Stress level Classification using EEG Signal with Deep Learning

Team – 12, Batch – A

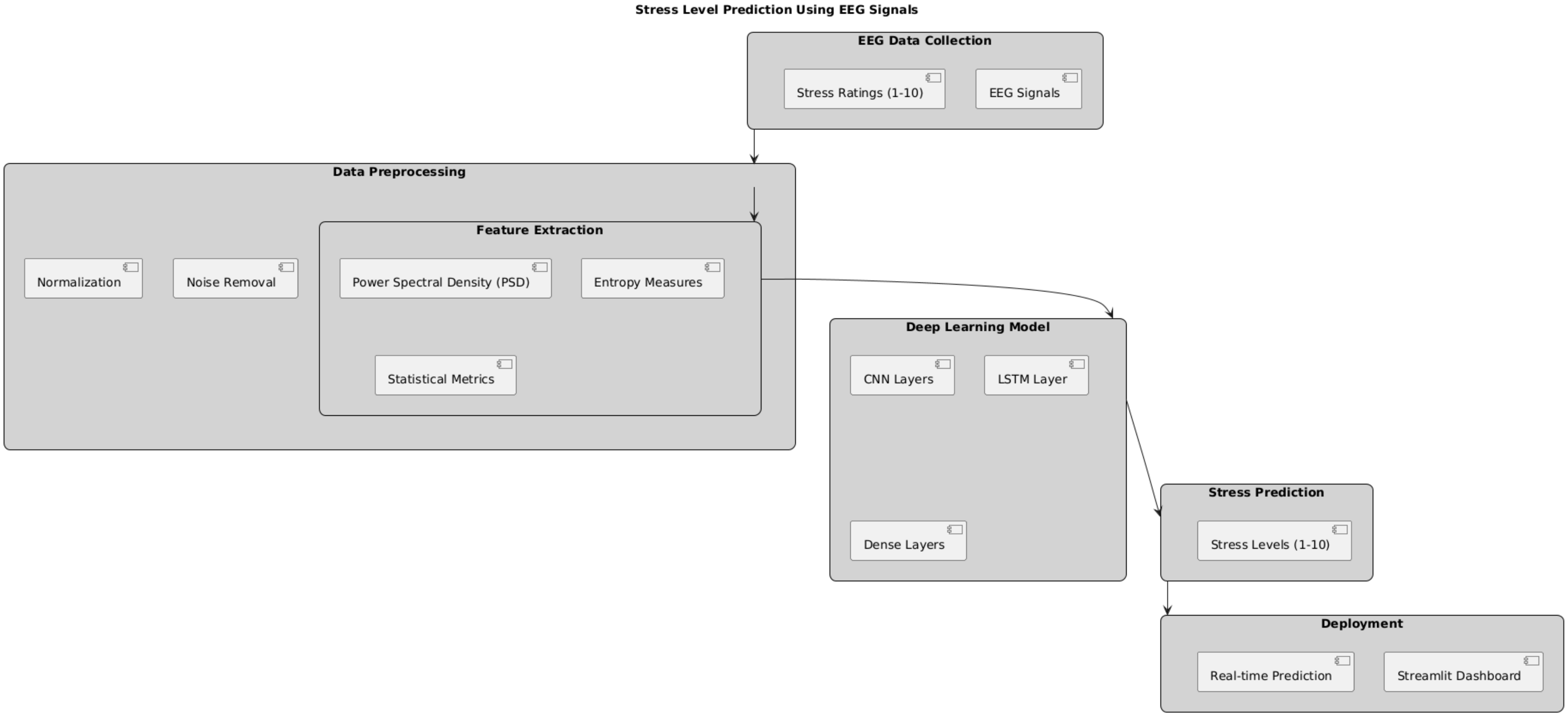
Problem Statement

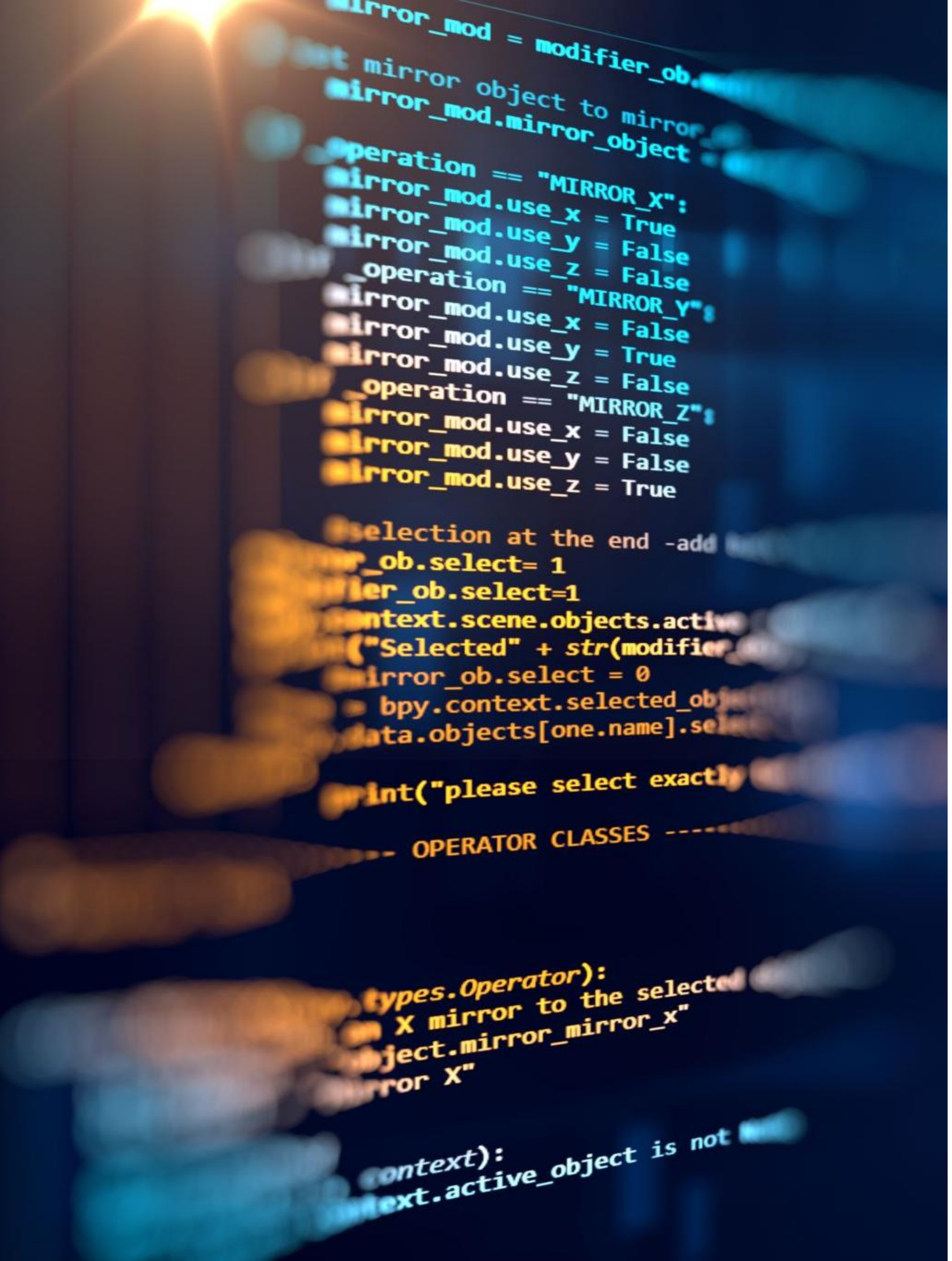
- In today's fast-paced world, stress has become a pervasive issue, impacting mental health and overall well-being. Traditional methods of stress detection, often reliant on self-reporting and basic physiological measurements, lack the precision and real-time responsiveness needed for effective intervention. Furthermore, current EEG-based stress detection techniques are generally limited to classification tasks, which do not capture the continuous nature of stress variations experienced in real life.

Procedure

- This project aims to address these limitations by developing a system capable of predicting stress levels on a continuous scale of 1 to 10 using EEG signals and advanced deep learning techniques.The core deep learning models include Convolutional Neural Networks (CNN) for spatial feature extraction and Long Short-Term Memory (LSTM) networks for temporal analysis of the EEG signals. The challenge lies in accurately processing and interpreting complex EEG data, dealing with noise, and ensuring model robustness across diverse populations. Additionally, the system must provide real-time feedback and personalized stress management advice, making it a comprehensive tool for detection

Block diagram





Dataset used

EEG Dataset for Cognitive Load and Stress Assessment During Task Performance

Link:
https://www.sciencedirect.com/science/article/pii/S2352340921010465

- This **EEG dataset**, sourced from a study on cognitive load and stress, includes recordings from participants performing tasks such as the Stroop Color-Word test, arithmetic problem-solving, and mirror image recognition. EEG signals were captured across multiple trials, with subjects providing stress ratings on a scale of **1-10** after each task.
- It has 32 Channels of EEG signal for each person

Methodology

EEG Signal Collection:

EEG signals are recorded from participants under various conditions to capture a range of stress levels. The raw signals from multiple brainwave channels.

Noise Removal:

Apply adaptive noise filtering techniques to eliminate artifacts and external noise from the raw EEG data, ensuring cleaner and more reliable inputs.

