

Article

# Boosting EEG and ECG Classification with Synthetic Biophysical Data Generated via Generative Adversarial Networks

Archana Venugopal and Diego Resende Faria \* 

School of Physics, Engineering and Computer Science, University of Hertfordshire,  
Hatfield AL10 9AB, Hertfordshire, UK; za22abg@herts.ac.uk

\* Correspondence: d.faria@herts.ac.uk

**Abstract:** This study presents a novel approach using Wasserstein Generative Adversarial Networks with Gradient Penalty (WGAN-GP) to generate synthetic electroencephalography (EEG) and electrocardiogram (ECG) waveforms. The synthetic EEG data represent concentration and relaxation mental states, while the synthetic ECG data correspond to normal and abnormal states. By addressing the challenges of limited biophysical data, including privacy concerns and restricted volunteer availability, our model generates realistic synthetic waveforms learned from real data. Combining real and synthetic datasets improved classification accuracy from 92% to 98.45%, highlighting the benefits of dataset augmentation for machine learning performance. The WGAN-GP model achieved 96.84% classification accuracy for synthetic EEG data representing relaxation states and optimal accuracy for concentration states when classified using a fusion of convolutional neural networks (CNNs). A 50% combination of synthetic and real EEG data yielded the highest accuracy of 98.48%. For EEG signals, the real dataset consisted of 60-s recordings across four channels (TP9, AF7, AF8, and TP10) from four individuals, providing approximately 15,000 data points per subject per state. For ECG signals, the dataset contained 1200 real samples, each comprising 140 data points, representing normal and abnormal states. WGAN-GP outperformed a basic generative adversarial network (GAN) in generating reliable synthetic data. For ECG data, a support vector machine (SVM) classifier achieved an accuracy of 98% with real data and 95.8% with synthetic data. Synthetic ECG data improved the random forest (RF) classifier's accuracy from 97% with real data alone to 98.40% when combined with synthetic data. Statistical significance was assessed using the Wilcoxon signed-rank test, demonstrating the robustness of the WGAN-GP model. Techniques such as discrete wavelet transform, downsampling, and upsampling were employed to enhance data quality. This method shows significant potential in addressing biophysical data scarcity and advancing applications in assistive technologies, human-robot interaction, and mental health monitoring, among other medical applications.

**Keywords:** synthetic data generation; GAN; WGAN-GP; EEG; ECG



**Citation:** Venugopal, A.; Resende Faria, D. Boosting EEG and ECG Classification with Synthetic Biophysical Data Generated via Generative Adversarial Networks.

*Appl. Sci.* **2024**, *14*, 10818. <https://doi.org/10.3390/app142310818>

Academic Editor: Christos Bouras

Received: 24 October 2024

Revised: 18 November 2024

Accepted: 20 November 2024

Published: 22 November 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

In the rapidly advancing fields of neuroscience and artificial intelligence, generating realistic biophysical data has emerged as a critical focus. EEG data are particularly significant, as they provide valuable insights into the intricate workings of the human brain. The generation of synthetic EEG data holds immense potential for applications in brain-computer interfaces (BCIs), robotic movement control, neuroscience research, and cognitive assessment tools. Similarly, ECG waves, which record the electrical signals generated via the heart's rhythmic contractions, are essential for understanding cardiovascular health. They play a vital role in medical diagnostics, monitoring heart conditions, and guiding treatments. Beyond traditional healthcare, ECG signals have found applications in human-computer interfaces (HCIs), enabling the control of devices such as wearable health monitors and driving innovative solutions across diverse fields [1,2]. The ability to generate synthetic ECG data enables researchers to simulate and examine cardiac behaviors

without relying on large, real-world datasets. This can expedite the development of health-care technologies and improve machine learning models for more accurate predictions in cardiac-related applications.

According to Salehi et al. [3], the development of GANs has provided a powerful tool for generating synthetic data that closely resemble real data. Deep learning methods have shown significant potential to enhance decoding performance, but their effectiveness often hinges on the availability of large datasets. The generation of synthetic data addresses several critical challenges in this domain. Traditional EEG data acquisition is labor-intensive and requires extensive subject participation and careful calibration to ensure high-quality data, as noted by [4]. Nik et al. [5] emphasize that, while large datasets are crucial for training machine learning models, collecting high-quality EEG data is particularly challenging due to its subject- and session-dependent nature, which necessitates precise calibrations. Similarly, Galván et al. [6] highlight that the accurate identification of motor imagery (MI) patterns in EEG signals is constrained by data-related limitations, hindering the practical implementation of such systems. Furthermore, Chen et al. [7] point out that, while BCIs offer non-invasive communication methods, their efficiency is heavily dependent on individual training data, often acquired during lengthy calibration sessions.

Another significant challenge in accessing medical data is the heavy restrictions imposed due to its sensitive nature, limiting its availability for research and clinical training purposes. As noted by Chaurasia et al. [8], unique brainwave patterns of individuals can be used for authentication techniques, making EEG data a form of biometric information and, therefore, highly sensitive. Zhang et al. [9] explore EEG-based biometric cryptosystems for authentication, further underscoring their sensitivity. Standard de-identification techniques, aimed at facilitating data sharing, are often insufficient to fully protect the privacy of individuals in the dataset, as mentioned by Delaney et al. [10]. For ECG data, imbalances are a frequent issue because abnormal cases are relatively rare [11]. Additionally, the use of real patient ECGs is heavily regulated due to privacy concerns. This creates a constant demand for additional ECG data, particularly for training machine learning models in automatic diagnosis, which perform better with balanced datasets. The dependency on human subjects also raises ethical concerns related to privacy and consent.

Synthetic EEG data provide a way to circumvent many of these issues, enabling innovative research and practical applications. Furthermore, synthetic data can reduce commercial risks in product development for neurotechnology and BCI applications by offering a reliable and scalable source of training data. It addresses challenges such as insufficient or unreliable data, extended timelines for real-world data collection, and the difficulties of obtaining ethical approval for human participants. By minimizing the need for extensive data collection and navigating regulatory landscapes with anonymized data, synthetic data also significantly reduces costs.

The main contribution of this research is the development of a novel approach to generating synthetic EEG and ECG data using WGAN-GP combined with our CNN architecture to improve data classification. The synthetic EEG data closely resemble real data for concentration and relaxation states, while the synthetic ECG data correspond to normal and abnormal states. This method increases the dataset size, enhancing the generalization and performance of machine learning models. The application of WGAN-GP for generating synthetic biophysical data represents a significant advancement in medical data generation and analysis, enabling improved diagnostics and research. By leveraging deep learning techniques, the model produces high-fidelity synthetic waves, which augment real datasets—a crucial step for training machine learning models, as larger and more diverse datasets lead to better performance and generalization. Our approach has demonstrated superior accuracy, with the generated synthetic EEG data being effectively classifiable into concentrated and relaxed states. Additionally, the inclusion of synthetic ECG data has been shown to improve classification accuracy, further highlighting the potential of this approach in biomedical and other research fields that depend on biophysical data-driven applications.

The structure of this paper is outlined as follows. Section 2 explores background research, reviewing the relevant literature. In Section 3, we introduce our proposed work and outline its development. Section 4 showcases our experimental results and insights. Finally, Section 5 presents the conclusions drawn from our findings and outlines avenues for future research.

## 2. Related Work

Introduced by Goodfellow et al. [12] in 2014, GANs have become a significant generative technique. Fatemeh Fahimi et al. [13] highlight the rising interest in using GANs for EEG data generation due to their success in mimicking the temporal, spectral, and spatial features of authentic EEG signals. Since their inception, various GAN types have been developed to overcome initial limitations. Conditional GANs (cGANs) were introduced by Mirza and Osindero in 2014 [14]. Deep convolutional GANs (DCGANs) were early GAN adaptations that improved training by using deep convolutional neural networks (CNN) for the discriminator and generator [15]. Arjovsky et al. [16] introduced Wasserstein GAN (WGAN). Gulrajani et al. [17] further presented the WGAN-GP as a solution to WGAN restrictions. WGAN-GP could provide more stable training, preventing issues like mode collapse and leading to better overall performance of the GAN.

Despite advancements in GAN architectures, several challenges persist, particularly in the context of EEG data generation. Habashi et al. [18] reviewed the application of GANs across various EEG domains, including motor imagery, P300, rapid serial visual presentation (RSVP), emotion recognition, and epilepsy. However, limited attention has been given to generating EEG signals related to specific mental states, such as concentration and relaxation. The review also highlighted the challenges in creating diverse synthetic data that adequately capture the complexity of EEG signals.

Cheng et al. [19] introduced the “SleepEGAN” model, which demonstrated the potential of GANs in generating minority class samples for sleep-stage classification. However, its architecture was specifically designed for sleep studies, limiting its generalizability to other EEG applications. This underscores a critical gap in developing GAN models capable of generalizing across different mental states and EEG applications.

Shin et al. [20] explored GAN-based data augmentation and anonymization in medical imaging. While their approach showed promise in the medical imaging domain, its application to EEG data generation remains underexplored. Hazra and Byun’s “SynSigGAN” model [21] generates four types of synthetic biomedical signals, including EEG data related to epilepsy diagnosis and seizure classification. However, their results indicated that the synthetic data were overly similar to the real dataset, limiting its diversity and utility for training machine learning models.

Salazar et al. [22] introduced the generative adversarial network synthesis for over-sampling (GANSO), a novel method designed to improve classifier training with extremely limited data. GANSO integrates vector Markov random fields (vMRF) with GANs to synthesize data while preserving the structural properties of the original dataset. Its generative block uses graph Fourier transform to maintain graph connectivity, while the discriminative block ensures that the synthetic data are indistinguishable from the original. GANSO demonstrates strong potential for biomedical applications, particularly in scenarios with limited training samples. Unlike GANSO, WGAN-CG operates in the feature space of time-series data, such as ECG and EEG, without explicitly modeling graph dependencies. In this work, we focus on feature-space GANs for time-series data, with plans for future research to incorporate graph-based models. This will enable a direct comparison between feature-space and graph-based methods, such as GANSO, to evaluate their respective efficiencies.

Zhao et al. [23] demonstrated that WGAN-GP could generate precise and varied EEG signals with improved spectral performance, aiding dataset expansion for traditionally hard-to-collect data. However, their work focused solely on P300 brain waves, with a limited analysis of real-world applicability and signal effectiveness across diverse contexts.

Additionally, insufficient data preprocessing resulted in the need for 2000 epochs to achieve satisfactory results, highlighting both computational inefficiency and the necessity for more robust preprocessing techniques to reduce training time and enhance signal quality.

