. High accuracy, precision, and recall values indicated the model's robustness and effectiveness at capturing the intricate patterns and features present in EEG data associated with various motor imagery tasks.

This evaluation process validated the efficacy of the proposed deep learning-based BCI systems in facilitating seamless human-machine interaction. By leveraging EEG signals and advanced deep learning techniques, the developed BCI systems showcased their potential in controlling servo motors with high precision and reliability. The successful integration of EEG data processing, feature extraction, and deep learning-based classification in a real-time setup demonstrated a promising step towards the advancement and practical application of BCIs in various domains.

### 3. Results

The results of this study substantiate the successful development and real-time implementation of a BCI system tailored for controlling servo motors based on EEG signals. The principal findings can be encapsulated below.

The study's outcomes underscore the immense potential of BCI technology in facilitating intuitive and efficient human-machine interactions, especially in the domain of robotic control. The seamless integration of the BCI system with hardware platforms like Arduino not only validates its effectiveness but also paves the way for innovative advancements and future research in the realm of BCIs and assistive technology.

Moreover, the movement of the servo motors exhibited commendable performance, with minor adjustments made to the delay to achieve smoother and more precise movements. This fine-tuning ensured that the servo responded accurately to the EEG predictions, enhancing the overall functionality and reliability of the system.

#### 3.1. Evaluation Metrics

This section provides a comparative analysis of the performance metrics for the RNN, MLP, Transformer, and CNN-LSTM models. Table 3 summarizes precision, recall, F1-score, and support for each class, as well as the overall accuracy, macro average, and weighted average.

The CNN-LSTM model achieves the highest overall accuracy at 98%, outperforming the RNN and MLP models, which have accuracies of 95% and 94%, respectively. The Transformer model, however, records the lowest accuracy at 78%.

In terms of class-specific performance, the CNN-LSTM model consistently excels. For Class 0, it outperforms all other models, while the Transformer model exhibits the lowest metrics. Similarly, in Class 1, the CNN-LSTM maintains superior performance. Although the MLP model demonstrates high precision, its lower recall results in a slightly diminished F1-score, with the Transformer model again ranking lowest. For Class 2, both the RNN and CNN-LSTM models show robust performance, with the latter achieving the highest scores. The MLP model has high recall but slightly reduced precision, and the Transformer model remains the weakest.

When examining macro and weighted averages, the CNN-LSTM model stands out with the highest performance across all metrics and classes, followed by the RNN model. While the MLP model shows commendable performance, it is slightly lower than the RNN and CNN-LSTM models. The Transformer model, despite being effective for certain tasks, exhibits the lowest overall performance. Future work could focus on optimizing the Transformer model to enhance its accuracy and consistency across different classes.

**Table 3.** Performance metric summary for different models.

Model	Class	Precision	Recall	F1-Score	Support
RNN	0	0.94	0.96	0.95	44,118
	1	0.94	0.94	0.94	44,136
	2	0.97	0.96	0.96	44,121
	Accuracy			0.95	132,375
	Macro Avg.	0.95	0.95	0.95	132,375
	Weighted Avg.	0.95	0.95	0.95	132,375
MLP	0	0.93	0.94	0.93	43,996
	1	0.96	0.88	0.92	44,066
	2	0.92	0.99	0.95	44,313
	Accuracy			0.94	132,375
	Macro Avg.	0.94	0.94	0.93	132,375
	Weighted Avg.	0.94	0.94	0.93	132,375
Transformer	0	0.74	0.78	0.76	43,984
	1	0.73	0.77	0.75	44,183
	2	0.89	0.80	0.84	44,208
	Accuracy			0.78	132,375
	Macro Avg.	0.79	0.78	0.78	132,375
	Weighted Avg.	0.79	0.78	0.78	132,375
CNN-LSTM	0	0.97	0.99	0.98	44,288
	1	0.98	0.98	0.98	44,234
	2	0.99	0.98	0.98	43,853
	Accuracy			0.98	132,375
	Macro Avg.	0.98	0.98	0.98	132,375
	Weighted Avg.	0.98	0.98	0.98	132,375

### 3.2. Statistical Analysis

#### ANOVA

The ANOVA is used to determine if there are statistically significant differences between the means of three or more independent groups. In this case, we applied the ANOVA to compare the precision scores of four different models: RNN, MLP, Transformer, and CNN-LSTM.

Here are the detailed results and their implications.

#### F-statistic: 10.095

The F-statistic is a ratio of the variance between the group means to the variance within the groups. A higher F-statistic indicates a greater degree of separation between the group means relative to the within-group variability.