Feature Extraction:

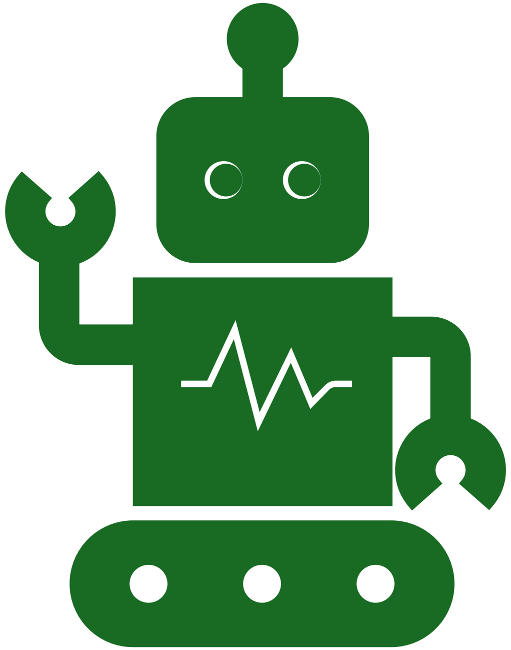
Extract critical features from the EEG signals, such as power spectral density, frequency band power, and nonlinear dynamics. These features are essential for accurately capturing the physiological markers of stress.

- **Deep Learning Model:**

  Develop a regression-based deep learning model like CNN and LSTM to predict stress levels on a continuous scale of 1 to 10..

- **Model Training:**

  Train the model using advanced optimization techniques, after training the Model.



Filtering

Why Filtering is Crucial in EEG Signal Processing

- EEG signals are often contaminated by various types of noise and artifacts, which can distort the underlying brain activity signals. These interferences make it difficult to extract meaningful information from the raw EEG data, especially in applications like stress prediction where precision is critical

Types of filtering that we used

Butterworth filter

Why Chosen: Smooth frequency response, minimal signal distortion.

Order: 4th order filter, balances sharp cut-off with smooth transitions.

Purpose: Removes unwanted frequencies (1-50 Hz band-pass).

Savitzky-Golay Smoothing Filter

Role: Smooths the EEG signal to reduce high-frequency noise.

Parameters:

- Window_size=21: Controls the extent of smoothing.
- polyorder=3:Polynomial order for fitting the signal.

Independent Component Analysis (ICA)

Purpose: Removes physiological artifacts like eye blinks.

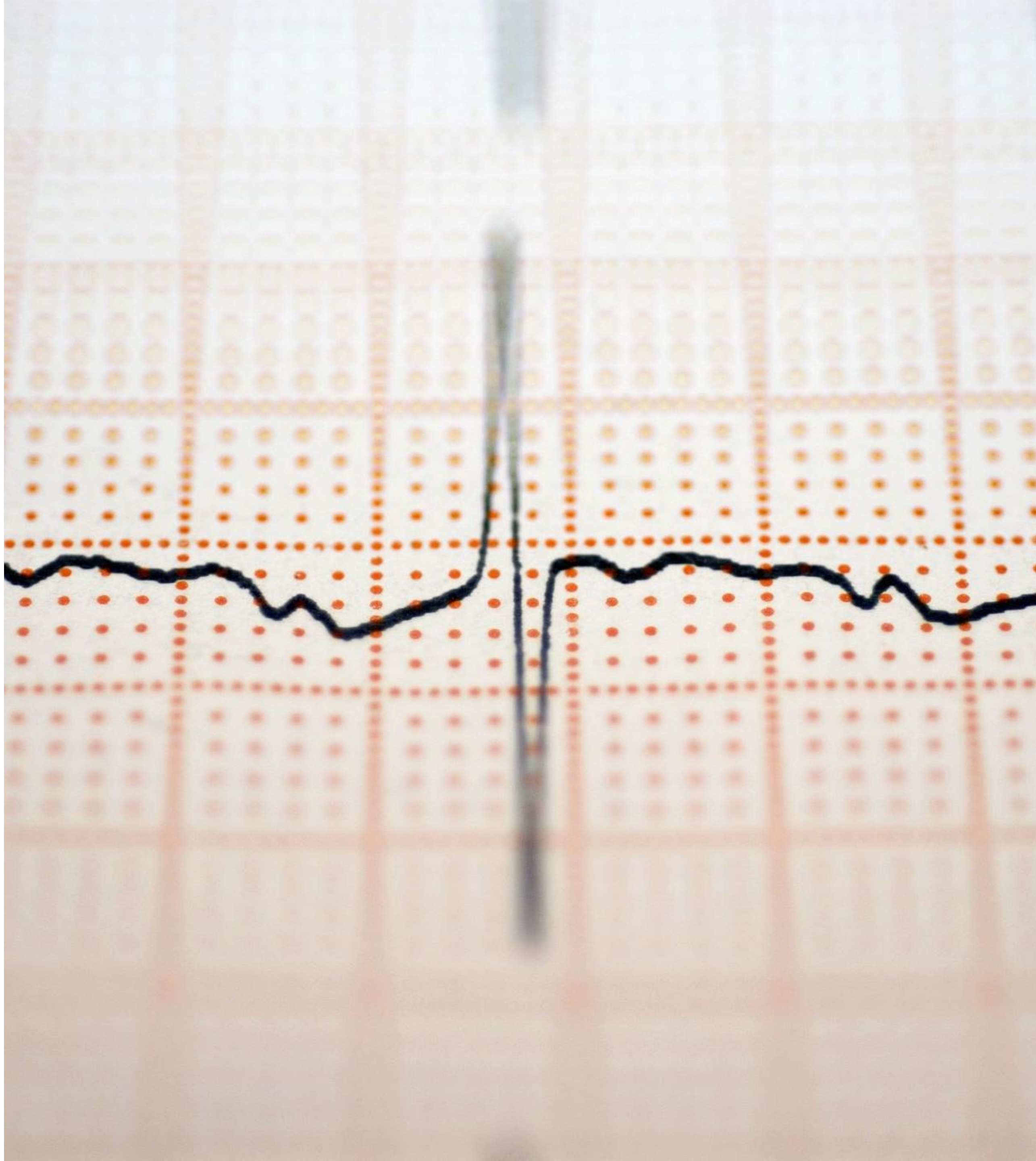
Usage: ICA(n_components=15)

decomposes the signal into components.

Artifact Removal: ICA helps isolate and exclude noise components.

Butterworth bandpass

- **Key Characteristics of the Butterworth Filter**
- **Maximally Flat Frequency Response**: Unlike other filters that may have ripples in the passband or stopband, the Butterworth filter has a smooth frequency response, making it ideal for applications where a flat response is desired.
- **Frequency Range Control**: By adjusting the cutoff frequencies, you can control the range of frequencies that are allowed to pass, which is crucial for isolating specific EEG frequency bands like delta, theta, alpha, etc.



Butterworth Bandpass Filter

Butterworth Filter Transfer Function

- The transfer function H(s) of an n-order Butterworth filter is defined as:
- $H(s) = \frac{1}{\sqrt{1+\left(\frac{s}{w_c}\right)^{2n}}}$

Where:

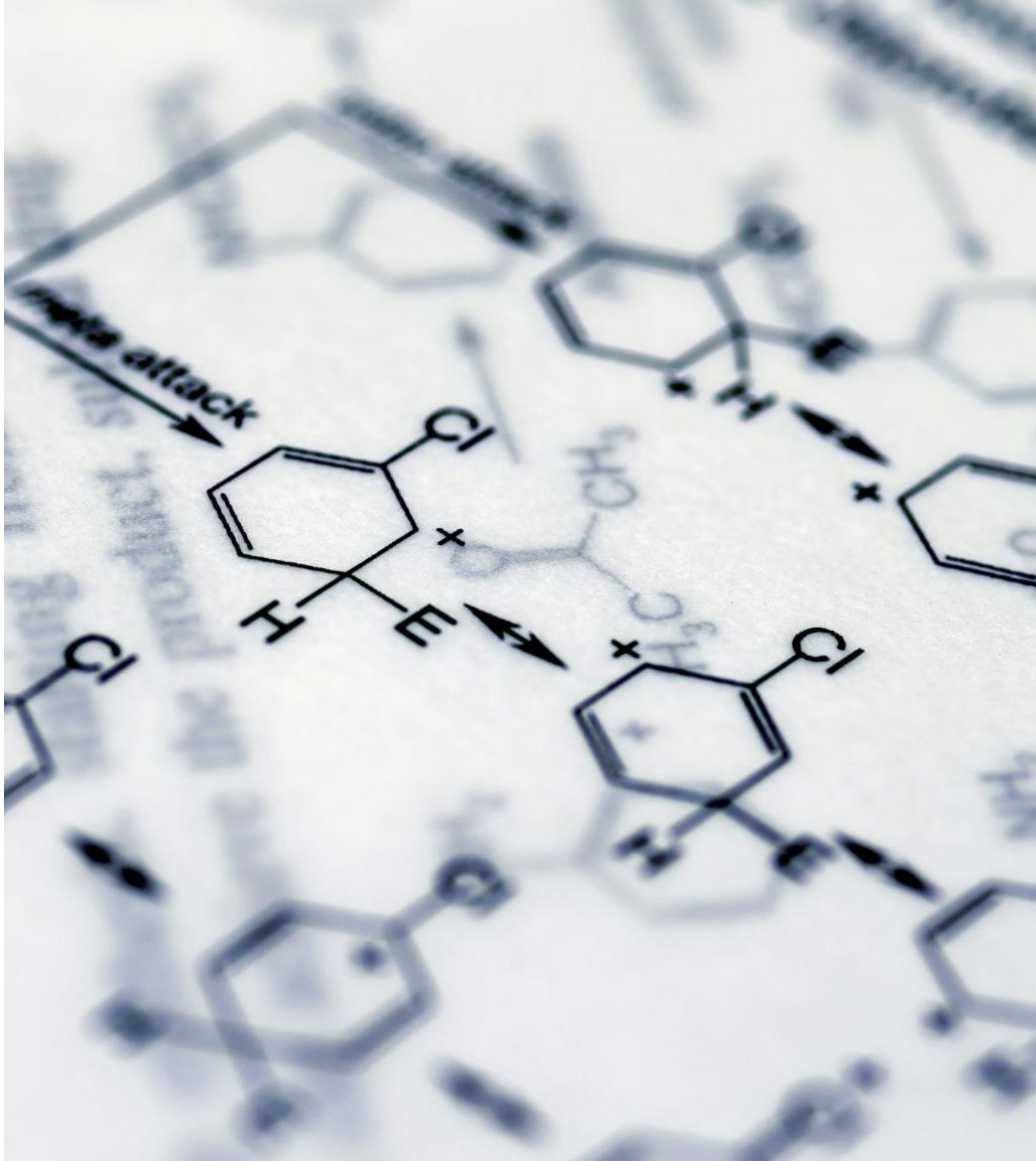
- s is the complex frequency variable, $s = j.\omega$ as the angular frequency and j as the imaginary unit),
- w_c is the cutoff frequency,
- n is the filter order.