The limitations identified in previous studies highlight the need for further research into generating EEG data specific to the mental states of concentration and relaxation. While existing models have shown some success in generating EEG signals, they are often tailored to specific applications (e.g., sleep stages and epilepsy) and fail to produce data that are sufficiently diverse and representative of the complex nature of EEG signals. Moreover, the reliance on specific architectures, such as those used in “SleepEGAN,” restricts the versatility of these models.

To better understand the classification of brain waves based on their frequency, amplitude, shape, and other characteristics, as noted in [24], it is essential to recognize that these frequencies vary with different mental states, as illustrated in Table 1.

**Table 1.** Different types of brainwaves.

Frequency (Hz)	Wave	Description
30–100	Gamma	Problem-solving, concentration
13–30	Beta	Awake state, excitement, thinking
8–13	Alpha	Daydreaming, inability to focus, restful
4–8	Theta	Drowsiness, reduced consciousness, sleep
0–4	Delta	Deep sleep, loss of bodily awareness

Evidence from the literature suggests that people with high levels of beta and gamma activity are in a state of concentration, while people with considerable alpha activity are in a state of relaxation. There are cases where the usage of real EEG is considered a privacy breach. Schiliro et al. [25] mention that brain data can reveal private mental states, necessitating cognitive privacy protections against unauthorized access and collection. In [26], the authors mention that privacy-preserving methods like homomorphic encryption (HE) are limited by high computational demands, noise buildup, and restricted applicability. These limitations further indicate the necessity of generating synthetic data to address privacy concerns effectively.

Similar to EEG generation, there are several studies related to synthetic ECG generation. In [27], the authors discuss various approaches to the synthetic generation of ECG using GANs, variational autoencoder–decoders (VAEs), and large language models (LLMs), as well as the limitations of each. The study in [28] focuses primarily on privacy aspects and does not extensively cover the performance or accuracy of the generated synthetic ECGs. In [11], the authors focus on normal cardiac cycles and do not address the generation of abnormal ECG patterns, which are crucial for diagnostic purposes. Apart from GAN-based models, some research has been conducted using GPT models for biophysical data generation. While models like ChatEMG [29] and the GPT-2-based model [30] generate unlimited EMG signal sequences, the proposed WGAN-GP model generates high-quality synthetic data specific to each state, thus achieving higher accuracy in generating classifiable EEG signals and surpassing the fixed sequence limitation. The EEG signals generated via [30] using GPT-2 will be compared to our approach in subsequent sections.

This study sought to address these gaps by proposing a novel approach using WGAN-GP for the generation of EEG and ECG signals. The proposed model incorporates efficient data preprocessing techniques, including the use of discrete wavelet transform (DWT) with wavelet Db2 and level 5, to capture both high- and low-frequency components of EEG signals. This approach not only reduces the training time but also enhances the diversity and utility of the synthetic data in training machine learning models.

Moreover, this research examines the impact of using real, synthetic, and combined datasets to determine the optimal proportion of synthetic data required to enhance classification accuracy in real-world BCI applications. For EEG signals, we adopted the approach outlined in the research by Manoharan and Faria [31], which achieved notable success in

classifying EEG data into mental states with an accuracy of 92% using a CNN classifier. For ECG signals, classification is performed using support vector machine (SVM) and random forest (RF) classifiers due to their effectiveness and robustness. SVMs are particularly adept at handling high-dimensional data, and they can deliver accurate results even with limited sample sizes. For instance, a study demonstrated that SVM-based arrhythmic beat classification effectively identified heart-related abnormalities in ECG signals [32]. Similarly, RF classifiers, known for their ensemble learning approach, provide strong generalization capabilities and resilience against overfitting, making them highly suitable for physiological data analysis. Research by [33] compared the performance of RF and SVM for ECG quality assessment, finding that both classifiers, when combined with nonlinear features, effectively assessed ECG quality. These studies underscore the reliability of SVM and RF classifiers in ECG analysis within complex biomedical datasets.

### 3. Materials and Methods

This section outlines the roadmap for generating synthetic EEG and ECG waveforms that closely mimic real data and can be accurately classified into different states.

#### 3.1. Datasets

For EEG, this study utilizes a dataset originally published by Bird et al. [34], which is openly accessible on both GitHub and Kaggle. As the dataset does not contain any identifiable personal information, ethical approval for its use is not required. Bird et al. [34] collected data from four individuals (two male and two female) over 60 s intervals in three cognitive states: relaxed, concentrating, and neutral. EEG signals were recorded using a Muse EEG headband with dry electrodes positioned at the TP9, AF7, AF8, and TP10 sites. The Muse band includes a right aux channel designed as a placeholder for an auxiliary electrode; however, without an attached electrode, this channel produces only noise, and it was excluded from this study's analysis. Each .csv file in the dataset is structured with five columns: a timestamp, and EEG readings from the TP9, AF7, AF8, TP10, and right aux channels (the right aux channel is excluded from all calculations).

For ECG, we utilized a publicly available dataset from Kaggle [35]. Each row in this dataset represents a complete ECG recording for a subject, consisting of 140 data points. Columns 0–139 contain the ECG readings, while column 140 provides the labels—either 0 or 1—indicating whether the ECG is normal or abnormal.

#### 3.2. EEG Data Preprocessing

Although it is possible in many situations to train a neural network using raw datasets, raw data training is ineffective due to the temporal and auto-correlated nature of EEG waves, as mentioned in Chiu et al. [36]. Also, processing raw data, in many cases, can lead to computational losses and time consumption. Preprocessing data, therefore, becomes especially crucial for EEG data.

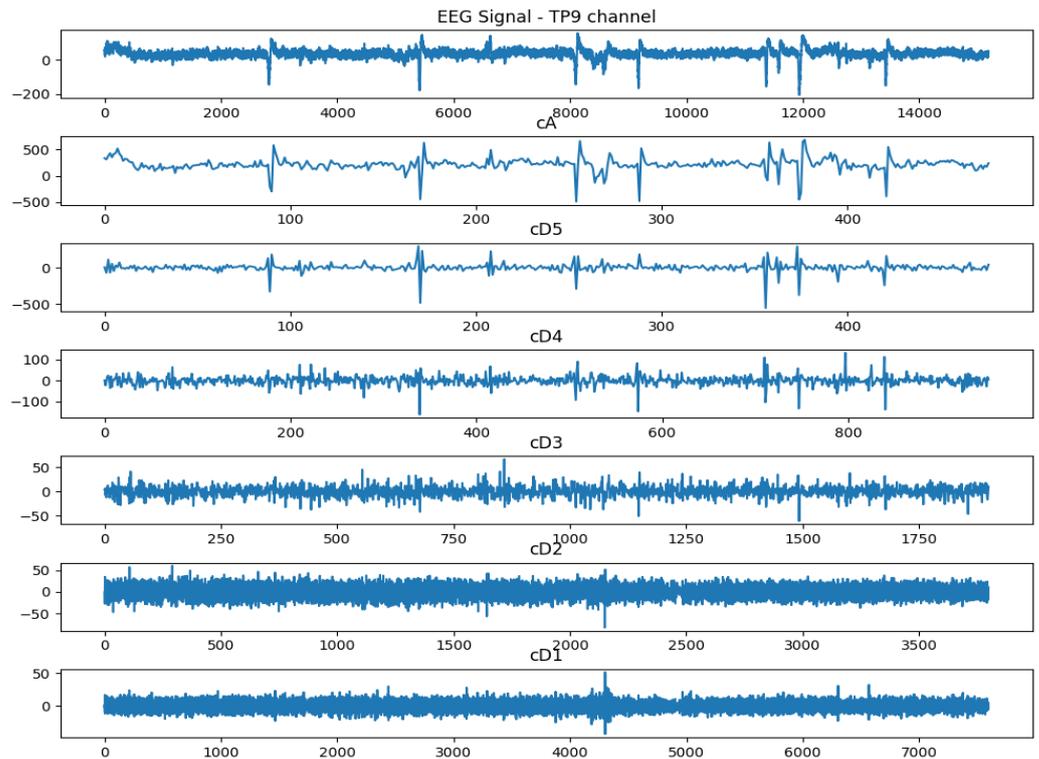
##### 3.2.1. Discrete Wavelet Transforms

Wavelet transforms are particularly valuable for EEG analysis because they capture both localized and global features of the signal at different scales. This capability enables the identification of transient events, such as spikes or artifacts, as well as the detection of rhythmic patterns and frequency changes associated with various mental states or neurological conditions. Classification tasks that utilize feature extraction with discrete wavelet transforms (DWTs) on EEG signals during complex tasks have demonstrated an accuracy of 98% using support vector machines, multi-layer perceptrons, and K-nearest neighbor classifiers [37].

While DWT is not exclusive to EEG, it has also been shown to be effective for ECG (electrocardiogram) signal processing [38]. The *Db2* wavelet is particularly suitable for EEG analysis because it closely resembles the spike-wave patterns observed in EEG data [39].

Additionally, level 5 decomposition with *Db2* provides valuable insights for preprocessing EEG signals, contributing to improved classification accuracy.

The approximation coefficient (*cA*) and detailed coefficient (*cD*[1–5]) of the DWT signal of channel TP9 are plotted in Figure 1. The approximation coefficient represents the low frequency of the EEG wave capturing the overall trend or baseline activity. Detailed coefficients (*cD*[1–5]) represent the high-frequency components at different levels of decomposition. *cD*1 captures high-frequency details at the first level, *cD*2 refines the details further, *cD*3 provides even finer resolution, and so on up to *cD*5 (Level 5).



**Figure 1.** Five decomposition of EEG waves using discrete wavelet transform (DWT) into approximation coefficient (*cA*) and detailed coefficients (*cD*<sub>1–5</sub>).

Let us denote the EEG signal as  $x[k]$ . The approximation and detailed coefficient at level 1 are represented in Equations (1) and (2).

Approximation coefficient:

$$a_1[n] = \sum_k x[k] * h[2n - k]. \tag{1}$$

Detailed coefficient:

$$d_1[n] = \sum_k x[k] * g[2n - k], \tag{2}$$

where  $h$  and  $g$  are the low-pass and high-pass filters associated with the *Db2* wavelet,  $a_1[n]$  is the approximation coefficient,  $d_1[n]$  is the detailed coefficient,  $n$  is the index of the output coefficients at level 1,  $x[k]$  is the original signal at the sample index  $k$ , and the factor  $2n - k$  ensures downsampling, as DWT operates on every second element to create the next level. This procedure can be implemented in Python using the library PyWavelet by calling `pywt.wavedec`.