In this context, an F-statistic of 10.095 suggests that there is a considerable variation in precision scores among the different models compared to the variation within the models.

#### p-value: 0.0043

The *p*-value measures the probability that the observed differences in precision scores occurred by chance under the null hypothesis (which states that there are no differences in means).

A *p*-value of 0.0043 is much lower than the common significance level of 0.05, indicating strong evidence against the null hypothesis.

This low  $p$ -value suggests that the differences in precision scores among the models are statistically significant.

The ANOVA results indicate that at least one model’s precision score is significantly different from the others. Since the  $p$ -value is less than 0.05, we can conclude that not all models have the same precision performance.

**Model Comparisons**

RNN: Precision scores for the RNN model across classes are 0.94, 0.94, and 0.97.

MLP: Precision scores for the MLP model are 0.93, 0.96, and 0.92.

Transformer: Precision scores for the Transformer model are 0.74, 0.73, and 0.89.

CNN-LSTM: Precision scores for the CNN-LSTM model are 0.97, 0.98, and 0.99.

Transformer: This model has the lowest precision scores, indicating it performs worse compared to the other models in terms of precision.

CNN-LSTM: This model has the highest precision scores, suggesting it performs best in terms of precision.

RNN and MLP: Both models have high precision scores, with MLP showing slightly better performance in some classes than the RNN.

**Tukey’s HSD Post Hoc Analysis Results**

The Tukey’s HSD (Honestly Significant Difference) test was conducted to determine which specific pairs of models have significant differences in their precision scores. Table 4 shows the results.

**Table 4.** Tukey’s HSD test.

Group 1	Group 2	Mean Difference	$p$ -Value	Lower Bound	Upper Bound	Reject Null Hypothesis
CNN-LSTM	MLP	0.0533	0.3497	−0.0492	0.1559	No
CNN-LSTM	RNN	0.04	0.5202	−0.0625	0.1426	No
CNN-LSTM	Transformer	0.2233	0.0019	0.1207	0.3258	Yes
MLP	RNN	−0.0133	0.9596	−0.1159	0.0892	No
MLP	Transformer	0.17	0.0101	0.0674	0.2726	Yes
RNN	Transformer	0.1833	0.0049	0.0807	0.2859	Yes

CNN-LSTM vs. Transformer: The difference in precision is 0.2233, and the  $p$ -value is 0.0019, indicating a significant difference. This suggests that CNN-LSTM has significantly higher precision than the Transformer model.

MLP vs. Transformer: The difference in precision is 0.17, and the  $p$ -value is 0.0101, indicating a significant difference. This suggests that MLP has significantly higher precision than the Transformer model.

RNN vs. Transformer: The difference in precision is 0.1833, and the  $p$ -value is 0.0049, indicating a significant difference. This suggests that RNN has significantly higher precision than the Transformer model.

**Non-significant Differences:**

CNN-LSTM vs. MLP: The difference in precision is 0.0533, and the  $p$ -value is 0.3497, indicating no significant difference.

CNN-LSTM vs. RNN: The difference in precision is 0.04, and the  $p$ -value is 0.5202, indicating no significant difference.

MLP vs. RNN: The difference in precision is −0.0133, and the  $p$ -value is 0.9596, indicating no significant difference.

The post hoc analysis reveals that the Transformer model has significantly lower precision compared to the other models (CNN-LSTM, MLP, and RNN). However, there are no significant differences in precision between CNN-LSTM, MLP, and RNN. This further emphasizes the relatively poor performance of the Transformer model in terms of precision.

### Friedman Test

As a non-parametric alternative to ANOVA, the Friedman test is suitable for comparing more than two related groups. In this case, it was applied to the precision scores across the classes for each model. The test determines if there are significant differences in the precision scores among the models, accounting for the related nature of the data (precision scores for the same classes). Table 5 shows the results.

**Table 5.** Friedman test results.

Statistic	Value
Test Statistic ( $\chi^2$ )	8.20
<i>p</i> -value	0.0421

Since the *p*-value (0.0421) is less than the common significance level of 0.05, we reject the null hypothesis. This indicates that there are significant differences in the precision scores among the models. This reinforces the findings from the ANOVA and Tukey's HSD test.

