Brain to Joint: Joint Velocity Decoding and Visualization

Team 1: Bingan Chen, Donovan Clay, Heather Wood, Meitong Li

University of Washington

May 2024

Outline

Project Goals Review

- Create a **GUI** that allows users to gain an **intuitive understanding** of some of the data **preprocessing** steps and **trade-offs** between different methods with regards to using EEG data to predict joint movements.
- This would:
  - Provide a brief roadmap to researchers and college students
  - Allow the user to gain intuition behind parameters through interactive visualization.

Review of Related Research

A Mobile Brain-Body Imaging Dataset Recorded during Treadmill Walking with a Brain-Computer Interface [1]:

- Unique comprehensive dataset
- Unscented Kalman filter
- Artifact Removal

Figure: Dataset Collection

The MNE-Python to visualized the data [2]

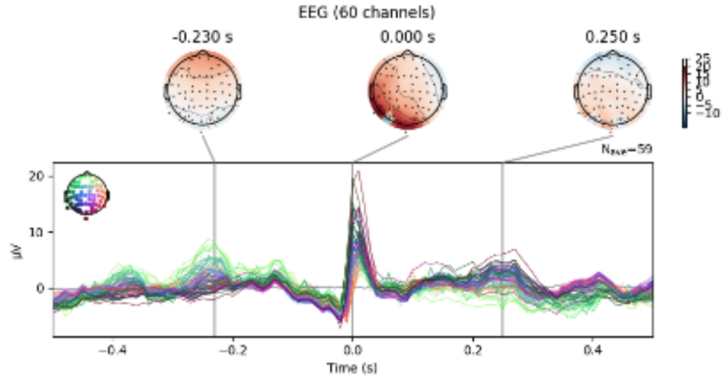


Figure: MNE-Python

- Pros: Good Visualization
- Cons: Requires high technical knowledge

Remove Ocular Artifacts

Figure: Ocular Artifact Position

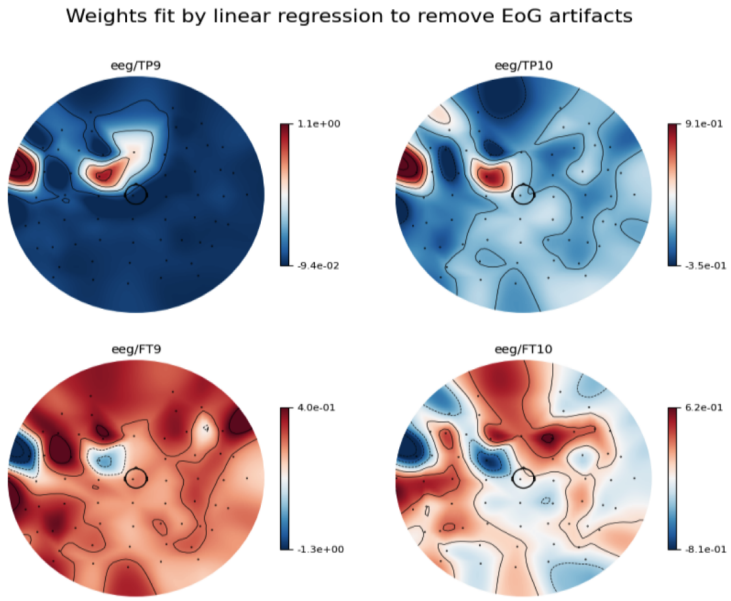


Figure: Ocular Artifact Removal

$$\mathrm{EEG}_{\mathrm{cor}} = \mathrm{EEG}_{\mathrm{raw}} - \gamma F(\mathrm{HEOG}) - \delta F(\mathrm{VEOG})$$