### 3.2.2. Downsampling

Downsampling is performed to reduce the number of samples, simplifying computations while preparing input for the WGAN-GP. High-frequency coefficients from lower levels—*cD*1, *cD*2, and *cD*3—are selected for downsampling because they contain more data

points and can tolerate this reduction. These coefficients typically represent noise or fine details, which can be adequately captured with fewer samples. This process helps decrease the data size while preserving essential information needed for further analysis. Conversely, high-level coefficients— $cD4$ ,  $cD5$ , and  $cA$ —are critical for capturing the overall shape and slower variations in the signal. Downsampling these components could result in the loss of critical information. By selectively downsampling lower-level detail coefficients ( $cD1$ ,  $cD2$ , and  $cD3$ ), we optimize the data size and computational efficiency without significantly compromising the signal reconstruction quality. Avoiding downsampling for higher-level coefficients ( $cD4$ ,  $cD5$ , and  $cA$ ) ensures the preservation of low-frequency trends and essential signal features. After DWT and downsampling are applied, the processed wave is provided as input to the WGAN-GP model.

### 3.3. WGAN-GP Model Development

Once the data are preprocessed, they can be fed into the GAN model. GANs consist of two neural networks: the generator and the discriminator. As described by Broll et al. [40], the discriminator's objective is to distinguish between the generator's output and the real training data, while the generator aims to produce samples that closely resemble the training data. To address the learning instabilities associated with GANs, the WGAN was introduced.

While the original GAN focuses on minimizing the Jensen–Shannon (JS) divergence between the real distribution  $P_r$  and the generator distribution  $P_g$ , the WGAN optimizes the Wasserstein distance between  $P_r$  and  $P_g$ . Also called the Earth mover's distance ( $EM(p, q)$ ), this metric represents the minimum cost required to move elements of one distribution ( $q$ ) to match another distribution ( $p$ ), where the cost is calculated as the product of the mass and the transport distance [18]. To further enhance the stability of the WGAN, the gradient penalty (GP) was introduced. The WGAN-GP adds a gradient penalty term to the objective function, which helps enforce the Lipschitz constraint required for the Wasserstein distance. This penalty term ensures that the gradients of the discriminator ( $D$ ) are well behaved, leading to improved gradient quality during training. This approach promotes more stable learning and better convergence [41].

The objective function of WGAN-GP is given in Equation (3):

$$L = E_{\tilde{x} \in P_g} [D(\tilde{x})] - E_{x \in P_r} [D(x)] + \lambda E_{\hat{x} \in P_{\tilde{x}}} [(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2], \quad (3)$$

where  $\lambda$  is the gradient penalty coefficient, and  $\hat{x}$  is generated by interpolating between real data  $x$  and the generated data  $\tilde{x}$ , as shown in Equation (4).

$$\hat{x} = \varepsilon x + (1 - \varepsilon)\tilde{x}, \varepsilon \in \text{uniform}[0, 1], x \in P_r, \tilde{x} \in P_g. \quad (4)$$

This interpolation helps enforce the Lipschitz continuity requirement by computing gradients at these points.

#### 3.3.1. Algorithm: WGAN-GP with Advanced Architecture

This section explains the architecture of the WGAN-GP model. Algorithm 1 outlines the steps for model definition, data preparation, gradient penalty implementation, and training with  $\omega_0$  as the initial discriminator and  $\theta_0$  as the initial generator parameter.

Input:  $\lambda = 10$  (gradient penalty coefficient),  $train\_data$  (training dataset),  $num\_epochs = 100$  (number of epochs),  $batch\_size = 100$  (size of each training batch), and  $lr = 0.001$  (learning rate). Figure 2 represents the WGAN-GP architecture. The output of the WGAN-GP model consists of generated synthetic coefficients for  $cA$ ,  $cD5$ ,  $cD4$ ,  $cD3$ ,  $cD2$ , and  $cD1$  for each of the 4 channels.

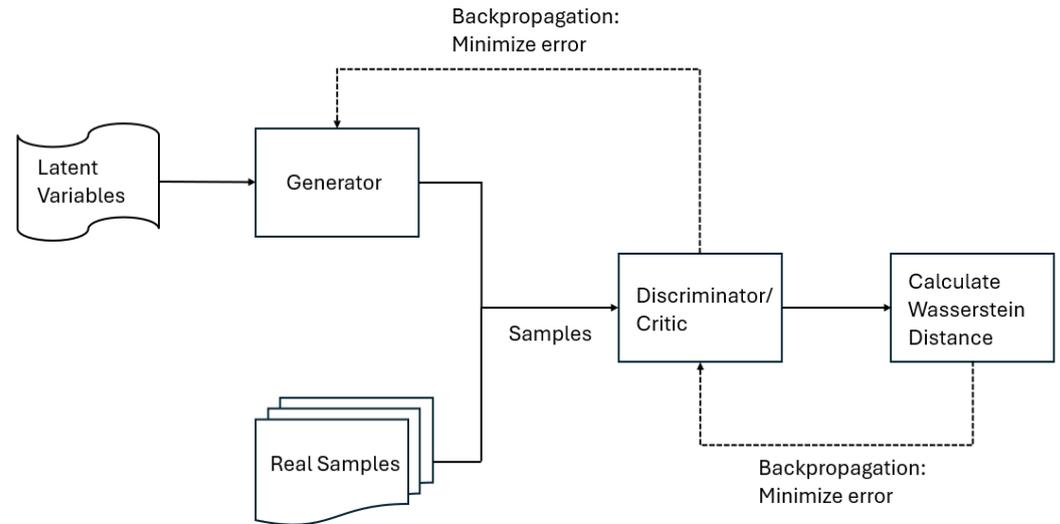


Figure 2. WGAN-GP architecture.

---

#### Algorithm 1 WGAN-GP training

---

- 1: **Define** generator model:  $G_\theta(z) = \text{ReLU}(W_4 \cdot \text{ReLU}(W_3 \cdot \text{ReLU}(W_2 \cdot \text{ReLU}(W_1 \cdot z))))$
- 2: **Define** discriminator model:
- 3:  $D_\omega(x) = \text{LeakyReLU}(W_4 \cdot \text{Dropout}(\text{LeakyReLU}(W_3 \cdot \text{Dropout}(\text{LeakyReLU}(W_2 \cdot \text{Dropout}(\text{LeakyReLU}(W_1 \cdot x)))))))$
- 4: Initialize generator and discriminator models:  $G_\theta$  and  $D_\omega$
- 5: Initialize optimizers: Adam optimizers for both models with learning rate  $lr$  and  $\beta$
- 6: **for** epoch = 1, ..., num\_epochs **do**
- 7:   Randomly permute training data indices
- 8:   **for**  $i = 1, \dots, m$  **do**
- 9:     Sample a batch of  $m$  real data  $x(i) \sim P_{data}$ , and  $m$  noisy data  $z(i) \sim P_z$
- 10:     Generate fake data:  $\tilde{x} \leftarrow G_\theta(z)$
- 11:     Interpolate data:  $\hat{x} \leftarrow \epsilon x + (1 - \epsilon)\tilde{x}$
- 12:     Compute gradient penalty:  $\nabla_{\hat{x}} D_\omega(\hat{x}), L_{GP} = \lambda(\|\nabla_{\hat{x}} D_\omega(\hat{x})\|_2 - 1)^2$
- 13:     Compute discriminator loss:  $L_D = D_\omega(\tilde{x}) - D_\omega(x) + L_{GP}$
- 14:     Update discriminator:  $\omega \leftarrow \text{Adam}(\nabla_\omega L_D)$
- 15:   **end for**
- 16:   Every  $k$  iterations, update the generator:
- 17:   **if**  $i \bmod (5 \times m) = 0$  **then**
- 18:     Generate fake data:  $\tilde{x} \leftarrow G_\theta(z)$
- 19:     Compute generator loss:  $L_G = -D_\omega(\tilde{x})$
- 20:     Update generator:  $\theta \leftarrow \text{Adam}(\nabla_\theta L_G)$
- 21:   **end if**
- 22: **end for**
- 23: **return**: trained  $G_\theta$

---

#### Generator

The generator model was designed to transform a single latent variable into a high-dimensional output through multiple layers. Each layer is progressively larger, facilitating the generation of complex patterns:

Input layer: A single dimension input.

Hidden layers: Four fully connected layers with increasing units (256, 512, 1024, and 2048), each followed by ReLU activation.

Output layer: A single-dimension output to match the real data's dimensionality.

### Discriminator

The discriminator model aims to distinguish between real and generated data through several layers, incorporating dropout for regularization and LeakyReLU for stable gradient flow:

Input layer: A single-dimension input.

Hidden layers: Four fully connected layers with decreasing units (512, 256, and 128), each followed by LeakyReLU activation (with a negative slope of 0.2) and dropout (0.3).

Output layer: A single-dimension output representing the authenticity score.

### Training Procedure with WGAN-GP

Training GANs is challenging due to their instability. WGAN-GP improves stability by enforcing a Lipschitz constraint with a gradient penalty. The training process involves alternating updates to the discriminator and generator:

1. Initialize models and optimizers: Initialize the generator,  $G_\theta$ , and discriminator,  $D_\omega$ , models. Both are optimized using the Adam optimizer learning rate,  $lr = 0.001$ , and betas (0.5, 0.999).
2. Gradient penalty (GP): GP is calculated to enforce the Lipschitz constraint.
3. Discriminator update: the discriminator is trained on real and fake samples; the loss is calculated, to which the gradient penalty is added and backpropagated.
4. Generator update: the generator is updated less frequently (every 5 batches) to ensure discriminator training.

### 3.4. EEG Synthetic Data: Post-Generation Processing Steps

#### 3.4.1. Upsampling

The output of the WGAN-GP model for EEG consists of synthetic data corresponding to each wavelet component:  $cD5, cD4, cD3, cD2, cD1$ , and  $cA$ . The higher-frequency components ( $cD3, cD2$ , and  $cD1$ ) are then upsampled or interpolated, as they were downsampled prior to being fed into the WGAN-GP network. Downsampling was employed to reduce data complexity and enable the network to focus on lower-frequency patterns. However, after generating the synthetic data, the high-frequency components must be restored to their original resolution in order to preserve fine details and signal accuracy. This upsampling or interpolation process ensures that the reconstructed data retains the essential high-frequency information required for realistic signal generation. The interpolation was performed using the `scipy.interpolate` module.