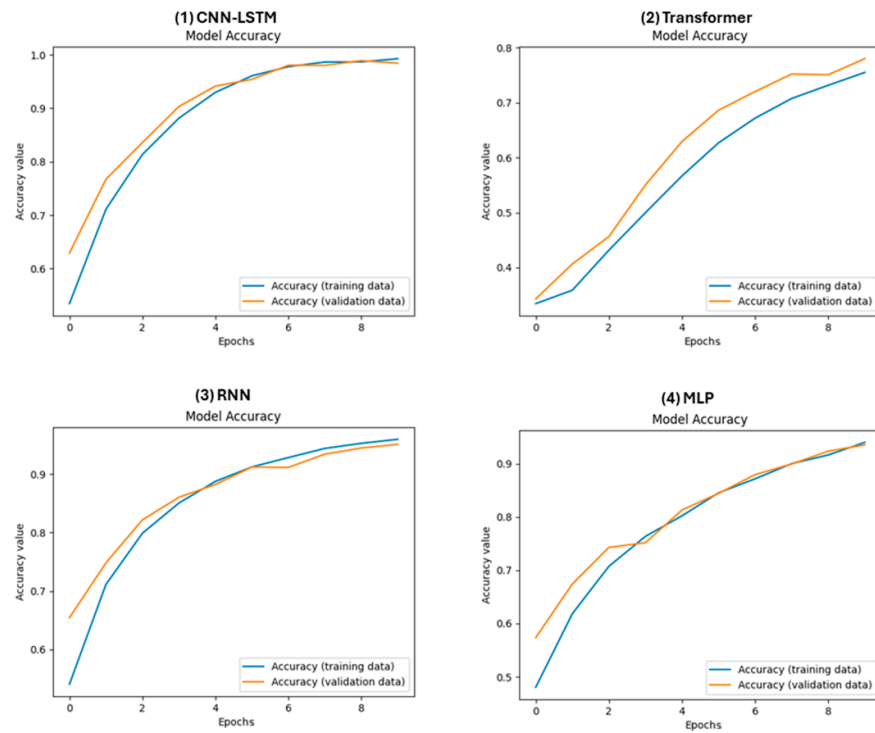
## 4. Discussion

In this study, various deep learning architectures were evaluated for their ability to classify motor imagery tasks from EEG signals, including RNN, MLP, Transformer, and CNN-LSTM models. Despite Transformer's known proficiency in capturing long-range dependencies and handling sequential data, its performance was inferior in this context. This can be attributed to the relatively small EEG dataset, which is insufficient for training Transformer effectively, and the inherent noise and high dimensionality of EEG signals. Transformer demands substantial computational resources and large, diverse datasets to fully leverage its potential.

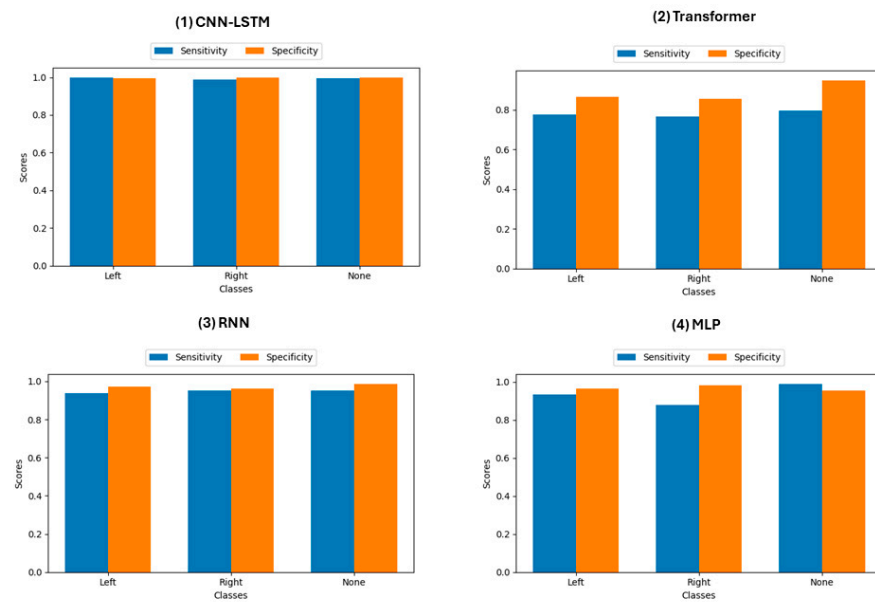
The RNN model, although proficient in handling sequential data, exhibited lower accuracy than the CNN-LSTM model. The vanishing gradient problem hampers the RNN's ability to learn long-term dependencies, which are critical for capturing extended temporal dynamics in EEG signals. Additionally, the RNN's sensitivity to noise and variations in EEG data, along with its high computational demands, contribute to its lower performance in this study. MLP model, while capable of capturing complex patterns through dense layers, demonstrated lower accuracy compared to CNN-LSTM. MLPs' limitations lie in their inability to exploit the sequential nature of EEG data, as they are designed for tabular data processing and lack mechanisms to capture temporal dependencies. This shortcoming makes them less effective at handling the high dimensionality and noise of EEG signals without advanced preprocessing and feature extraction techniques.