For a **bandpass filter**, you design it with two cutoff frequencies:

- **Lower cutoff frequency** w_L : The lower edge of the frequency band.
- **Upper cutoff frequency** w_U : The upper edge of the frequency band.

Savitzky-Golay Smoothing Filter

- **Key Characteristics of the Savitzky-Golay Filter**
- **Key Characteristics of the Savitzky-Golay Filter**
- **Preservation of Features**: Unlike standard moving average filters, which can distort the signal, the Savitzky-Golay filter fits a polynomial to the data points within a specified window, preserving the original shape and features of the data.
- **Adjustable Parameters**: The filter's performance can be adjusted by changing the polynomial order and the window size, allowing for flexible smoothing depending on the signal characteristics.



Savitzky-Golay Smoothing Filter

- **Mathematical Representation**
- Given a time series data set y at points t_i, the smoothed value y(t_i) at point t_i is calculated as follows:

$$y(t_i)=\sum_{j=-p}^{p} c_j y\left(t_{i+j}\right)$$

- Where:
- y(t_i) is the smoothed value at time t_i.
- c_j are the Savitzky-Golay filter coefficients for the polynomial.
- p is the number of points on either side of used for smoothing.
- $y\left(t_{i+j}\right)$ is the original data value at the corresponding time point.

ICA (Indepentedent Component Analysis) Filterning

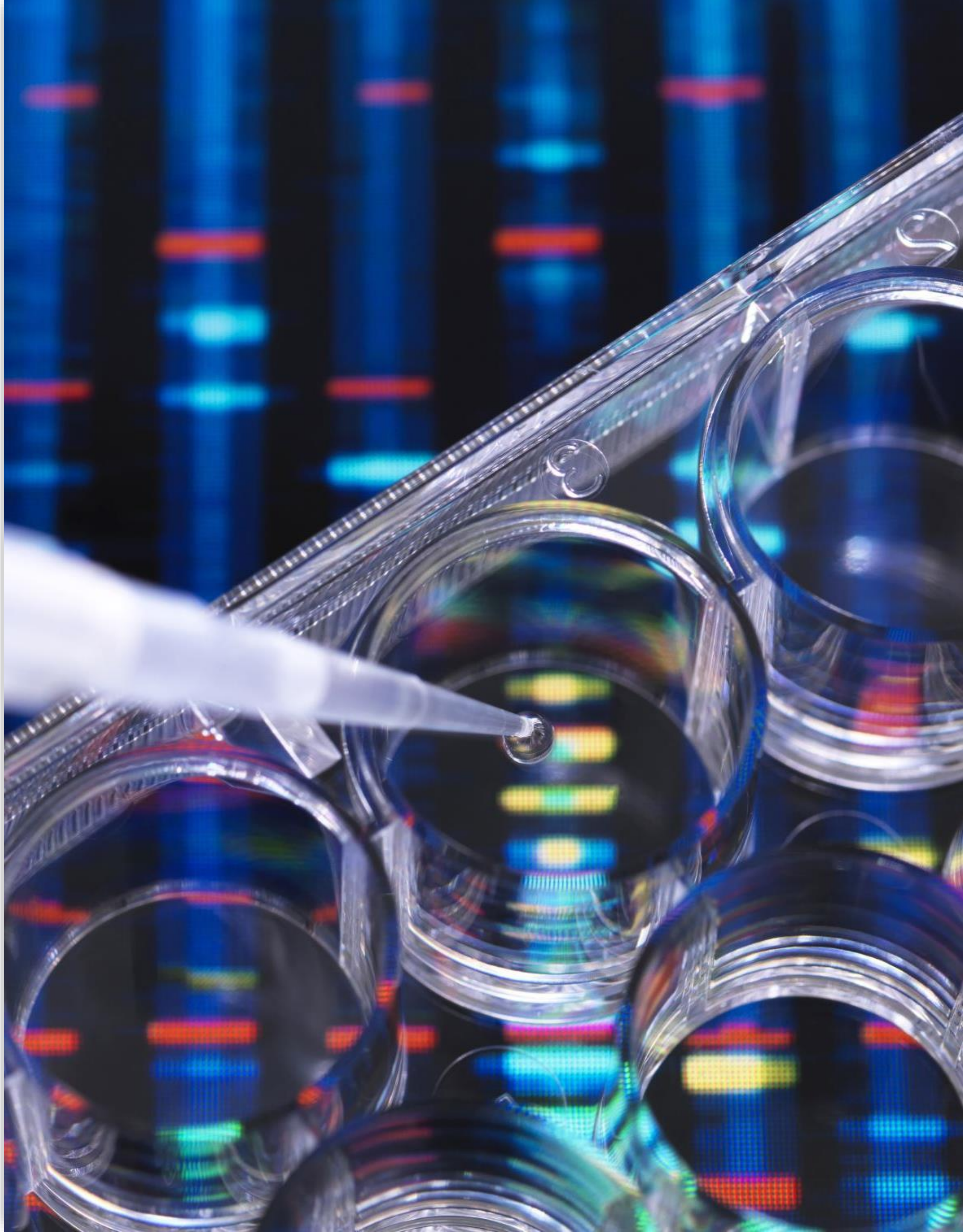
Mathematical Formulation of ICA

Suppose we have a set of observed signals represented as a matrix X:

$$X = AS$$

Where:

- X is the observed signal matrix (e.g., EEG data).
- A is the mixing matrix, which describes how the sources are mixed to form the observed signals.
- S is the matrix of source signals (the independent components we want to recover).