Figure 3 shows the detailed framework for generating synthetic data using the WGAN-GP model, including the necessary pre-processing, post-processing steps, and testing of the synthetic data with a classifier.

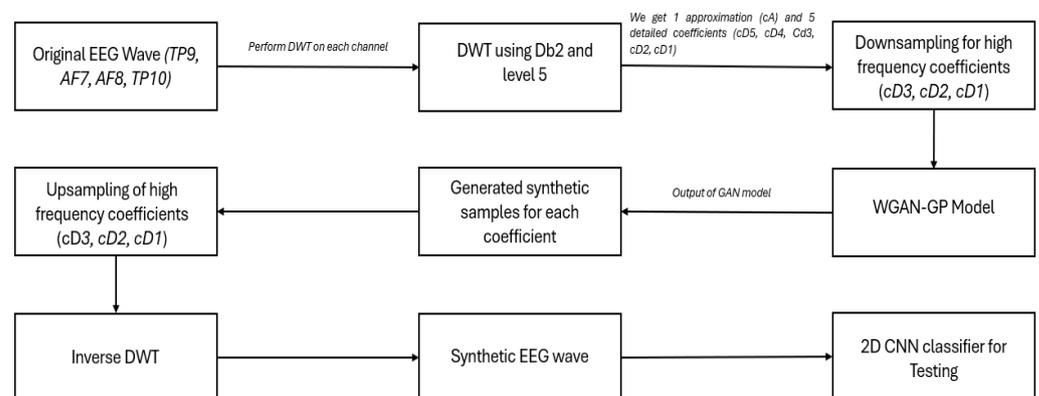


Figure 3. Workflow of synthetic EEG wave generation using WGAN-GP model.

#### 3.4.2. Inverse Discrete Wavelet Transform (IDWT):

Afterward, we perform the inverse discrete wavelet transform (IDWT), a mathematical operation that reconstructs a signal from its wavelet coefficients. This operation is the

counterpart to the discrete wavelet transform (DWT) used earlier to decompose the real EEG signal into its various frequency components ( $cA$ ,  $cD5$ ,  $cD4$ ,  $cD3$ ,  $cD2$ , and  $cD1$ ). While DWT decomposes the signal into approximation and detail coefficients at different scales, the inverse DWT reassembles these components to reconstruct the original signal. This process ensures that the signal retains the characteristics and features necessary for accurate representation and analysis.

To carry out the reconstruction, we used the `pywt.waverec` function from the Py-Wavelets library [42]. After reconstructing the generated samples, the next step involved classifying the generated EEG signals into their corresponding mental states for evaluation purposes. We utilized a CNN-based classifier from [31] to classify the signals into concentration and relaxation states.

### 3.5. ECG Data Processing

For the ECG data, given its lower complexity compared to EEG and the limited data size per individual (140 data points), we omitted the use of DWT, inverse DWT, downsampling, and upsampling techniques. Instead, the data were directly input into the WGAN-GP model, as outlined in Section 3.3, with only an adjustment to the learning rate (set to 0.0001). All other model parameters remained unchanged:

1. `num_epochs = 100`
2. `batch_size = 100`
3. `gradient_penalty_coef = 10`
4. `betas = 0.5, 0.999`

The synthetic data generated via WGAN-GP often exhibit sudden, unnatural spikes or abrupt transitions, contrasting with the consistent, rhythmic patterns typically seen in real ECG data. To address this, a smoothing process with a window size of 10 is applied to the synthetic ECG output, reducing spikes and enhancing its resemblance to the original dataset. This step is specifically required for ECG data, as the periodic, smooth pattern is essential for accurate representation. Conversely, for EEG data, this smoothing step is unnecessary due to the inherent presence of spike-wave patterns in natural EEG signals. Ultimately, smoothing the synthetic ECG data enhances the quality of the generated output, aligning it more closely with real ECG patterns and thereby making it more suitable for subsequent analysis or training applications.

### 3.6. 2D CNN for EEG Classification

To classify the dataset (original and synthetic data), we adopted the methodology from previous work [31], which demonstrated success in classifying EEG data. The 2D-CNN model employed in this study was specifically designed to classify mental states. The raw EEG data are preprocessed via their segmentation into overlapping windows of one second to extract relevant features. Each window is transformed into a 2D matrix using either a continuous wavelet transform (CWT) or a discrete wavelet transform (DWT), depending on the specific configuration. From the five signals extracted ( $\alpha$ ,  $\beta$ ,  $\theta$ ,  $\delta$ ,  $\gamma$ ), statistical features such as the mean, skewness, kurtosis, maximum, and minimum values are extracted. Additionally, frequency components are derived through fast Fourier transform (FFT) analysis. These extracted features serve as inputs to the CNN model. The CNN architecture consists of two convolutional layers with 32 and 64 filters, respectively, followed by max-pooling and dropout layers for regularization. The pooled feature maps are flattened and passed through dense layers with ReLU activation. The final layer employs a SoftMax activation function to classify the input into three distinct categories. This architecture effectively captures the temporal and spectral characteristics of EEG signals, enabling accurate mental state classification. It is important to note that the final 2D-CNN architecture consists of five CNNs, as aforementioned, one for each signal, followed by its feature extraction, and then the five CNNs are merged in the flatten layer for the final classification.

Figure 4 illustrates the workflow for data acquisition, pre-processing, feature selection, and CNN-based classification into mental states. Algorithm 2 describes the CNN classifier algorithm. For more details related to the classification process, refer to [31].

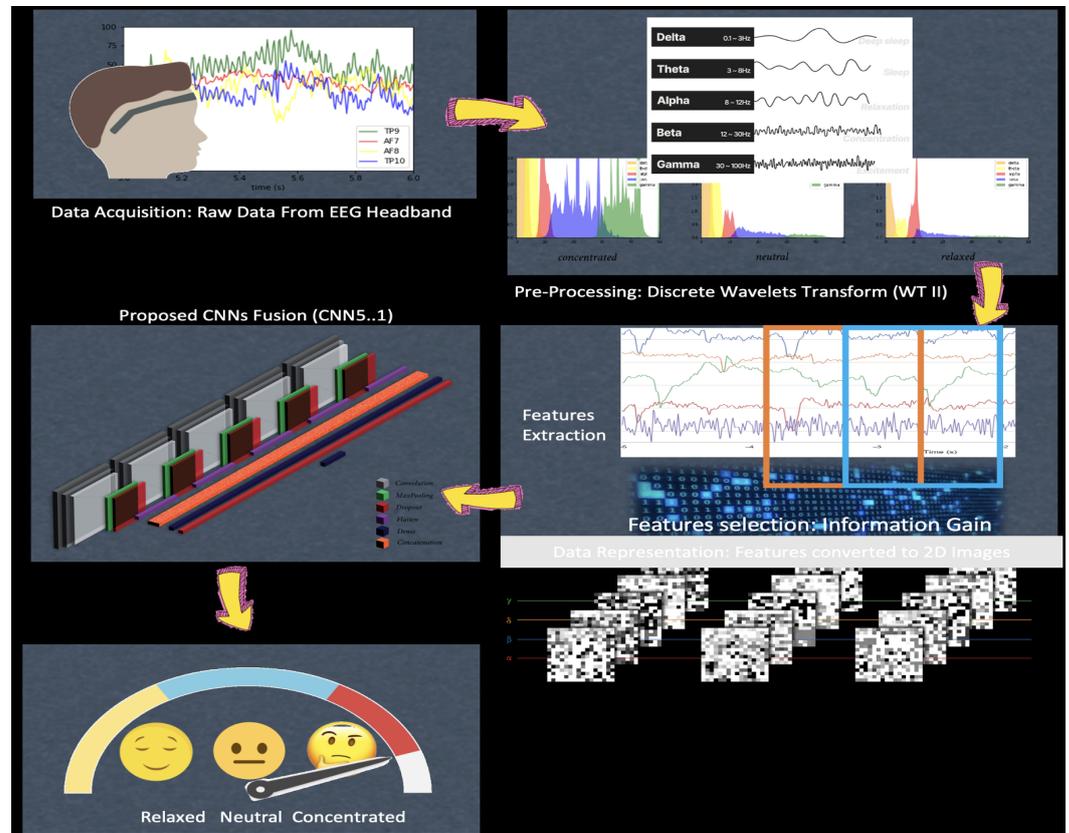


Figure 4. Two-dimensional CNN for EEG classification.

**Algorithm 2** Mental state classification merging 5 CNNs

- 1: **Input:** EEG signals as grayscale images of size  $16 \times 16$  for wave types  $\{\alpha, \beta, \theta, \delta, \gamma\}$
- 2: **Output:** Classified mental state: concentrated, neutral, or relaxed
- 3: **Step 1: Preprocessing EEG Data**
- 4: **for** each EEG signal **do**
- 5:     Apply discrete wavelet transform (DWT) to extract  $\{\alpha, \beta, \theta, \delta, \gamma\}$
- 6:     Compute wavelet coefficients:  $c_{jk} = \sum_n x[n]\psi_{jk}(n)$
- 7:     Reconstruct signal with inverse transform:  $x[n] = \sum_j \sum_k c_{jk}\psi_{jk}(n)$
- 8:     Extract statistical, spatio-temporal, and frequency-based features for each wavelet
- 9:     Transform features into  $16 \times 16$  grayscale images
- 10: **end for**
- 11: **Step 2: CNN Layers for Each Image**
- 12: **for** each EEG grayscale image **do**
- 13:     **Convolutional Layer 1:**
- 14:     **for** each filter  $f \in \text{Conv1}$  **do**
- 15:         **for** each pixel  $(i, j)$  in the image **do**
- 16:              $Z_{ij}^{(1)} = \text{ReLU}\left(\sum_{m=1}^3 \sum_{n=1}^3 W_{mn}^{(1)} X_{(i+m)(j+n)} + b^{(1)}\right)$
- 17:         **end for**
- 18:     **end for**