The CNN-LSTM model was chosen for its superior ability to capture both spatial and temporal features of EEG signals, making it particularly effective for the motor imagery classification. CNN layers excel at extracting spatial features and identifying local patterns and hierarchies within the EEG signals. These spatial features are then passed to LSTM layers, which capture temporal dependencies and sequential patterns. This hybrid approach leverages the strengths of both the CNN and LSTM, providing a comprehensive understanding of the data. The CNN layers reduce data dimensionality and highlight relevant features, enabling the LSTM layers to learn temporal dynamics without being overwhelmed by noise. This dual-layered approach improves the prediction accuracy and contributes to the reliability and responsiveness of the BCI system, ensuring the precise control of robotic devices based on user intentions.

The evaluation metrics presented in Figures 4 and 5, along with the confusion matrices in Figure 6, validate the efficacy and reliability of the proposed deep learning-based BCI system in accurately classifying and controlling robotic devices using EEG signals. High accuracy, precision, and recall values, along with minor adjustments to the servo motors, affirm the system's potential and pave the way for its practical application in various domains, fostering a seamless human-machine interaction.



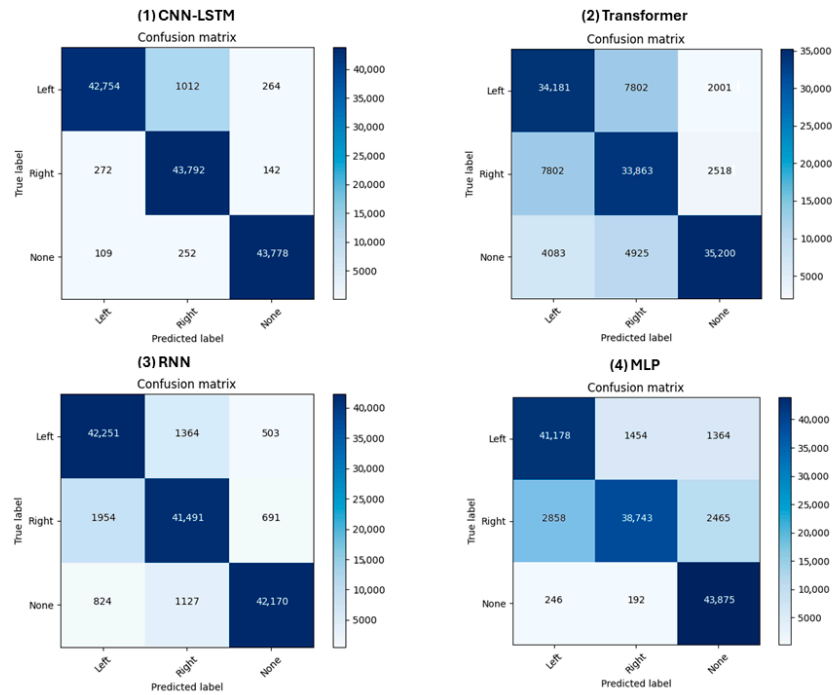
**Figure 4.** Accuracy of the training dataset (blue line) and the validation dataset (orange line) with respect to number of epochs for the four models.