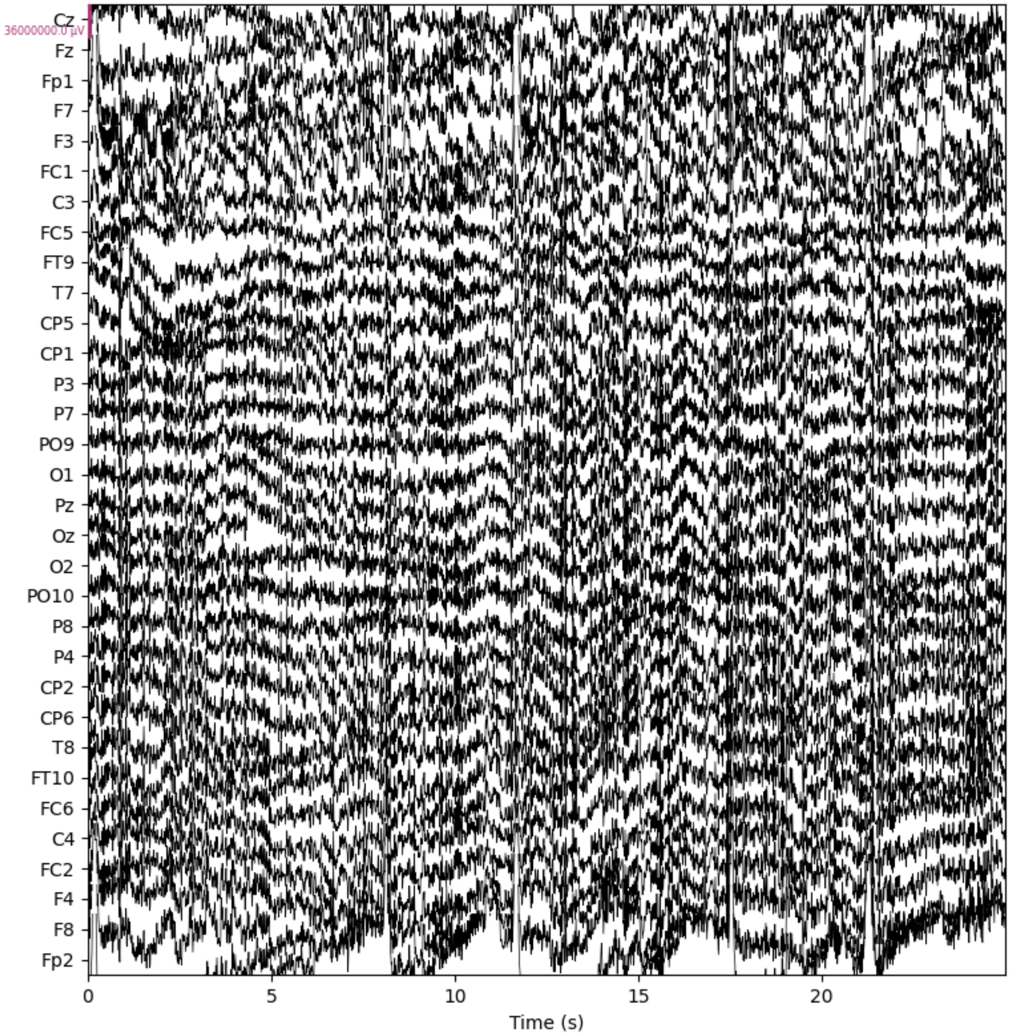
Importance of Filtering

We chose the **Butterworth Band-Pass filter** for its ability to isolate EEG activity (1-50 Hz) with minimal distortion, capturing essential neural rhythms. The **Savitzky-Golay filter** provides smoothing while preserving temporal details critical for EEG analysis. Finally, **ICA** is used to remove artifacts like eye blinks, improving the accuracy of neural signal interpretation.

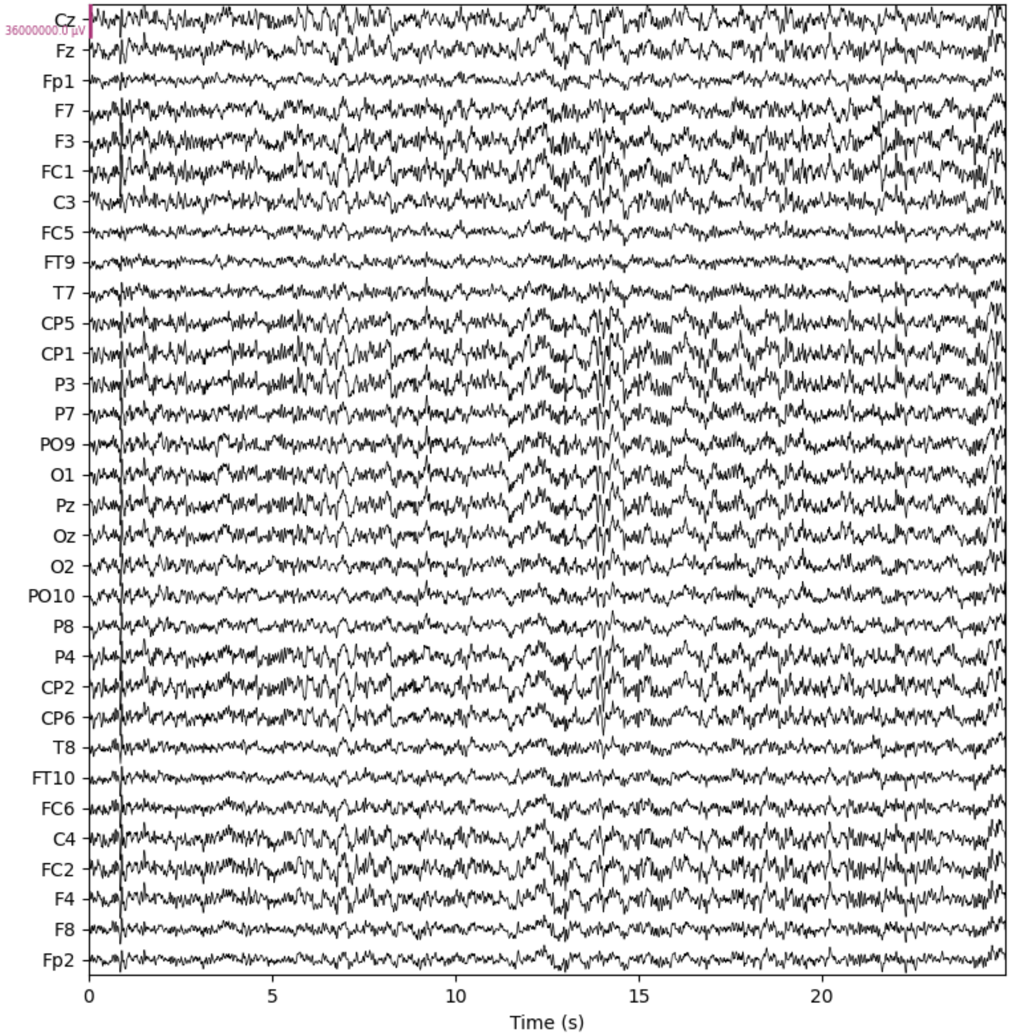
Other filters, like FIR, might introduce more delay or computational load, and moving average filters could oversmooth critical EEG components, leading to loss of important frequency content.

Plotting results

EEG signal before filtering



EEG signal after filtering



Feature Extraction

Analyzing EEG Data for Stress Prediction

These are the feature extraction techniques used in our project

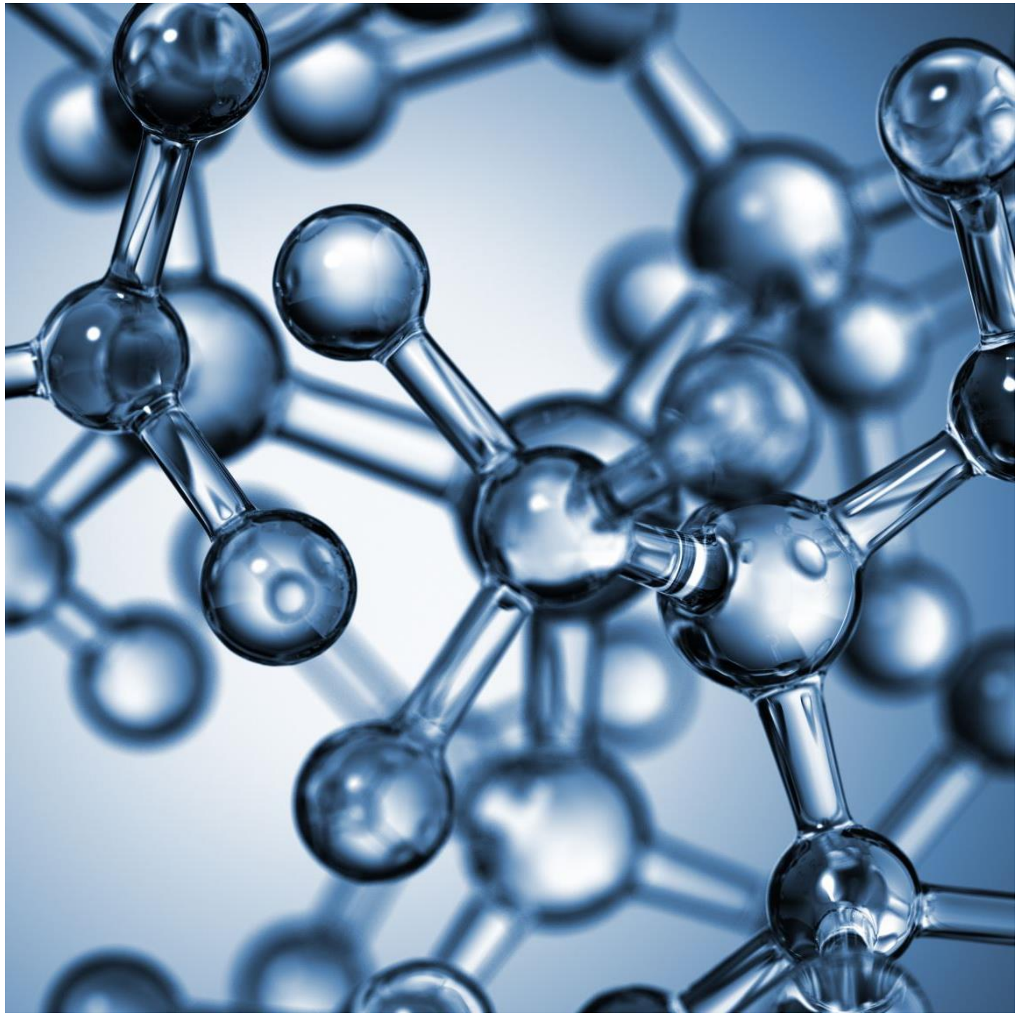
Time-Series Features

Frequency Band Features

Hjorth Features

Fractal Features

Entropy Features



1. Time-Series Features (Variance, RMS, and Peak-to-Peak Amplitude)

- **Variance**: Measures the spread of the data, indicating the signal's variability.

$$Variance = \frac{1}{N}\sum_{i=1}^{N}(xi - \mu)^2$$

- **Root Mean Square (RMS)**: Represents the magnitude of the signal and is useful for understanding the signal's overall power.