Remove Muscle Artifacts

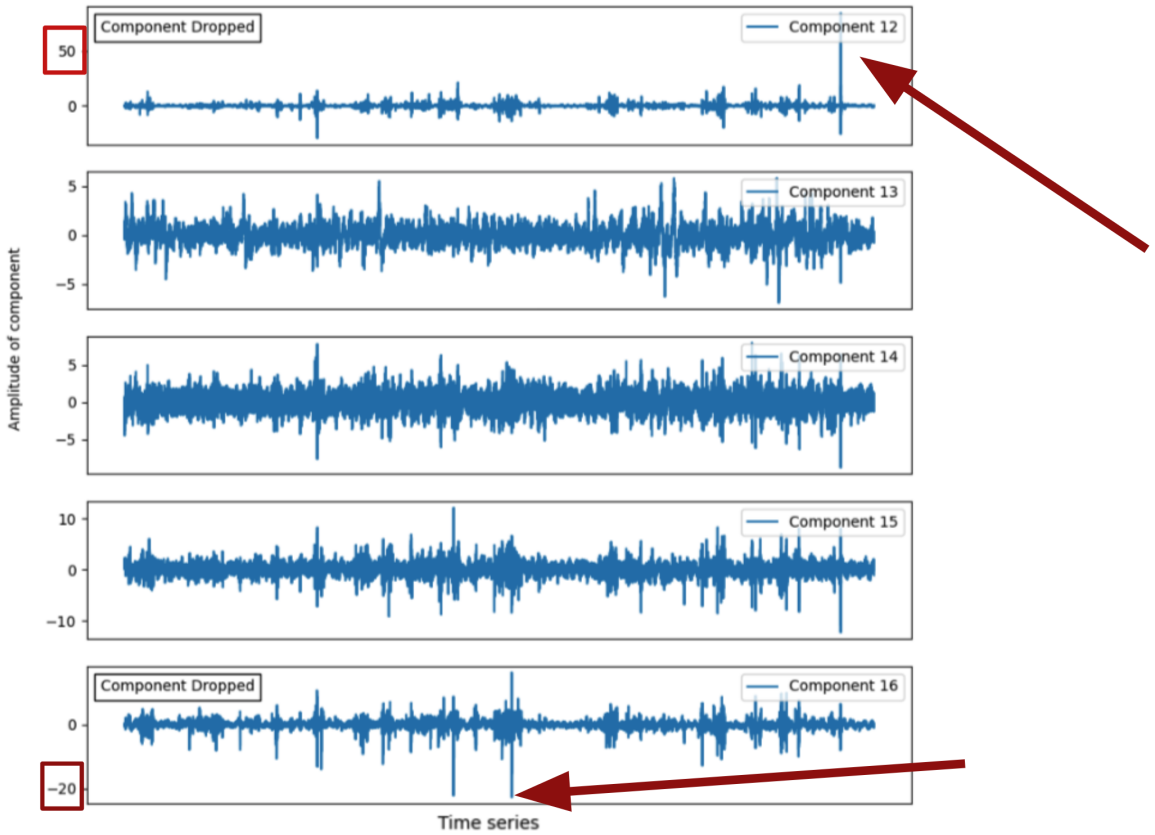


Figure: Muscle Artifact

Bandpass Filtering

- Bandpass filtering removes other more arbitrary noise from the data after known artifacts have been removed.
- We filter the data between 0.09 and 45 Hz.