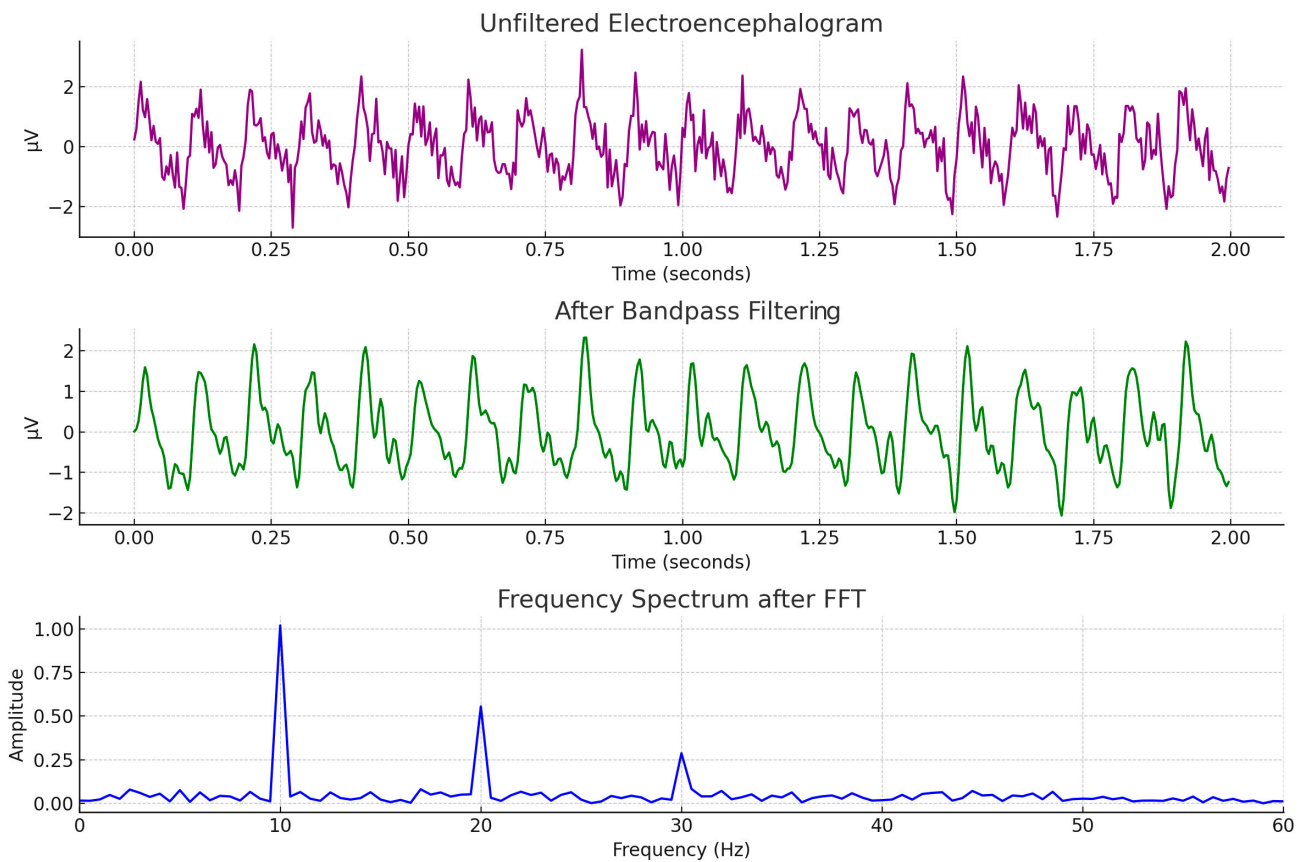
**Figure 5.** The figure presents bar charts comparing sensitivity (blue) and specificity (orange) across the following different classes: 'left', 'right', and 'none'.

A fourth-order Butterworth bandpass filter was applied with a frequency range of 1 Hz to 50 Hz to the raw EEG signals to isolate the desired frequency components while eliminating unwanted low and high-frequency noise. Following this, we utilized the fast Fourier transform (FFT) to convert the time-domain EEG signals into the frequency domain, allowing us to analyze the spectral components more effectively. Figure 7 illustrates the frequency spectra of the EEG signals both before and after the application of the Butterworth bandpass filter.





**Figure 6.** The confusion matrices visualize the models’ performances by comparing the predicted labels to the true labels. The numbers in the matrix indicate the count of each type of prediction, showing where the model is making correct and incorrect predictions. The colors represent the density of the counts, with darker shades indicating higher counts.



**Figure 7.** Frequency spectra of EEG signals before and after Butterworth bandpass filtering.

This visualization effectively demonstrates the impact of the preprocessing steps, showing how the Butterworth bandpass filter isolates the desired EEG components and eliminates noise. The frequency spectrum after FFT closely follows the filtered signal, with a significant attenuation of frequencies above 50 Hz, providing a clear representation of the EEG signal's primary frequency components.

In research focused on comparing the performance of various machine learning models, it is essential to employ robust statistical methods to identify significant differences in model metrics. This analysis evaluates four models, RNN, MLP, Transformer, and CNN-LSTM, with the primary metric of interest being precision, measured across three distinct classes for each model.