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i^2)}$$

- **Peak-to-Peak Amplitude**: Measures the difference between the maximum and minimum values in a signal.

$$Peak - to - Peak\ Amplitude = \max(x) - \min(x)$$

2. Frequency Band Features (Delta, Theta, Alpha, Beta, Gamma)

- These features involve **Power Spectral Density (PSD)** analysis to decompose the signal into frequency components.

- The most common frequency bands are:
- **Delta (0.5–4 Hz): Deep Sleep**
- **Theta (4–8 Hz): Drowsiness, relaxation**
- **Alpha (8–13 Hz): Calm, wakeful rest**
- **Beta (13–30 Hz): Alert, focused**
- **Gamma (30–50 Hz): High-level cognition, problem-solving**

- **Power Spectral Density (PSD)** is used to compute the power in each frequency band. It is computed as follows:
- $PSD = \frac{1}{N} \sum_{i=1}^{N} |X(f_i)|^2$

- These features are helpful for studying rhythmic brain activity and identifying patterns

Hjorth Features (Mobility and Complexity)

- **Mobility**: Reflects the signal's frequency content, indicating how much the frequency changes over time.It is computed as the ratio of the standard deviation of the first derivative of the sign

$$Mobility = \frac{\sqrt{Var\left(\frac{dx(t)}{dt}\right)}}{Var(x(t))}$$

- **Complexity**: Measures how complicated the signal's shape is. Higher complexity means the signal has a more intricate pattern, while lower complexity indicates simpler, more regular patterns.

$$Complexity = \frac{\sqrt{Var\left(\frac{d^2x(t)}{dt^2}\right)}}{Mobility.Var(x(t))}$$

4. **Fractal Features (Higuchi and Katz Fractal Dimensions)**

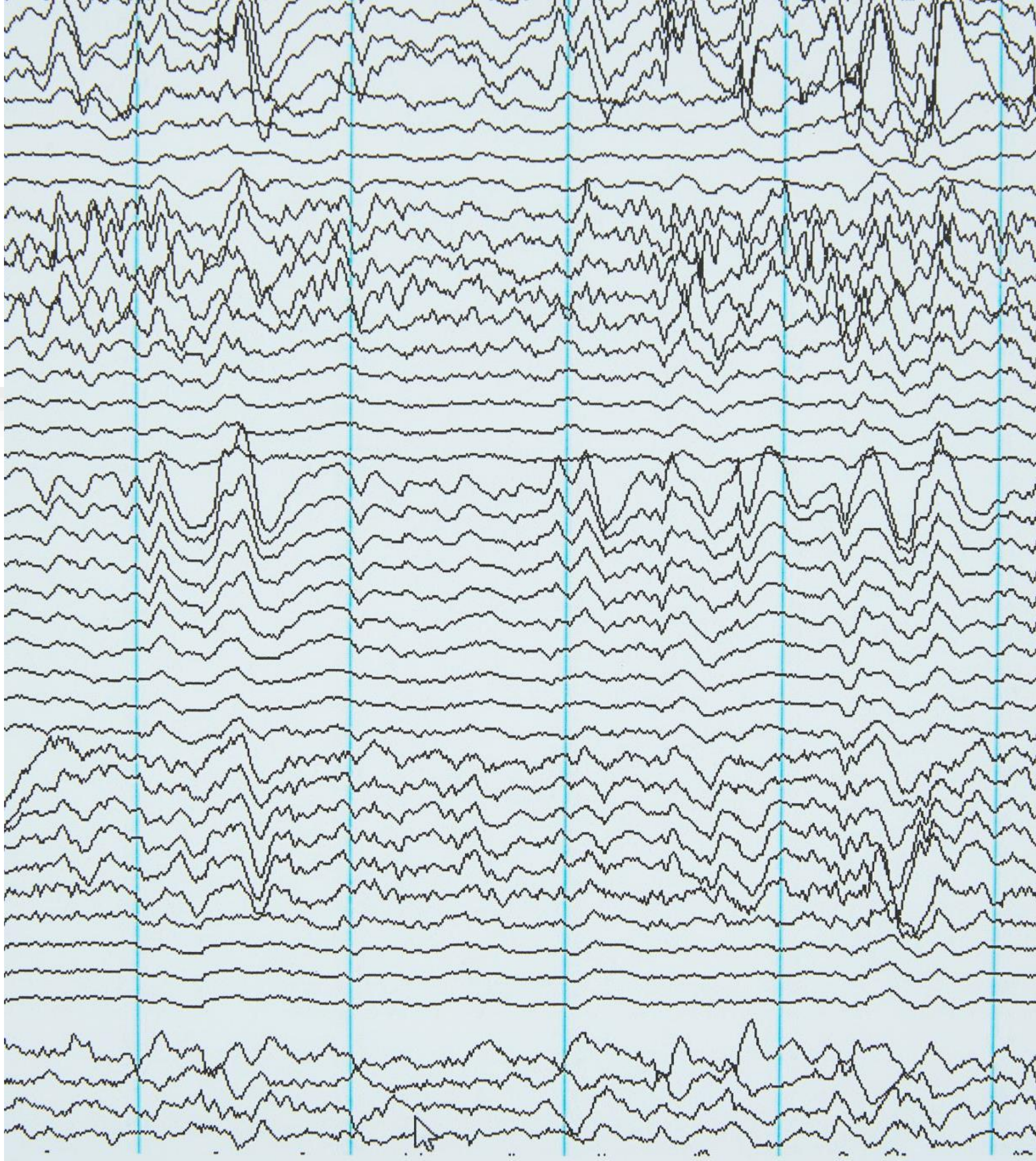
- Fractal dimensions measure the complexity of a signal in terms of self-similarity. Two common fractal dimensions are:
- **Higuchi Fractal Dimension**: Measures complexity by analyzing how the signal behaves when split into smaller segments. It looks at the rescaled range of the signal at different intervals.

$$D_H = \lim_{\tau \to 0} \frac{1}{N} \ln \left(\sum_{i=1}^{N-\tau} \left(\frac{|x(i) - x(i+\tau)|}{\tau} \right)^2 \right)$$

Katz Fractal Dimension: Estimates complexity by comparing the length of the signal to the distance between its points. It calculates how much the signal stretches within its enclosing box.

$$D_K = \frac{\ln \left(\frac{N}{\sum_{i=1}^{N-1} |x(i) - x(i+\tau)|} \right)}{\ln(N)}$$

5. **Entropy Features (Approximate, Sample, Spectral, and SVD Entropy)**



Entropy is a measure of unpredictability or randomness in the signal. Several types of entropy are commonly used for EEG signal analysis:

- **Approximate Entropy (ApEn)**: Measures the regularity and complexity of time series data:
- **Sample Entropy (SampEn)**: Similar to ApEn but avoids self-matching, making it more robust
- **Spectral Entropy**: Measures the entropy of the power spectral density (PSD) of the signal, quantifying its spectral complexity.
- **SVD Entropy**: Measures the entropy of the singular value decomposition (SVD) of the signal.

Convolutional Neural Networks (CNN)

- **Overview:** Convolutional Neural Networks (CNNs) are a class of deep learning models primarily used for processing structured grid data, such as images. They are particularly effective in tasks like image classification, object detection, and even processing sequential data, such as time-series or EEG signals.

Parts of CNN

Convolutional Layers: These layers apply convolution operations to the input data. They utilize filters (kernels) that slide over the input data to capture local patterns, such as edges or textures in images. The result is a feature map that retains important spatial hierarchies.

01

Activation Function: After convolution, an activation function (commonly ReLU) is applied to introduce non-linearity, enabling the network to learn complex patterns.