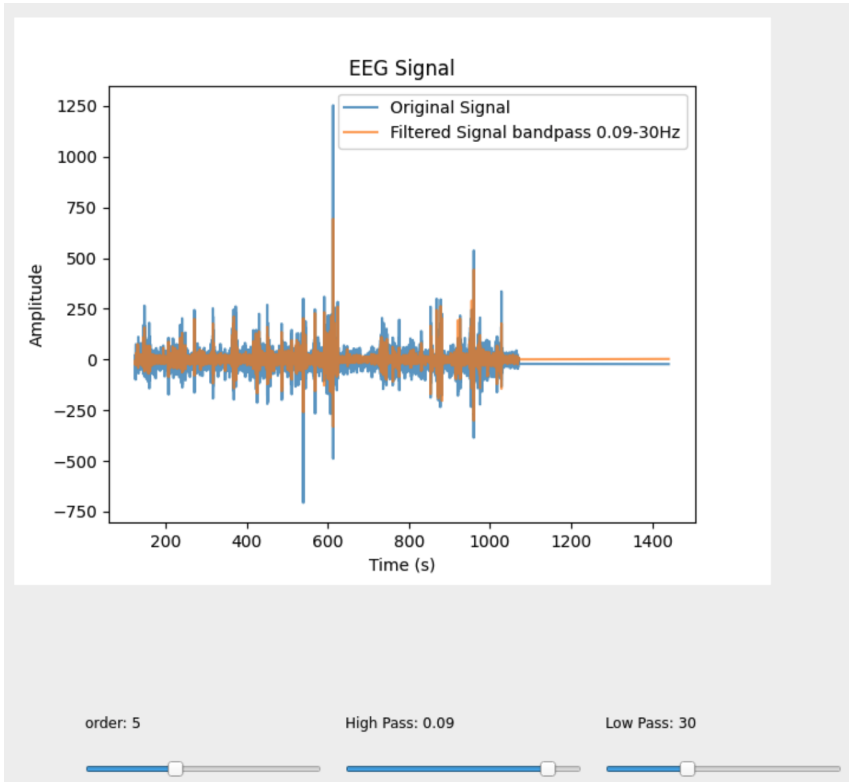


Figure: Bandpass Filtering

Normalization and Filtering

Velocity: Periodic data type

- Gaussian filtering

$$y_{\text{norm}} = \frac{y' - \min(y)}{\max(y) - \min(y)}$$

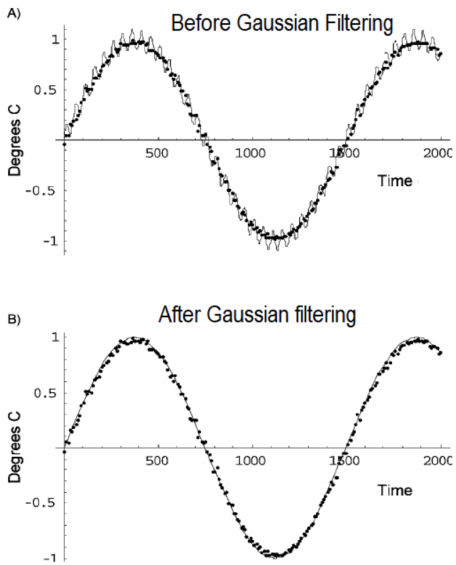


Figure: Velocity Normalization

EEG: Long tailed data type: huber normalization

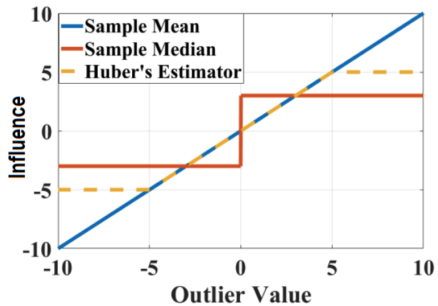


Figure: Huber Normalization

Velocity Data

- Velocity believed to correspond more to brain signals than joint position

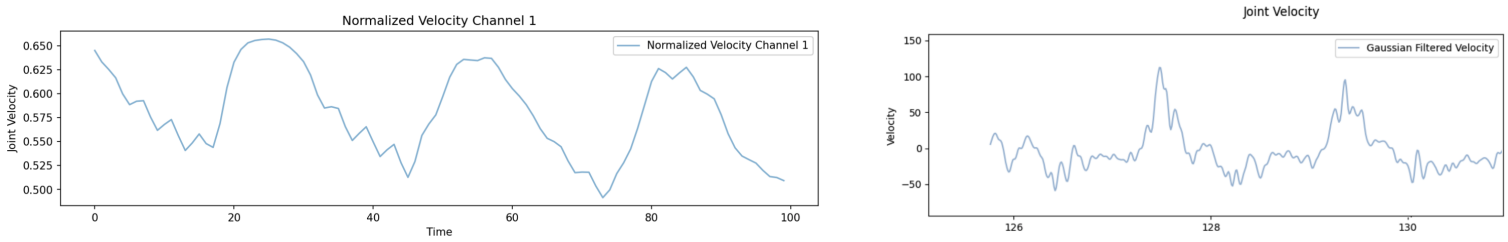


Figure: Velocity Data Calculation

The mapped angle, y_{map}, is calculated using the formula:

$$y_{\text{map}} = \frac{y}{x} \cdot x_0$$

- x: the joint angle maximum.
- $x_0 = 90^\circ$ for the hip and knee joints.
- $x_0 = 43^\circ$ for the ankle.