To determine if there are significant differences in the precision scores among these models, several statistical tests were utilized. Analysis of variance (ANOVA) was chosen first because it is designed to compare the means of three or more independent groups. In this context, ANOVA was employed to evaluate whether there are statistically significant differences in the precision scores of the four models. This test helps in understanding if any model performs differently in terms of precision compared to the others, providing a global view of the differences across all models.

Following a significant result from the ANOVA test, Tukey's honestly significant difference (HSD) post hoc test was used to conduct pairwise comparisons between the models. Tukey's HSD test was selected because it is effective at identifying which specific models' precision scores differ significantly from each other after establishing that there are overall differences. It provides detailed insights into the relative performance of each model against the others, helping to pinpoint where the significant differences lie, as shown in Table 2.

Additionally, the Friedman test was conducted as a non-parametric alternative to ANOVA. This test is suitable for comparing more than two related groups, which is particularly relevant for this analysis as the precision scores for each class can be considered related. The Friedman test was applied to the precision scores across the classes for each model, as the results in Table 3 show, to determine if there are significant differences among the models while accounting for the related nature of the data (precision scores for the same classes).

The results of these statistical tests suggest that the CNN-LSTM model has the highest precision, indicating it as the best-performing model among those evaluated. The Friedman test results highlight that the significant differences between the models are primarily due to the poor performance of the Transformer model in terms of precision. This comprehensive analysis guides further efforts in model selection and improvement, ensuring that the chosen models offer the best performance based on the precision metric.

The study's findings underscore the potential of BCI technology in enabling an intuitive and efficient human-machine interaction, particularly in robotic control applications. The successful implementation of the BCI system and its integration with hardware platforms such as Arduino demonstrate promising avenues for future research and development in BCIs and assistive technology. The robust performance of the CNN-LSTM model in accurately predicting motor imagery tasks highlights its effectiveness in capturing and interpreting complex brainwave patterns associated with motor actions that are crucial for reliable robotic control based on user intentions. Seamless integration with an Arduino board enables the real-time control of servo motors, demonstrating the feasibility of translating EEG predictions into actionable commands. The Python script effectively interfaces with Arduino, ensuring the precise execution of the desired actions, which is essential for applications requiring rapid and accurate responses, such as assistive technology and neurorehabilitation. The replication of this work is facilitated by the detailed specifications provided, including comprehensive descriptions of data preprocessing, model training, and real-time control, along with specifications of the hardware setup. This ensures that other researchers can easily replicate the experiments and build upon these findings.

This study significantly advances BCIs by developing accurate methods for predicting actions from EEG signals. Potential applications include assistive technology, prosthetics, and neurorehabilitation, enhancing the autonomy and quality of life for individuals with disabilities. The integration of BCI systems with hardware platforms like Arduino showcases the practical feasibility of such technologies.

However, several limitations must be acknowledged. The study considered a specific range of motor imagery tasks, and the use of a relatively small EEG dataset is a common practice to ensure the feasibility of initial research efforts. The use of publicly available datasets, while ensuring reproducibility and comparability, does come with certain constraints. These datasets may not encompass the full variability observed in real-world applications, which can impact the model's generalizability. While this study did not include testing of unseen out-of-distribution data, which is a valuable step for validating BCI systems, it lays a solid foundation for future research that can address this aspect.

The use of 5-fold cross-validation, while standard, was employed to ensure robustness and reliability. This approach provides a balanced evaluation and ensures that the model's performance is robust across different data splits, but to further enhance the model's robustness, future studies should incorporate more extensive validation techniques.

Despite these limitations, the research remains promising, suggesting that future evaluations of commercial BCI headsets, incorporating advancements in deep learning and robotics, will be conducted. EEG signal variability among individuals can pose challenges in real-world deployment. Future research should explore a broader range of motor imagery tasks to enhance the system's applicability, incorporate larger and more representative datasets to improve model training, and develop personalized calibration procedures and adaptive algorithms to enhance the system robustness and user experience. Integrating advanced machine learning techniques and neuroimaging methods could further improve the spatial and temporal resolution of EEG signals.

## 5. Conclusions

In conclusion, this study showcases the promising potential of a BCI system designed to control servo motors through EEG signals. The successful fusion of advanced machine learning algorithms with hardware platforms represents a significant step forward in the realm of human-machine interaction. The application of BCI technology has immense implications for enhancing the quality of life of individuals with motor impairments. By enabling direct communication between the brain and external devices, BCIs can offer a newfound sense of accessibility and autonomy to those with limited mobility. This not only empowers individuals to interact with their environment more effectively but also promotes a greater degree of independence and self-sufficiency.