**Algorithm 2** *Cont.*


---

```

19: Convolutional Layer 2:
20: for each filter  $f \in \text{Conv2}$  do
21:   for each pixel  $(i, j)$  in the output from Conv1 do
22:      $Z_{ij}^{(2)} = \text{ReLU}\left(\sum_{m=1}^3 \sum_{n=1}^3 W_{mn}^{(2)} Z_{(i+m)(j+n)}^{(1)} + b^{(2)}\right)$ 
23:   end for
24: end for
25: Max-Pooling Layer:
26: for each pooling region  $(i, j)$  do
27:    $P_{ij} = \max_{p,q \in [0,1]} \left( Z_{(i+p)(j+q)}^{(2)} \right)$ 
28: end for
29: Dropout Layer (0.25): Apply dropout with a 25% probability during training
30: Flatten Layer: Flatten the pooled output  $P_{ij}$  into a 1D vector
31: end for
32: Step 3: Concatenation Layer Concatenate the flattened outputs from all 5 CNNs:
33:  $C = [F_\alpha, F_\beta, F_\theta, F_\delta, F_\gamma]$ 
34: Step 4: Fully Connected (Dense) Layer
35: for each unit  $u$  in the Dense Layer (512 units) do
36:    $Y_u^{(l)} = \text{ReLU}(W_u^{(l)} \cdot C + b_u^{(l)})$ 
37: end for
38: Step 5: Dropout Layer (0.5) Apply dropout with a 50% probability
39: Step 6: Output Layer (Dense + SoftMax)
40: for each class  $i$  (concentrated, neutral, relaxed) do
41:   Compute the SoftMax probability:  $\hat{y}_i = \frac{\exp(z_i)}{\sum_{j=1}^3 \exp(z_j)}$ 
42: end for
43: Step 7: Classification
44: return: The class with the highest probability =  $\arg \max(\hat{y}_i)$ 

```

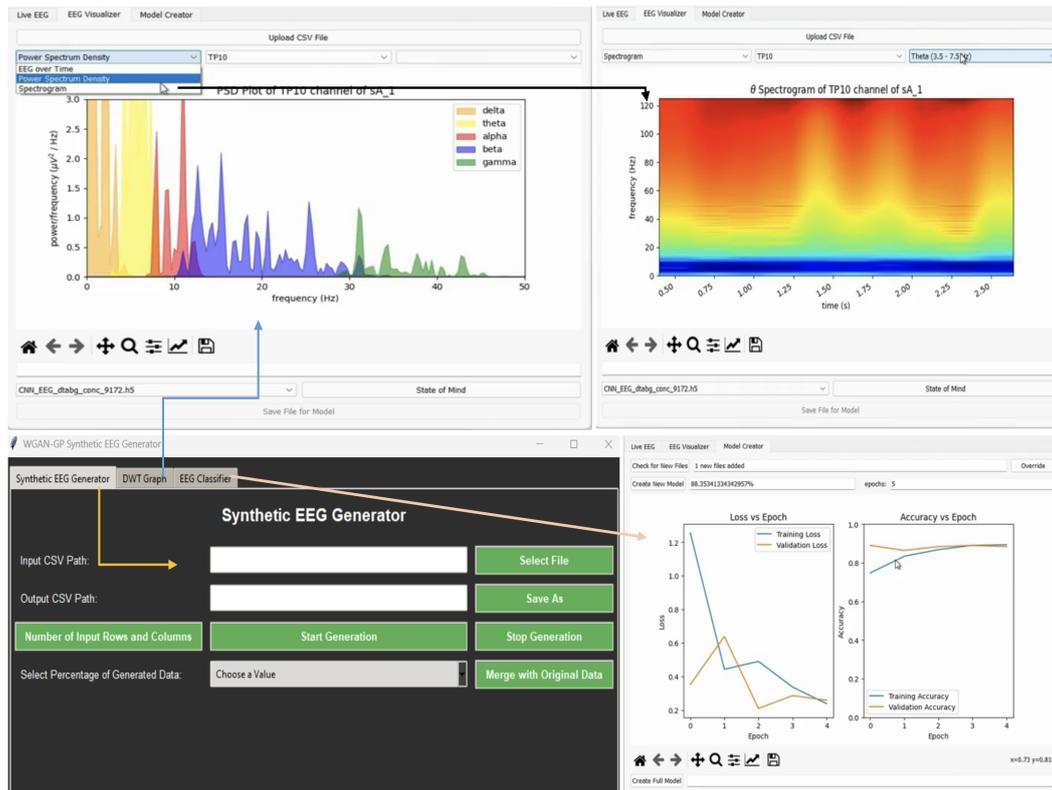
---

**3.7. Support Vector Machines and Random Forests for ECG Classification**

To classify the ECG dataset, we employed a linear SVM and an RF classifier with 100 decision trees and a random state set to 150 in order to ensure consistent results across runs. Both classifiers were trained on 500 individual records evenly distributed between normal and abnormal categories. The data were split into training and testing sets, with 70% allocated to training and 30% to testing. The classifiers were then used to classify unseen real data, synthetic data, and a combination of real and synthetic data in order to evaluate the impact on classification accuracy after using WGAN-GP.

**4. Results****4.1. EEG Results**

To enhance user-friendliness in synthetic EEG generation, an interface was developed using 'tkinter', enabling users to configure input and output paths without modifying the Python code, as shown in Figure 5. The interface allows users to merge real and synthetic data in varying proportions, and it includes tabs for Synthetic EEG Generation, DWT plotting, and CNN classification. Outputs from GAN and WGAN-GP models are saved as CSV files containing synthetic data for channels TP9, AF7, AF8, and TP10. These files match the original data in both data points and timestamps, ensuring seamless integration with real data for classification using the CNN classifier from [31], with results recorded per subject. The current interface is exclusively designed for synthetic EEG wave generation, while synthetic ECG generation is currently handled via a Python script. Future work aims to integrate synthetic ECG generation into the interface.



**Figure 5.** Interface of the synthetic EEG generator, visualization, and CNN classification.

Figures 6 and 7 collectively illustrate the effectiveness of the WGAN-GP model in accurately replicating real EEG patterns in both temporal and frequency domains, thereby validating the model's ability to generate realistic synthetic EEG data. In Figure 6, we observe the EEG waveforms from the TP9 channel for Subject A, presented for both concentration and relaxation states. The synthetic data, which extend seamlessly from the real EEG signals, demonstrate continuity in amplitude and oscillatory patterns, indicating that the model can produce consistent, biologically plausible waveforms. This continuity suggests that the synthetic data may be appended to real EEG data without introducing detectable discontinuities, which is essential for applications requiring prolonged EEG sequences. Figure 7 further supports the validity of the synthetic data by presenting the Power Spectral Density (PSD) of the TP9 channel for the same subject and states. In the concentration state, the PSD of the synthetic data captures prominent gamma wave activity, reflecting the heightened cognitive processing typically associated with concentration. This match in gamma power indicates that the synthetic EEG successfully mirrors the spectral characteristics of the real concentration-state EEG. Similarly, in the relaxation state, the synthetic data show minimal gamma and beta activity, consistent with the reduced cognitive and wakeful engagement expected in a relaxed state. This spectral match reinforces the evidence that the model accurately distinguishes between cognitive states, not only in waveform morphology but also in underlying spectral features.

Together, Figures 6 and 7 illustrate that the WGAN-GP model produces synthetic EEG data that closely align with the real data across both waveform and spectral dimensions, substantiating the model's potential as a tool for generating realistic, state-specific EEG patterns. This high degree of fidelity in both the temporal and spectral domains demonstrates that the synthetic output of the model effectively simulates real EEG data, making it a valuable resource for research and applications requiring synthetic EEG signals. Table 2 presents the number of data points recorded for each EEG channel (TP9, AF7, AF8, and TP10) across the mental states (concentration and relaxation) for real and corresponding synthetic EEG waves for both GAN and WGAN-GP. The data show consistent values across

all channels and subjects for both real and synthetic datasets. This consistency indicates that both models effectively generated synthetic data that matched the structure of the real data, making them suitable for further analysis.

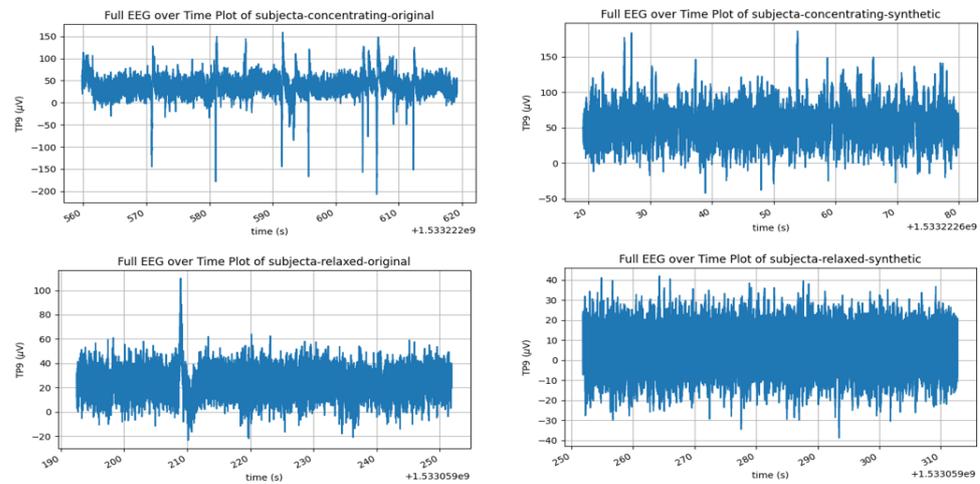


Figure 6. EEG plot of TP9 channel for Subject A in concentration and relaxation states using WGAN-GP.

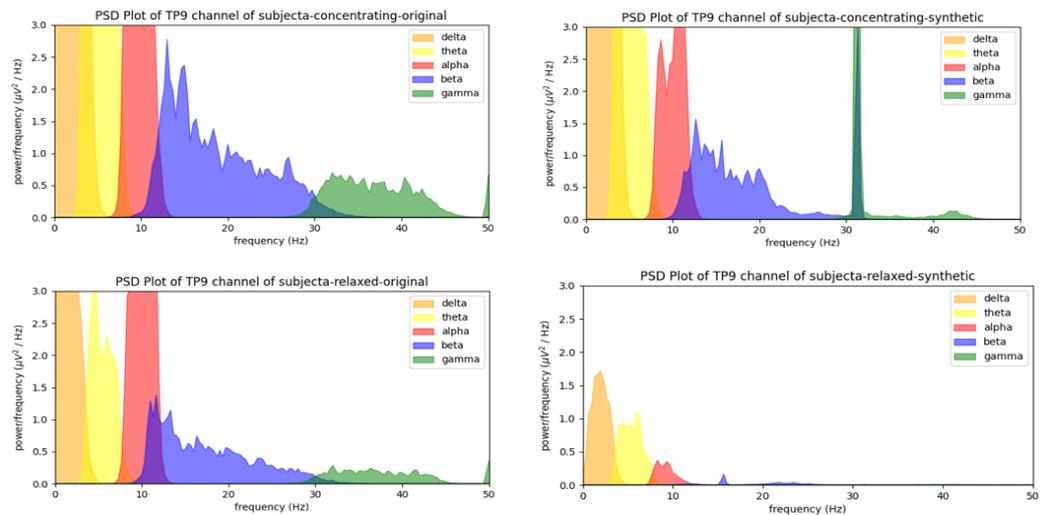


Figure 7. PSD plot of TP9 channel for Subject A in EEG concentration and relaxation states.