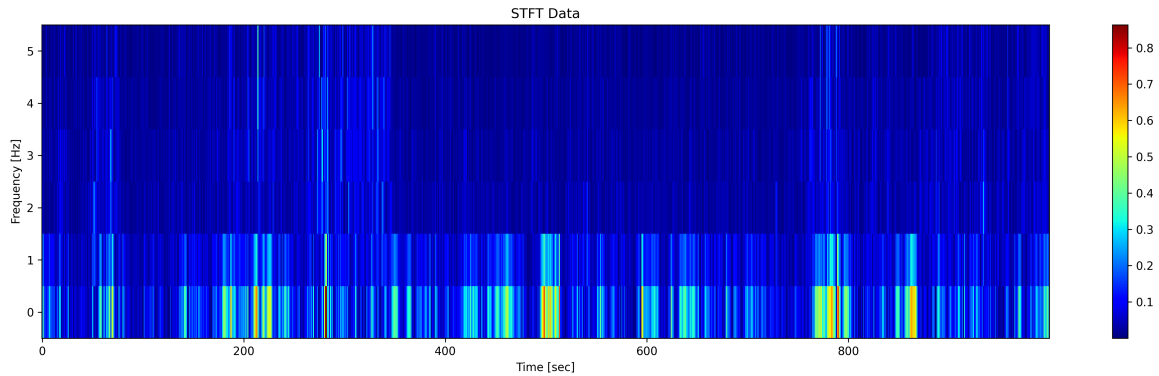
EEG Data Processing with STFT

- Short Time Fourier Transform (STFT) is used to convert the EEG data into a time-frequency domain.

$$X[k,n] = \sum_{m=0}^{N-1} x[m]w[n-m]e^{-j\frac{2\pi km}{N}}$$

where N is the number of points in each segment, k indexes the frequency bins, and n indexes the time steps.

- This allows us to see how the frequency of the brain signals change over time windows to map to the joint velocity.



LSTM

- Long Short-Term Memory (LSTM) is a type of recurrent neural network that is capable of learning long-term non-linear dependencies. It **selectively retains** or **forgets** information over time.

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$

  - c_t: New cell state at time t.
  - f_t: Output of the forget gate, decides which parts of the previous cell state c_{t-1} to retain or forget.
  - c_{t-1}: Previous cell state.
  - i_t: Output of the input gate, controls how much of the new candidate cell state to add to the cell state.
  - $\tilde{c}_t$: New candidate cell state, generated by processing the current input.

LSTM (Visualization)

The hyperparameters of the LSTM model are:

- **hidden size**: 50
- **num layers**: 2
- **dropout**: 0.85

```python
class LSTM(nn.Module):
    def __init__(self, input_size, hidden_size, num_layers, num_classes, dropout=0.5):
        super(LSTM, self).__init__()
        self.hidden_size = hidden_size
        self.num_layers = num_layers
        self.lstm = nn.LSTM(
                            input_size,
                            hidden_size,
                            num_layers,
                            batch_first=True,
                            dropout=dropout if num_layers > 1 else 0
                            )
        self.batch_norm = nn.BatchNorm1d(hidden_size)
        self.fc = nn.Linear(hidden_size, num_classes)
        self.dropout = nn.Dropout(dropout)

    def forward(self, x):
        h0 = torch.zeros(self.num_layers, x.size(0), self.hidden_size).to(x.device)
        c0 = torch.zeros(self.num_layers, x.size(0), self.hidden_size).to(x.device)
        out, _ = self.lstm(x, (h0, c0))
        out = self.batch_norm(out[:, -1, :])
        out = self.dropout(out)
        out = self.fc(out)
        return out
```

Figure: LSTM architecture

LSTM (Result)