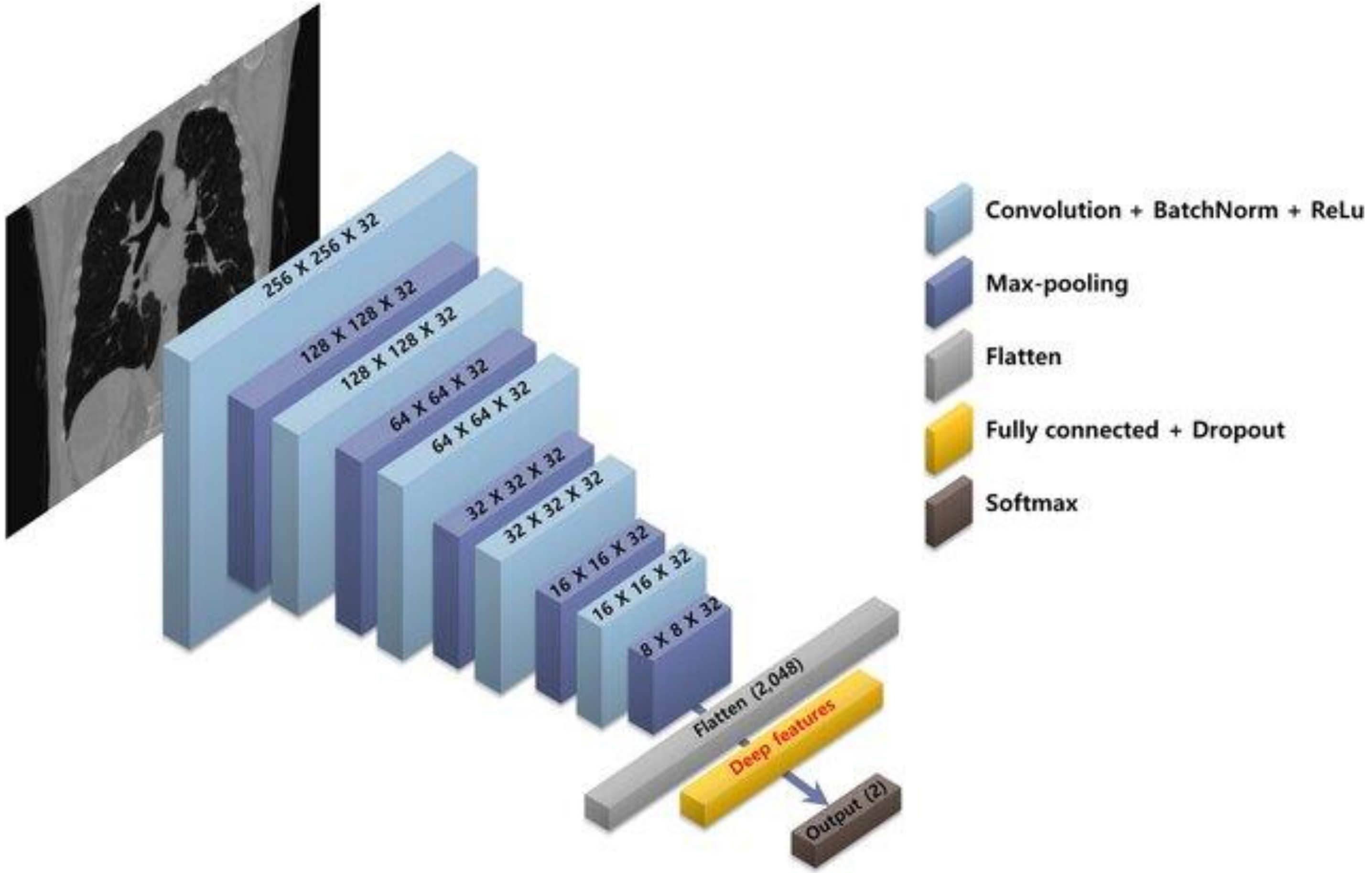
02

Pooling Layers: Pooling operations (like Max Pooling) are used to down-sample the feature maps, reducing dimensionality and computational load while preserving essential features.

03

Fully Connected Layers: At the end of the network, fully connected layers combine the features extracted by convolutional and pooling layers to make final predictions.

CNN Architecture:



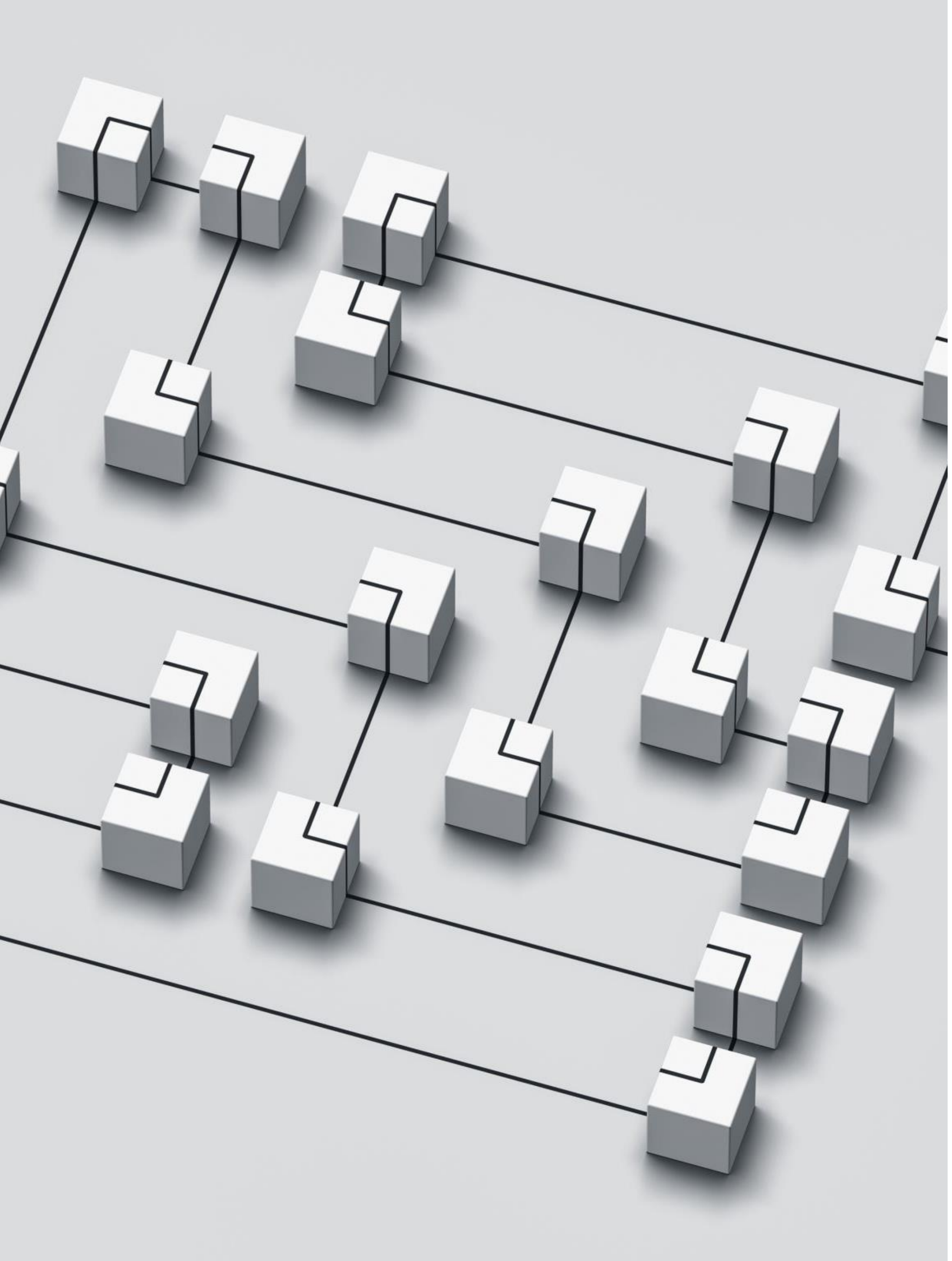
- **Formula:** The convolution operation can be mathematically expressed as:

$$Y(i,j) = \sum_m \sum_n X(i+m, j+n).K(m,n)$$

where Y is the output feature map, X is the input data, and K is the kernel/filter applied at each position.

LSTM

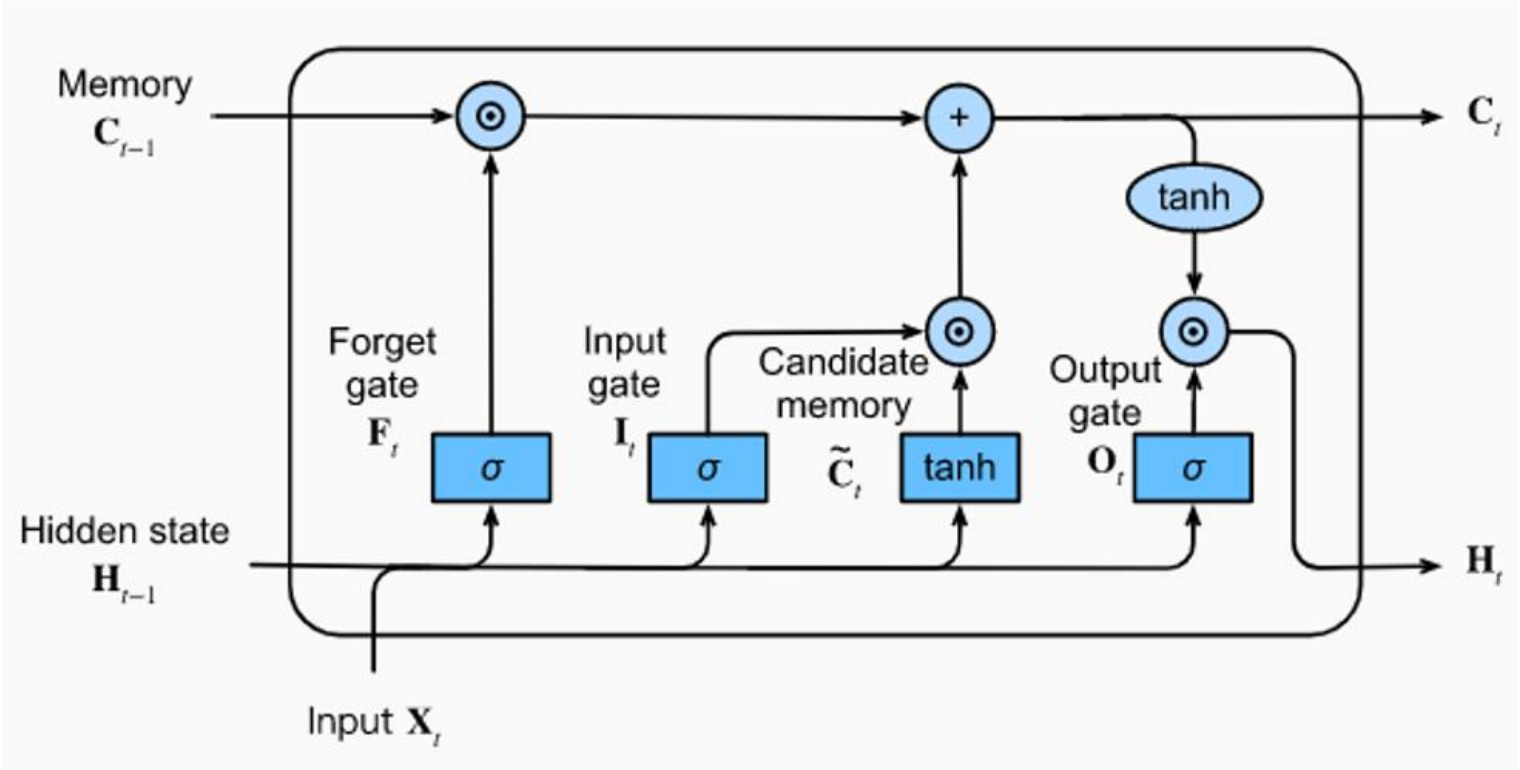
- **Overview:** Long Short-Term Memory network (LSTM) is a type of recurrent neural network (RNN) specifically designed to capture temporal dependencies in sequential data. They are widely used in applications such as time-series prediction, natural language processing, and speech recognition.