Moreover, the findings of this research underscore the transformative potential of BCI technology in the field of assistive technology. As technology continues to advance and as we gain a deeper understanding of the intricacies of brain signals, we can anticipate further breakthroughs in BCI systems. These advancements hold the promise to reshape the landscape of human-machine collaboration, paving the way for more intuitive and seamless interactions between humans and machines.

The novelty of this paper lies in its comprehensive comparative analysis and the implementation of an advanced model in real-time hardware testing. Unlike previous studies that primarily focus on either enhancing classification accuracy or feature extraction, this research uniquely integrates and evaluates a variety of neural network architectures, including RNNs, MLPs, Transformer models, and CNN-LSTM networks. This extensive comparison highlights the superior performance of the CNN-LSTM model in effectively capturing both spatial and temporal features from EEG signals. While earlier works have successfully implemented the real-time control of devices, this paper distinguishes itself by not only achieving a high classification accuracy but also rigorously validating the model's performance in a real-time setting. By continuously testing and providing immediate feedback through detailed performance metrics, the study underscores its

practical applicability and potential for real-world BCI applications. This dual focus on theoretical advancements and practical implementation significantly contributes to the field, offering a resilient framework for future human–machine interaction technologies.

In essence, this study not only validates the current capabilities of BCI systems but also points towards a future where assistive technology is more sophisticated, inclusive, and capable of enhancing the overall quality of life for individuals with disabilities.

**Author Contributions:** Conceptualization, D.A. and P.A.; methodology, D.A.; software, D.A.; validation, D.A., P.A., S.K., and E.C.V.; formal analysis, D.A.; investigation, D.A.; resources, D.A.; data curation, D.A.; writing—original draft preparation, D.A.; writing—review and editing, D.A.; visualization, P.A.; supervision, D.A.; project administration, D.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the University of West Attica, [ELKE, Special Account for Research Grants, grand protocol number 25432/24-07-2024].

**Institutional Review Board Statement:** The EEG data used in this study were sourced exclusively from publicly available datasets provided by the OpenBCI Community. The dataset consists of deidentified EEG recordings available for research purposes. As the data are publicly accessible, anonymized, and previously collected by the contributing entity, the study did not involve direct interactions with human subjects or participant recruitment. Consequently, Ethics Committee or Institutional Review Board (IRB) approval was considered not applicable, aligning with guidelines exempting such research from requiring ethical approval due to the absence of risk to individual privacy or welfare.

**Informed Consent Statement:** Not applicable for this research. In the initial data collection, all subjects gave written informed consent to collect information on brain signals and were paid for their participation. The data collected were used only for research purposes [13].

**Data Availability Statement:** Publicly available datasets were analyzed in this study. These data can be found at <https://openbci.com/community/publicly-available-eeg-datasets/> (accessed on 3 March 2024).

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Blinowska, K.; Durka, P. Electroencephalography (EEG). *Sensors* **2006**, *21*, 4626.
2. Hill, N.J.; Wolpaw, J.R. Brain–Computer Interface. In *Reference Module in Biomedical Sciences*; Elsevier: Amsterdam, The Netherlands, 2016; ISBN 978-0-12-801238-3.
3. Niedermeyer, E. *Niedermeyer's Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*; Oxford University Press: Oxford, UK, 2017; ISBN 978-0-19-022851-4.
4. Zozos, I.; Kakkos, I.; Miloulis, S.T.; Anastasiou, A.; Ventouras, E.M.; Matsopoulos, G.K. Applying Neural Networks with Time-Frequency Features for the Detection of Mental Fatigue. *Appl. Sci.* **2023**, *13*, 1512. [[CrossRef](#)]
5. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
6. Wang, P.; Jiang, A.; Liu, X.; Shang, J.; Zhang, L. LSTM-Based EEG Classification in Motor Imagery Tasks. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2018**, *26*, 2086–2095. [[CrossRef](#)] [[PubMed](#)]
7. Lin, J.-S.; She, B.-H. A BCI System with Motor Imagery Based on Bidirectional Long-Short Term Memory. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *719*, 012026. [[CrossRef](#)]
8. Garcia-Moreno, F.M.; Bermudez-Edo, M.; Rodríguez Fórtiz, M.J.; Garrido, J. A CNN-LSTM Deep Learning Classifier for Motor Imagery EEG Detection Using a Low-invasive and Low-Cost BCI Headband. In Proceedings of the 2020 16th International Conference on Intelligent Environments (IE), Madrid, Spain, 20–23 July 2020; pp. 84–91.
9. Li, H.; Ding, M.; Zhang, R.; Xiu, C. Motor imagery EEG classification algorithm based on CNN-LSTM feature fusion network. *Biomed. Signal Process. Control* **2022**, *72*, 103342. [[CrossRef](#)]
10. Martín-Chinea, K.; Ortega, J.; Gómez-González, J.F.; Pereda, E.; Toledo, J.; Acosta, L. Effect of time windows in LSTM networks for EEG-based BCIs. *Cogn. Neurodyn.* **2023**, *17*, 385–398. [[CrossRef](#)]
11. Angelakis, D.; Ventouras, E.; Asvestas, P. Brain Signals to Actions Using Machine Learning. *Eng. Proc.* **2023**, *50*, 7. [[CrossRef](#)]
12. Anguita, D.; Ghelardoni, L.; Ghio, A.; Oneto, L.; Ridella, S. The ‘K’ in K-fold Cross Validation. In Proceedings of the ESANN 2012 Proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, Belgium, 25–27 April 2012.
13. Publicly. EEG Datasets. Available online: <https://openbci.com/community/publicly-available-eeg-datasets/> (accessed on 3 March 2024).