The evaluation criteria for model accuracy in Table 3 are based on comparing classification performance across different models when trained on a real dataset and tested on unseen real, synthetic, and combined datasets. Specifically, the table highlights the accuracy achieved using each classifier type—including SVM, RF, and CNN—when applied to the respective datasets. For Bird et al. [30], accuracy values are reported from their baseline models using the same dataset, with the classifiers trained on the real dataset. The reported accuracy reflects the percentage of correct classifications made using each model on each dataset type, indicating each model’s effectiveness in correctly identifying the target classes. The performance of our models (GAN and WGAN-GP) is also presented for a direct comparison, where the same dataset as that of Bird et al. is used. The CNN classifier trained on real data was employed, with WGAN-GP’s CNN classifier demonstrating notably higher accuracy than the Bird et al. [30] baseline models across real, synthetic, and combined data. This comparison serves to assess the generalization capability and robustness of each model, especially WGAN-GP’s effectiveness in generating data that closely align with real data patterns.

The WGAN-GP model achieves the highest accuracy, with 98.42% for synthetic data and 98.45% for combined data, demonstrating its ability to generate high-quality synthetic EEG data that enhanced classification performance from 92% using only real data. In contrast, the basic GAN model achieved accuracy of 94.11%, surpassing the Bird et al. [30] SVM classification accuracy of 93.71%, but falling short of their random forest (RF) classification, which reached 96.69% for combined data. While GPT-2 provides a creative approach to generating synthetic data, its lower accuracy compared to WGAN-GP suggests that it struggles to capture the nuances of EEG signals. Our findings highlight the effectiveness of WGAN-GP in addressing data scarcity, improving classification accuracy from 92% to 98.45%. Overall, WGAN-GP emerges as a superior option for applications requiring the precise classification of complex biological signals.

Tables 4 and 5 analyze the classification performance with varying percentages of synthetic data, providing important insights into optimizing model accuracy. For the WGAN-GP model, integrating synthetic data significantly improves performance, with average accuracies for concentration and relaxation consistently increasing as the proportion of synthetic data rises from 25% to 50%. At 50%, the model achieves an average accuracy of 98.48%, representing a peak in performance. However, introducing 75% synthetic data results in a slight decline before improving again to 100%, suggesting diminishing returns beyond the 50% threshold.

**Table 2.** Number of datapoints for each channel (TP9, AF7, AF8, and TP10) of real and corresponding generated synthetic waves for each mental state table.

Mental State	Datapoints per Channel	Subject A	Subject B	Subject C	Subject D
Concentration	Real	15192	11364	15204	11364
	Synthetic	15192	11364	15204	11364
Relaxation	Real	15204	15204	15204	15204
	Synthetic	15204	15204	15204	15204

**Table 3.** Comparison of model accuracy(%) between our approach and Bird et al. [30] in classifying real, synthetic, and combined datasets.

Model	Classifier	Real Data	Synthetic Data	Real + Synthetic Data
Bird et al. [30]: GPT-2	Support vector machine (SVM)	90.84	66.88	93.71
Bird et al. [30]: GPT-2	Random forest (RF)	88.14	70.71	96.69
Our model: GAN	Convolutional neural network (CNN)	92	85.78	94.11
Our model: WGAN-GP	Convolutional neural network (CNN)	92	98.42	98.45

**Table 4.** Average classification accuracy(%) for mental states—concentration and relaxation—for varying synthetic data percentages added to the original dataset for a WGAN-GP model.

Original Data + X% Synthetic Data	Concentration	Relaxation	Average
25%	99.27	97.28	98.28
50%	99.59	97.36	98.48
75%	99.34	97.26	98.3
100%	99.51	97.39	98.45

In contrast, the GAN model exhibits a decline in accuracy as the proportion of synthetic data increases, particularly when transitioning from 25% to 50%. The optimal performance of the GAN was observed at 25% synthetic data, indicating a notable disparity between the two models. This suggests that the synthetic data generated via the GAN may not be of the same quality as that produced via WGAN-GP.

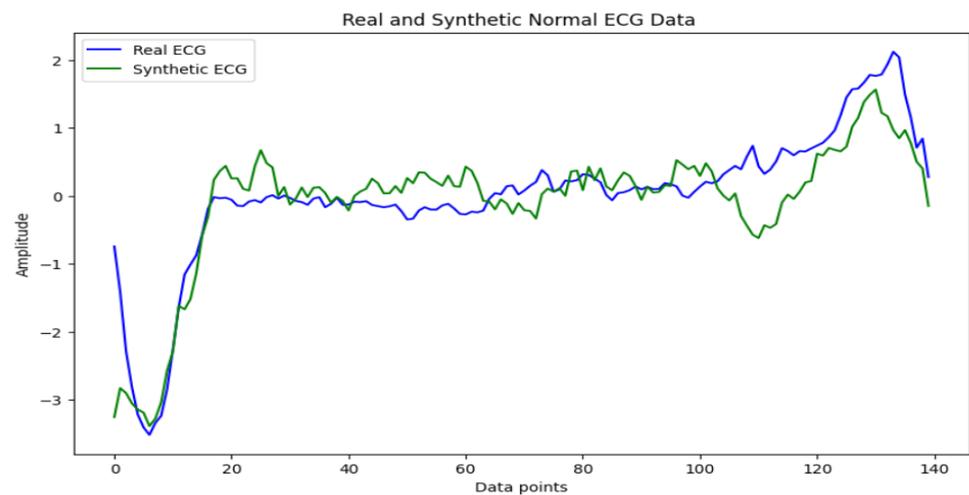
Overall, our findings indicate that incorporating up to 50% synthetic data enhances classification accuracy for WGAN-GP, establishing it as an effective strategy for improving EEG data analysis.

**Table 5.** Average classification accuracy(%) for mental states—concentration and relaxation—for varying synthetic data percentages added to the original dataset for a GAN model.

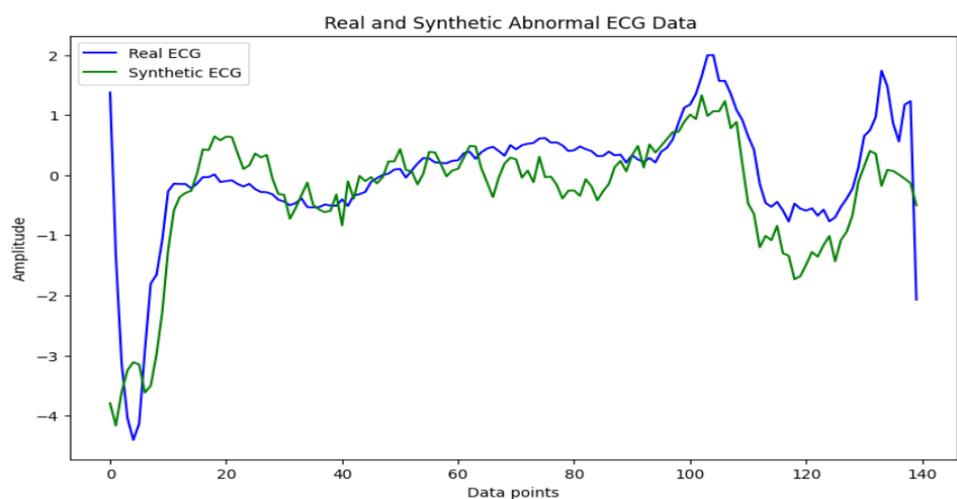
Original data + X% Synthetic Data	Concentration	Relaxation	Average
25%	95.39	100	97.69
50%	92.44	100	96.22
75%	90.77	100	95.39
100%	88.22	100	94.11

#### 4.2. ECG Results

Figures 8 and 9 display the normal and abnormal ECG samples of both real and synthetic data generated via the WGAN-GP model, illustrating the similarity between real and generated samples.



**Figure 8.** Real and synthetic normal ECG samples.



**Figure 9.** Real and synthetic abnormal ECG samples.

Table 6 presents the classification performance of ECG datasets (real, synthetic, and a combination of both) using SVM and RF classifiers. The datasets include 1200 real ECG samples, each containing 140 data points, and 1200 corresponding synthetic samples

generated via the WGAN-GP model. The combined dataset, comprising both real and synthetic data, totaled 2400 samples.

**Table 6.** ECG classification of real, synthetic, and combined datasets using SVM and RF classifiers.

Model	Classifier	Real Data	Synthetic Data	Real + Synthetic Data
WGAN-GP	Support vector machine (SVM)	98	95.8	97
WGAN-GP	Random factor (RF)	97	98.57	98.40

For the SVM classifier, the real dataset achieved the highest classification accuracy at 98%, while the synthetic dataset alone achieved 95.8%. Combining real and synthetic data resulted in an accuracy of 97%, suggesting that real data alone may provide more informative features for SVM. However, combining synthetic data still yielded robust performance.

In contrast, the RF classifier performed best with the synthetic dataset, achieving an accuracy of 98.57%, which outperformed the real dataset's accuracy of 97%. When the combined dataset was used, RF maintained a high accuracy of 98.40%. This indicates that RF benefits from the diversity provided via synthetic data and can effectively integrate both sources for robust classification.

Overall, the results suggest that, while SVM performs optimally with real data, RF demonstrates superior performance with synthetic data and remains highly effective when real and synthetic datasets are combined. Although the accuracy improvements from incorporating synthetic data for ECG are not as pronounced as those for EEG, the 1–2% increase still highlights the value of synthetic data for enhancing classification performance.

#### 4.3. Statistical Significance of the Results

We ran statistical significance tests to assess the performance differences between models across various datasets (real, synthetic, and real + synthetic). The method employed was the Wilcoxon signed-rank test. The Wilcoxon signed-rank test, a non-parametric test, was used for pairwise comparisons. It is particularly effective for comparing matched data when the assumption of normality cannot be guaranteed. The test identifies whether differences between paired observations are symmetric to zero, providing  $p$ -values to evaluate statistical significance.

The results of the statistical significance tests are summarized in Table 7 and further visualized in Figures 10 and 11.