Key concepts

- **Cell State:** LSTMs maintain a cell state that carries relevant information throughout the sequence, enabling the model to remember or forget information as needed.
- **Gates:** LSTMs use three types of gates to control the flow of information:
    - **Forget Gate:** Decides what information from the cell state should be discarded.
    - **Input Gate:** Determines which new information should be added to the cell state.
    - **Output Gate:** Controls what information from the cell state should be output at the current time step.

LSTM Architecture:



- **Formula:** The operations of an LSTM can be summarized with the following equations:

1. Forget gate:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f$$

2. Input gate:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] +$$

3. Cell state update:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

4. Output gate:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o$$

$$h_t = o_t \cdot \tanh(C_t)$$

Our Model

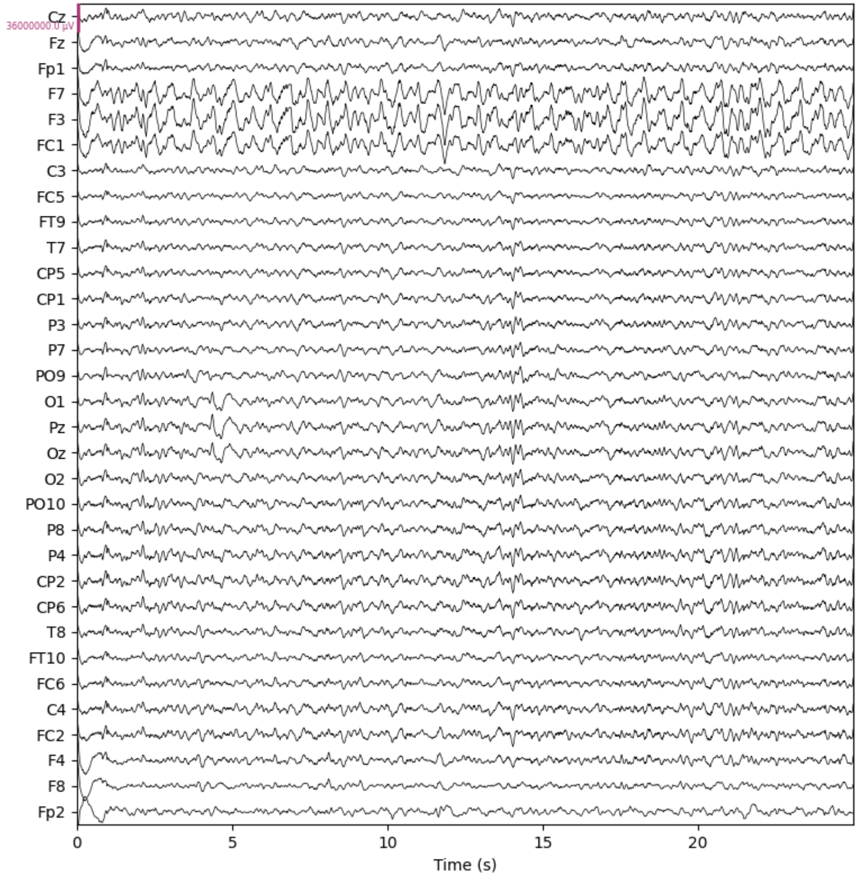
- **Content:CNN Layer:**
  - Input: Preprocessed EEG signals (e.g., spectrograms).
  - Role: Identifies spatial patterns and features in EEG signals.
  - Techniques: Convolutional layers, ReLU activation, MaxPooling.
- **LSTM Layer:**
  - Input: Flattened output from CNN.
  - Role: Captures sequential patterns in EEG time windows.
  - Techniques: Bidirectional LSTMs for improved temporal understanding.
- Fully connected layer maps output to stress levels.
- Visual: Show architecture with labeled layers and dimensions.

Model Performence

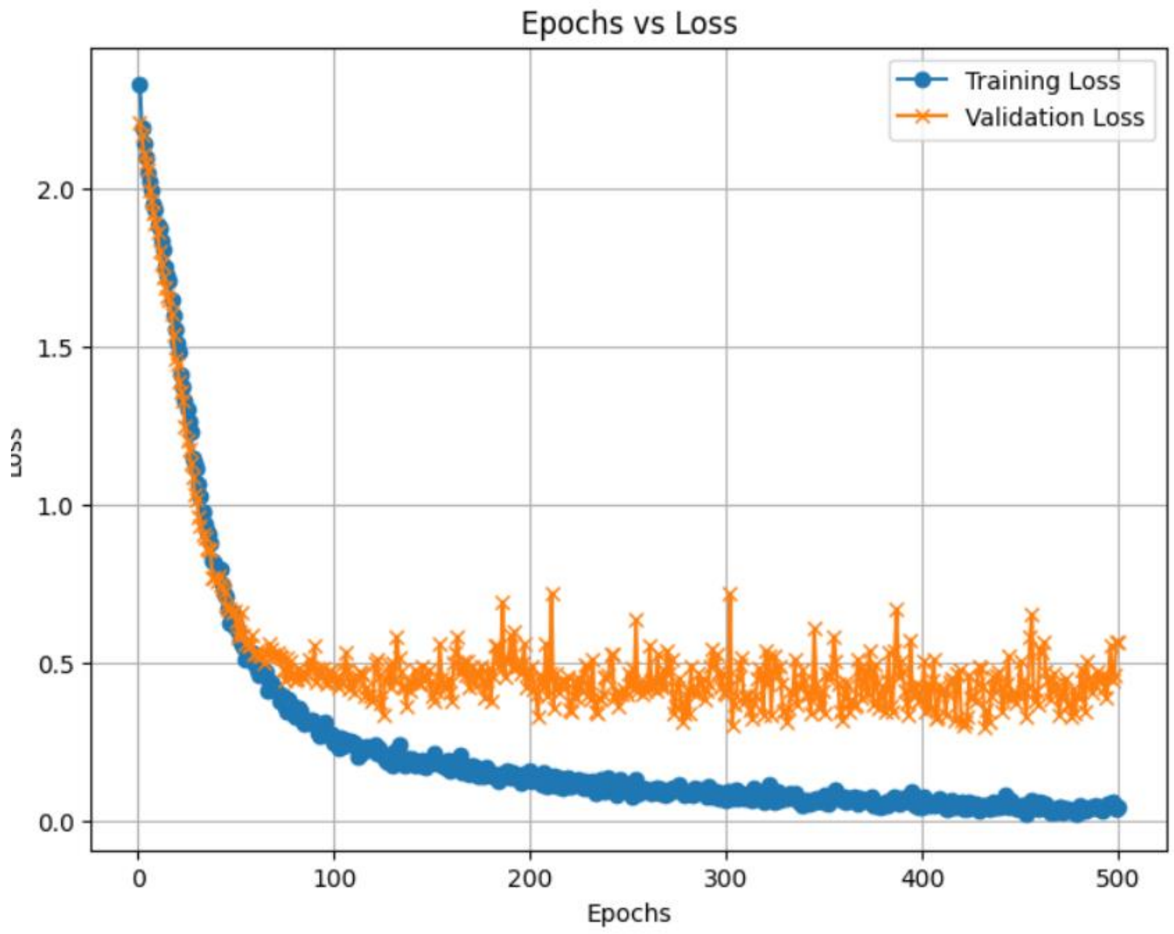
- Treditional ML model:
  - SVM: Accuracy = 55%
  - Random forest: Accuracy = 72%
- Multi Layer Perceptron:
  - Accuracy = 81%
- Our Model:
  - CNN & LSTM: Accuracy = 90% (Benchmark)