14. Schalk, G.; McFarland, D.J.; Hinterberger, T.; Birbaumer, N.; Wolpaw, J.R. EEG Motor Movement/Imagery Dataset. *PhysioNet* **2009**, *18*, 2011. [[CrossRef](#)]
15. EEG Motor Movement/Imagery Dataset. Available online: <https://www.kaggle.com/datasets/brianleung2020/eeg-motor-movementimagery-dataset> (accessed on 29 March 2024).
16. Biosemi EEG ECG EMG BSPM NEURO Amplifier Electrodes. Available online: [https://www.biosemi.com/Products\\_ActiveTwo.htm](https://www.biosemi.com/Products_ActiveTwo.htm) (accessed on 24 June 2024).
17. Mohamed, A.F.; Jusas, V. Comprehensive Analysis of Feature Extraction Methods for Emotion Recognition on Motor Imagery from Multichannel EEG Recordings. In *Proceedings of the Good Practices and New Perspectives in Information Systems and Technologies*; Rocha, Á., Adeli, H., Dzemyda, G., Moreira, F., Poniszewska-Marañda, A., Eds.; Springer: Cham, Switzerland, 2024; pp. 211–231.
18. Moctezuma, L.A.; Molinas, M. Classification of low-density EEG for epileptic seizures by energy and fractal features based on EMD. *J. Biomed. Res.* **2020**, *34*, 180–190. [[CrossRef](#)] [[PubMed](#)]
19. Barone, J.; Rossiter, H.E. Understanding the Role of Sensorimotor Beta Oscillations. *Front. Syst. Neurosci.* **2021**, *15*, 655886. [[CrossRef](#)] [[PubMed](#)]
20. Klimesch, W. EEG alpha and theta oscillations reflect cognitive and memory performance: A review and analysis. *Brain Res. Brain Res. Rev.* **1999**, *29*, 169–195. [[CrossRef](#)] [[PubMed](#)]
21. Hidalgo, D.; Yeh, S.-S.; Lee, J.-I. A Frequency Domain Approach for Tuning Control Parameters of CNC Servomotors to Enhance its Circular Contouring Accuracy. *Procedia CIRP* **2017**, *63*, 372–377. [[CrossRef](#)]
22. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [[CrossRef](#)]
23. Taud, H.; Mas, J.F. Multilayer Perceptron (MLP). In *Geomatic Approaches for Modeling Land Change Scenarios*; Camacho Olmedo, M.T., Paegelow, M., Mas, J.-F., Escobar, F., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 451–455, ISBN 978-3-319-60801-3.
24. Abibullaev, B.; Keutayeva, A.; Zollanvari, A. Deep Learning in EEG-Based BCIs: A Comprehensive Review of Transformer Models, Advantages, Challenges, and Applications. *IEEE Access* **2023**, *11*, 127271–127301. [[CrossRef](#)]
25. Nti, I.; Nyarko-Boateng, O.; Aning, J. Performance of Machine Learning Algorithms with Different K Values in K-fold Cross-Validation. *Int. J. Inf. Technol. Comput. Sci.* **2021**, *6*, 61–71. [[CrossRef](#)]
26. Fushiki, T. Estimation of prediction error by using K-fold cross-validation. *Stat. Comput.* **2011**, *21*, 137–146. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.