##### 1. Real data comparisons:

- GPT-2 + SVM vs. GPT-2 + RF: the difference in accuracies (90.84% vs. 88.14%) was not statistically significant ( $p > 0.05$ ), suggesting similar performance for these two models on real data.
- GPT-2 + SVM vs. WGAN-GP + CNN: WGAN-GP + CNN significantly outperformed GPT-2 + SVM (92% vs. 90.84%,  $p < 0.05$ ).
- GPT-2 + RF vs. WGAN-GP + CNN: the WGAN-GP + CNN model significantly surpassed GPT-2 + RF (92% vs. 88.14%,  $p < 0.01$ ), indicating its superior reliability.

##### 2. Synthetic data comparisons:

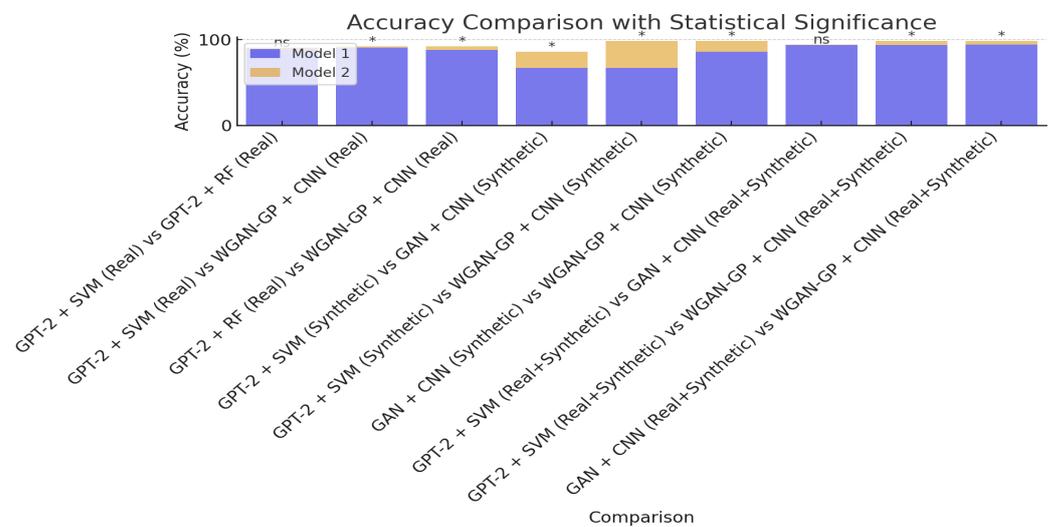
- GPT-2 + SVM vs. GAN + CNN: GAN + CNN significantly outperformed GPT-2 + SVM (85.78% vs. 66.88%,  $p < 0.01$ ).
- GPT-2 + SVM vs. WGAN-GP + CNN: WGAN-GP + CNN achieved markedly higher accuracy (98.42% vs. 66.88%,  $p < 0.01$ ).
- GAN + CNN vs. WGAN-GP + CNN: WGAN-GP + CNN was significantly better than GAN + CNN (98.42% vs. 85.78%,  $p < 0.01$ ), reinforcing the robustness of the WGAN-GP model for synthetic data generation.

##### 3. Real + synthetic data comparisons:

- GPT-2 + SVM vs. GAN + CNN: the performance difference between GPT-2 + SVM and GAN + CNN was not statistically significant (93.71% vs. 94.11%,  $p > 0.05$ ).
- GPT-2 + SVM vs. WGAN-GP + CNN: WGAN-GP + CNN significantly outperformed GPT-2 + SVM (98.45% vs. 93.71%,  $p < 0.01$ ).
- GAN + CNN vs. WGAN-GP + CNN: WGAN-GP + CNN significantly outperformed GAN + CNN (98.45% vs. 94.11%,  $p < 0.01$ ).

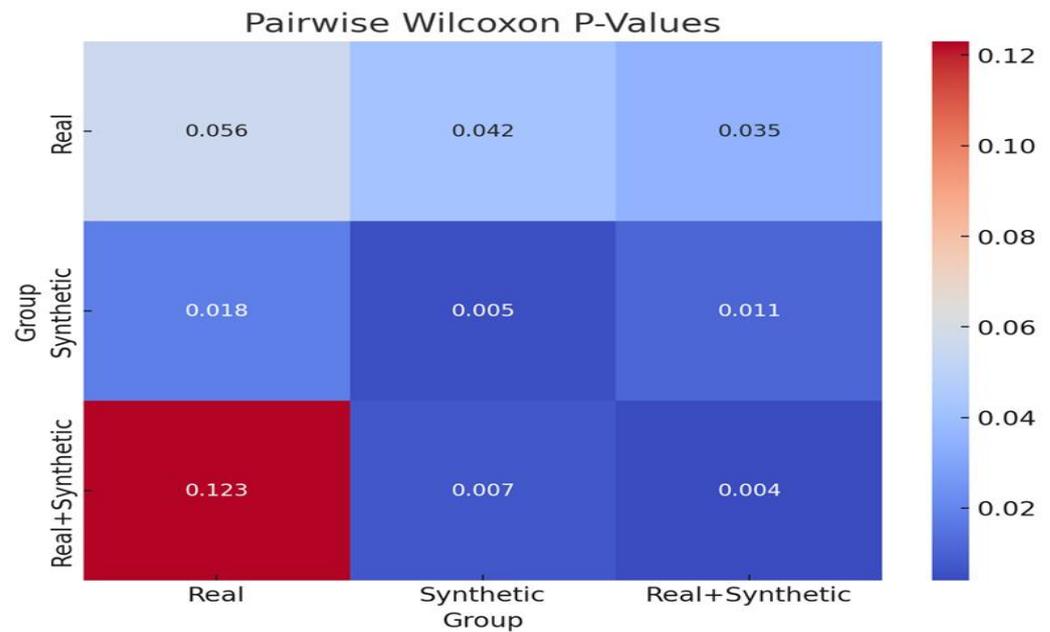
**Table 7.** Comparison of model performances based on accuracy and Wilcoxon  $p$ -value significance testing, where “S” means significant, and “NS” means not significant in the *Significance* column.

Comparison	Accuracy of Model 1 (%)	Accuracy of Model 2 (%)	Wilcoxon $p$ -Value	Significance
GPT-2 + SVM (Real) vs. GPT-2 + RF (Real)	90.84	88.14	0.056	NS
GPT-2 + SVM (Real) vs. WGAN-GP + CNN (Real)	90.84	92.0	0.042	S
GPT-2 + RF (Real) vs. WGAN-GP + CNN (Real)	88.14	92.0	0.035	S
GPT-2 + SVM (Synthetic) vs. GAN + CNN (Synthetic)	66.88	85.78	0.018	S
GPT-2 + SVM (Synthetic) vs. WGAN-GP + CNN (Synthetic)	66.88	98.42	0.005	S
GAN + CNN (Synthetic) vs. WGAN-GP + CNN (Synthetic)	85.78	98.42	0.011	S
GPT-2 + SVM (Real+Synthetic) vs. GAN + CNN (Real+Synthetic)	93.71	94.11	0.123	NS
GPT-2 + SVM (Real+Synthetic) vs. WGAN-GP + CNN (Real+Synthetic)	93.71	98.45	0.007	S
GAN + CNN (Real+Synthetic) vs. WGAN-GP + CNN (Real+Synthetic)	94.11	98.45	0.004	S



**Figure 10.** Bar chart of model accuracies with significance annotations. The label “ns” stands for no statistical significance and the label “\*” presents comparisons with statistical significance.

The bar chart shown in Figure 10 provides a visual comparison of model accuracies across the datasets, with statistical significance denoted as \* ( $p < 0.05$ ) or \*\* ( $p < 0.01$ ). It highlights the consistently superior performance of WGAN-GP + CNN, particularly for synthetic and real + synthetic data. The heatmap in Figure 11 illustrates the pairwise  $p$ -values from the Wilcoxon signed-rank test, where darker shades represent lower  $p$ -values, indicating stronger statistical significance. The clear contrast between WGAN-GP + CNN and other models reinforces its performance advantages. These results demonstrate the effectiveness of WGAN-GP + CNN, particularly for scenarios involving synthetic and augmented datasets. Its superior accuracy and statistical significance highlight its potential for applications requiring robust and reliable classification models.



**Figure 11.** Heatmap of pairwise statistical significance.

## 5. Conclusions and Future Work

This research has introduced a novel approach using the WGAN-GP model to generate synthetic EEG waveforms corresponding to concentrated and relaxed mental states. These waveforms can be effectively utilized in BCI applications across various domains requiring human–machine interaction. The synthetic data generated via WGAN-GP after the EEG signals are pre-processed significantly enhance machine learning model training by increasing the dataset size, leading to improved generalization and performance. Specifically, the classification accuracy when only the real dataset was used reached 92%, but this increased to 98.45% when combined with synthetic data generated via WGAN-GP. This performance surpasses state-of-the-art models, which have reported SVM accuracy at 93.71% and RF accuracy at 96.69% on the same datasets. Meanwhile, the original GAN model achieved an accuracy of 94.11% with a mix of real and synthetic data, underscoring the superior quality of synthetic data generated via WGAN-GP to augment real datasets. Notably, adding 25% synthetic data generated via WGAN-GP was sufficient to improve accuracy, while 50% proved optimal across mental states, achieving the highest classification accuracy with the proposed CNN classifier architecture. These findings highlight the WGAN-GP model’s capability to generate high-quality synthetic EEG data, and these results illustrate how combining real and synthetic data enhances overall classification accuracy. Additionally, a user interface was developed to improve the usability of the generator model, enabling the synthesis of EEG waveforms corresponding to both concentration and relaxation states, beyond data visualization and classification. For ECG classification, the SVM model performed best with real ECG data, achieving an accuracy of 98%, while synthetic data alone reached 95.8%. In contrast, the RF classifier excelled with synthetic data, achieving an accuracy of 98.75%. When real and synthetic data were combined, the RF model maintained a high accuracy of 98.40%, an increase from 97% when real data alone were used. This demonstrates the robustness and quality of the synthetic data generated via the WGAN-GP model. Additionally, statistical significance was determined using the Wilcoxon signed-rank test, further emphasizing the potential of WGAN-GP for applications requiring a robust and reliable classification model.