Results

Filtered Signal



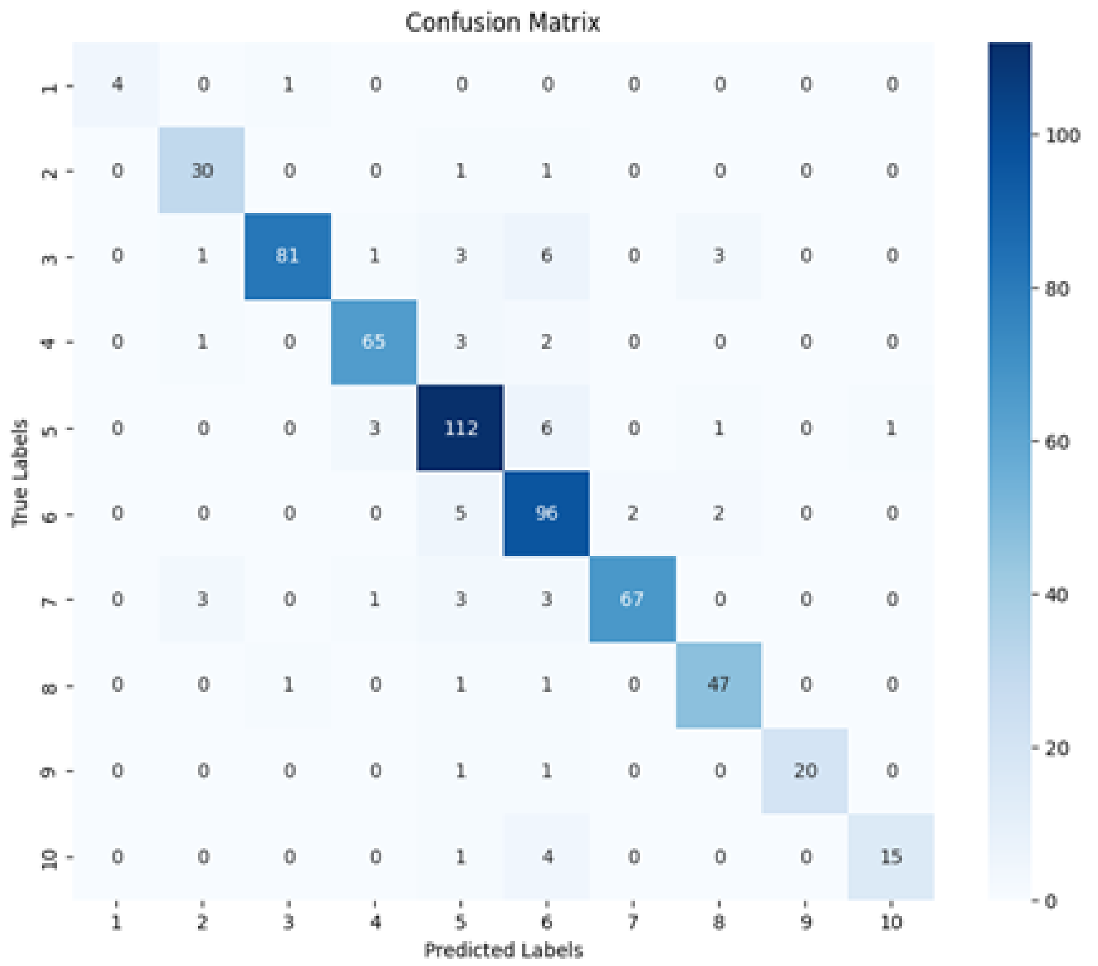
Epochs vs Training and Validation Loss



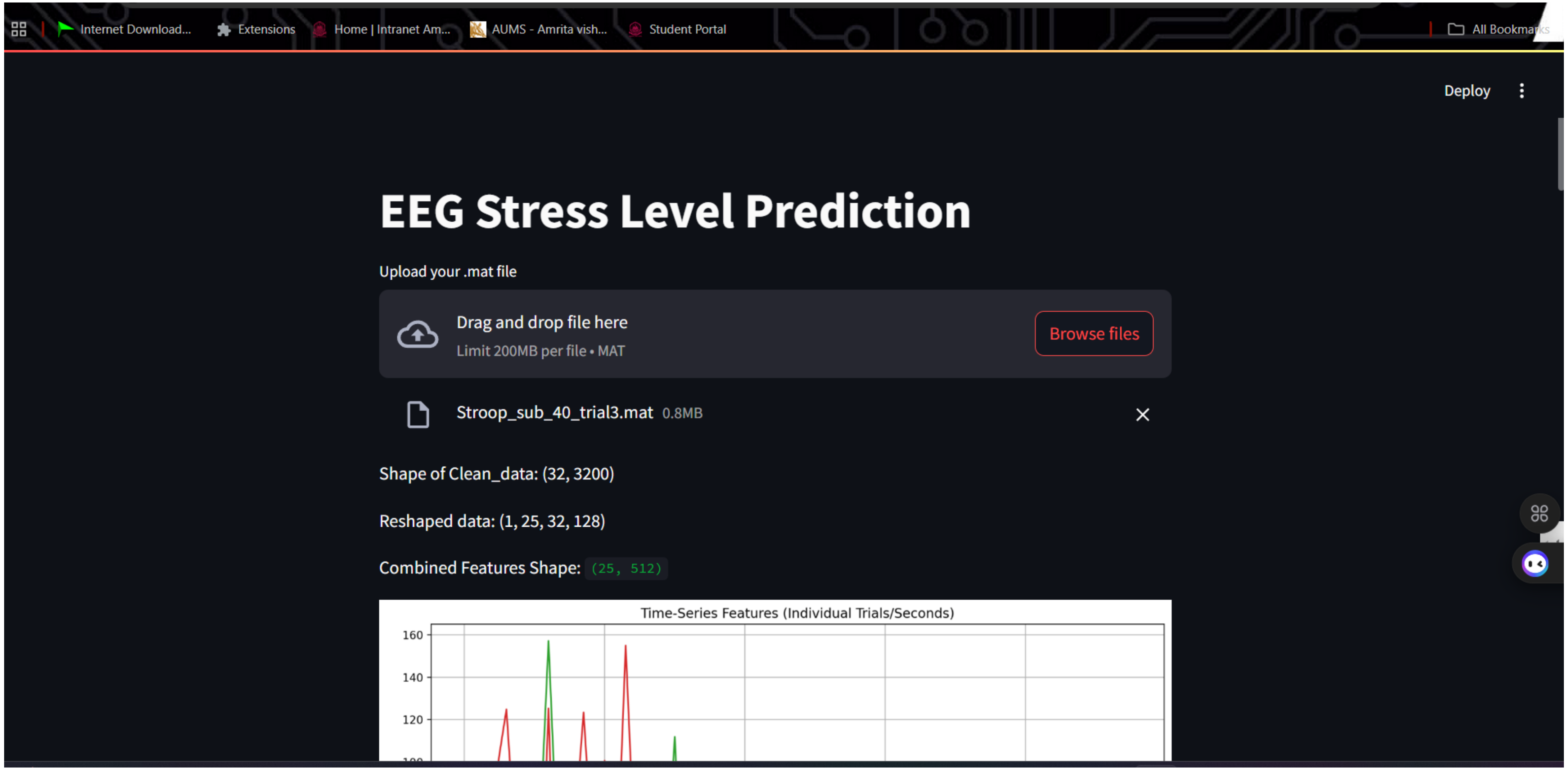
```
Classification Report:
              precision    recall  f1-score   support

     Class 1       1.00      0.80      0.89         5
     Class 2       0.86      0.94      0.90        32
     Class 3       0.98      0.85      0.91        95
     Class 4       0.93      0.92      0.92        71
     Class 5       0.86      0.91      0.89       123
     Class 6       0.80      0.91      0.85       105
     Class 7       0.97      0.87      0.92        77
     Class 8       0.89      0.94      0.91        50
     Class 9       1.00      0.91      0.95        22
    Class 10       0.94      0.75      0.83        20

    accuracy                           0.90       600
   macro avg       0.92      0.88      0.90       600
weighted avg       0.90      0.90      0.90       600
```



Streamlit Interface



For a given signal

- It will plot the Features extracted in the signal and real time prediction of each signal

CNN-LSTM Model Predictions

CNN-LSTM Model Predictions (Stress Levels for each second):

value
5
8
8
8
10
10
4
4
4
8

Average Stress (CNN-LSTM): 6.32, Maximum Stress (CNN-LSTM): 10

Literature review

- In our literature review, we referenced a total of 15 research papers covering various aspects of EEG-based stress detection, signal processing, and deep learning. The majority of these studies focused on traditional signal processing techniques, feature extraction, and classification-based stress detection models. However, they generally lacked advanced deep learning techniques, continuous stress level prediction, and synthetic data generation using Generative Adversarial Networks (GANs). Additionally, while several papers discussed stress detection, few incorporated real-time feedback or personalized stress management tools. Our project addresses these gaps by employing a regression-based deep learning model using CNN and LSTM for continuous stress prediction, leveraging GANs for synthetic data generation to improve model generalization, and integrating a real-time system with a user-friendly interface and AI-powered chatbot for personalized stress management advice.

Conclusion

- This project represents a significant advancement in the field of stress management by providing a comprehensive, real-time tool that leverages EEG signals and cutting-edge deep learning techniques(CNN & LSTM), including Generative Adversarial Networks (GANs). The system's ability to predict stress levels on a continuous scale of 1 to 10, coupled with its user-friendly interface and personalized AI-driven feedback, makes it a powerful resource for both individuals and healthcare professionals.

Thank you