### *Future Work*

Future work for the WGAN-GP model will include exploring its applicability to additional biological signals, such as EMG, in order to further evaluate its generalizability

and effectiveness. Moreover, we plan to enhance the interface in order to support direct ECG signal generation, as this process is currently performed via a Python script. Expanding the model in these directions could provide valuable datasets for advancing and refining machine learning algorithms. These advancements can significantly enhance human–technology interaction in applications such as assistive technologies, mental health monitoring, and cognitive load assessment, ultimately improving engagement and performance in interactive tasks.

**Author Contributions:** Methodology, A.V. and D.R.F.; Software, A.V.; Investigation, A.V.; Writing—original draft, A.V.; Writing—review & editing, D.R.F.; Supervision, D.R.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

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

## References

- Hoffmann, J.; Mahmood, S.; Fogou, P.S.; George, N.; Raha, S.; Safi, S.; Schmailzl, K.J.; Brandalero, M.; Hubner, M. A Survey on Machine Learning Approaches to ECG Processing. In Proceedings of the Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), Poznan, Poland, 23–25 September 2020.
- Benhamida, A.; Kozlovsky, M. Human ECG data collection, digitalization, streaming and storing. In Proceedings of the 18th World Symposium on Applied Machine Intelligence and Informatics (SAMII), Herlany, Slovakia, 23–25 January 2020.
- Salehi, P.; Chalechale, A.; Taghizadeh, M. Generative Adversarial Networks (GANs): An Overview of Theoretical Model, Evaluation Metrics, and Recent Developments. *arXiv* **2020**, arXiv:2005.13178.
- Abdelfattah, S.M.; Abdelrahman, G.M.; Wang, M. Augmenting the size of EEG datasets using generative Adversarial Networks. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018.
- Aznan, N.K.N.; Atapour-Abarghouei, A.; Bonner, S.; Connolly, J.D.; Al Moubayed, N.; Breckon, T.P. Simulating Brain Signals: Creating Synthetic EEG Data via Neural-Based Generative Models for Improved SSVEP Classification. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019.
- Galván, C.M.; Spies, R.D.; Milone, D.H.; Peterson, V. Neurophysiologically meaningful motor imagery EEG simulation with applications to data augmentation. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2024**, *32*, 2346–2355. [[CrossRef](#)] [[PubMed](#)]
- Chen, S.-Y.; Chang, C.-M.; Chiang, K.-J.; Wei, C.-S. SSVEP-DAN: Cross-Domain Data Alignment for SSVEP-based Brain-Computer Interfaces. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2024**, *32*, 2027–2037. [[CrossRef](#)] [[PubMed](#)]
- Chaurasia, A.K.; Fallahi, M.; Strufe, T.; Terhörst, P.; Cabarcos, P.A. NeuroIDBench: An open-source benchmark framework for the standardization of methodology in brainwave-based authentication research. *J. Inf. Secur. Appl.* **2024**, *85*, 103832. [[CrossRef](#)]
- Zhang, S.; Sun, L.; Mao, X.; Hu, C.; Liu, P. Review on EEG-Based Authentication Technology. *Comput. Intell. Neurosci.* **2021**, *2021*, 5229576. [[CrossRef](#)]
- Delaney, A.M.; Brophy, E.; Ward, T.E. Synthesis of Realistic ECG using Generative Adversarial Networks. *arXiv* **2019**, arXiv:1909.09150.
- Adib, E.; Afghah, F.; Prevost, J.J. Synthetic ECG Signal Generation Using Generative Neural Networks. *arXiv* **2021**, arXiv:2112.03268.
- Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *arXiv* **2014**, arXiv:1406.2661. [[CrossRef](#)]
- Fahimi, F.; Zhang, Z.; Goh, W.B.; Ang, K.K.; Guan, C. Towards EEG Generation Using GANs for BCI Applications. In Proceedings of the International Conference on Biomedical and Health Informatics (BHI), Chicago, IL, USA, 19–22 May 2019.
- Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. *arXiv* **2014**, arXiv:1411.1784.
- Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv* **2015**, arXiv:1511.06434.
- Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein Gan. *arXiv* **2017**, arXiv:1701.07875.
- Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A. Improved Training of Wasserstein GANs. *arXiv* **2017**, arXiv:1704.00028.
- Habashi, A.G.; Azab, A.M.; Eldawlatly, S.; Aly, G.M. Generative adversarial networks in EEG analysis: An overview. *J. Neuroeng. Rehabil.* **2023**, *20*, 40. [[CrossRef](#)] [[PubMed](#)]

19. Cheng, X.; Huang, K.; Zou, Y.; Ma, S. SleepEGAN: A GAN-enhanced ensemble deep learning model for imbalanced classification of sleep stages. *Biomed. Signal Process. Control.* **2024**, *92*, 106020. [CrossRef]
20. Shin, H.-C.; Tenenholtz, N.A.; Rogers, J.K.; Schwarz, C.G.; Senjem, M.L.; Gunter, J.L.; Andriole, K.; Michalski, M. Medical Image Synthesis for Data Augmentation and Anonymization using Generative Adversarial Networks. *arXiv* **2018**, arXiv:1807.10225.
21. Hazra, D.; Byun, Y.-C. Synsiggan: Generative adversarial networks for synthetic biomedical signal generation. *Biology* **2020**, *9*, 441. [CrossRef]
22. Salazar, A.; Vergara, L.; Safont, G. Generative adversarial networks and Markov random fields for oversampling very small training sets. *Expert Syst. Appl.* **2021**, *163*, 113819. [CrossRef]
23. Zhao, W.; Ye, L.; Cui, Z. EEG Generation Using Generative Adversarial Networks (GANs) [PDF]. Available online: <https://warrenzha.github.io/assets/pdf/GAN-EEG-Generation.pdf> (accessed on 17 August 2024).
24. Kumar, J.S.; Bhuvaneshwari, P. Analysis of Electroencephalography (EEG) Signals and Its Categorization—A Study. *Procedia Eng.* **2012**, *38*, 2525–2536. [CrossRef]
25. Schiliro, F.; Moustafa, N.; Beheshti, A. Cognitive Privacy: AI-enabled Privacy using EEG Signals in the Internet of Things. In Proceedings of the 6th International Conference on Dependability in Sensor, Cloud and Big Data Systems and Application (DependSys), Nadi, Fiji, 14–16 December 2020.
26. Popescu, A.B.; Taca, I.A.; Nita, C.I.; Vizitiu, A.; Demeter, R.; Suci, C.; Itu, L.M. Privacy Preserving Classification of EEG Data Using Machine Learning and Homomorphic Encryption. *Appl. Sci.* **2021**, *11*, 7360. [CrossRef]
27. Goyal, M.; Mahmoud, Q.H. A Systematic Review of Synthetic Data Generation Techniques Using Generative AI. *Electronics* **2024**, *13*, 3509. [CrossRef]
28. Piacentino, E.; Guarner, A.; Angulo, C. Generating Synthetic ECGs Using GANs for Anonymizing Healthcare Data. *Electronics* **2021**, *10*, 389. [CrossRef]
29. Xu, J.; Wang, R.; Shang, S.; Chen, A.; Winterbottom, L.; Hsu, T.-L.; Chen, W.; Ahmed, K.; La Rotta, P.L.; Zhu, X.; et al. ChatEMG: Synthetic Data Generation to Control a Robotic Hand Orthosis for Stroke. *arXiv* **2024**, arXiv:2406.12123.
30. Bird, J.J.; Pritchard, M.; Fratini, A.; Ekart, A.; Faria, D.R. nSynthetic Biological Signals Machine-Generated by GPT-2 Improve the Classification of EEG and EMG Through Data Augmentation. *IEEE Robot. Autom. Lett.* **2021**, *6*, 3498–3504. [CrossRef]
31. Manoharan, G.; Faria, D.R. Enhanced Mental State Classification Using EEG-Based Brain-Computer Interface Through Deep Learning. In *Intelligent Systems and Applications. IntelliSys 2024; Lecture Notes in Networks and Systems*; Arai, K., Ed.; Springer, Cham, Switzerland, 2024; Volume 1067.
32. Venkatesan, C.; Karthigaikumar, P.; Paul, A.; Satheeskumaran, S.; Kumar, R. ECG Signal Preprocessing and SVM Classifier-Based Abnormality Detection in Remote Healthcare Applications. *IEEE Access* **2018**, *6*, 9767–9773. [CrossRef]
33. Zhang, Y.; Wei, S.; Zhang, L.; Liu, C. Comparing the Performance of Random Forest, SVM and Their Variants for ECG Quality Assessment Combined with Nonlinear Features. *J. Med. Biol. Eng.* **2018**, *39*, 381–392. [CrossRef]
34. Bird, J.T.; Manso, L.; Ribeiro, E.P.; Ekárt, A.; Faria, D.R. A Study on Mental State Classification using EEG-based Brain-Machine Interface. In Proceedings of the International Conference on Intelligent Systems, Madeira, Portugal, 25–27 September 2018.
35. ECG Dataset. Available online: <https://www.kaggle.com/datasets/devavratatripathy/ecg-dataset> (accessed on 5 May 2024).
36. Chiu, T.Y.; Leonard, T.; Tsui, K.W. The matrix-logarithmic covariance model. *J. Am. Stat. Assoc.* **1996**, *91*, 198–210. [CrossRef]
37. Amin, H.U.; Malik, A.S.; Ahmad, R.F.; Badruddin, N.; Kamel, N.; Hussain, M.; Chooi, W.-T. Feature extraction and classification for EEG signals using wavelet transform and machine learning techniques. *Australas. Phys. Eng. Sci. Med.* **2015**, *38*, 139–149. [CrossRef]
38. Fekih, R.T.; Ouni, R. Electrocardiogram analysis using discrete wavelet transform for anomalies detection. *Comput. Sci.* **2023**, *4*, 348.
39. Aliyu, I.; Lim, C.G. Selection of optimal wavelet features for epileptic EEG Signal Classification with LSTM. *Neural Comput. Appl.* **2021**, *35*, 1077–1097. [CrossRef]
40. Broll, A.; Goldhacker, M.; Hahnel, S.; Rosentritt, M. Generative deep learning approaches for the design of dental restorations: A narrative review. *J. Dent.* **2024**, *145*, 104988. [CrossRef]
41. Khodja, H.A.; Boudjeniba, O. Application of WGAN-GP in recommendation and questioning the relevance of gan-based approaches. *arXiv* **2022**, arXiv:2204.12527v2.
42. Inverse Discrete Wavelet Transform (IDWT)—PyWavelets Documentation. Available online: <https://pywavelets.readthedocs.io/en/latest/ref/idwt-inverse-discrete-wavelet-transform.html> (accessed on 5 May 2024).

**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.