- Overfitting
- Batch normalization
- Dropout layer

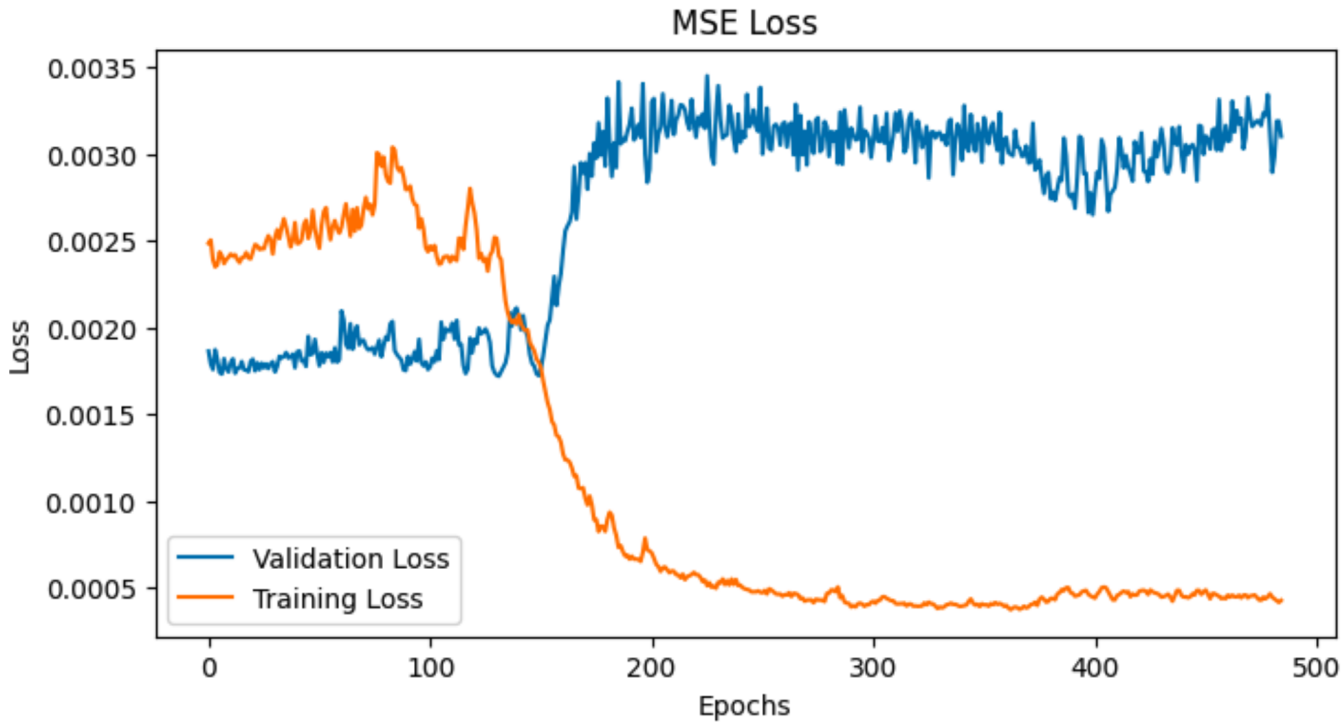


Figure: LSTM result

LSTM (Prediction)

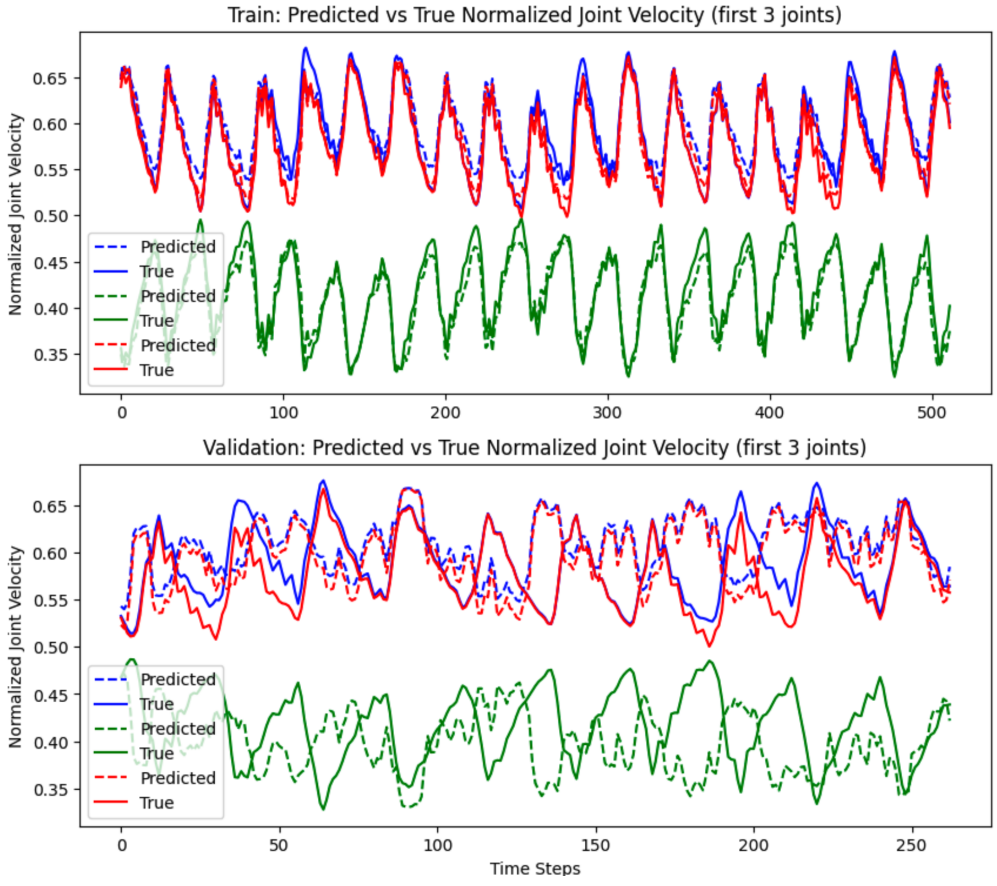
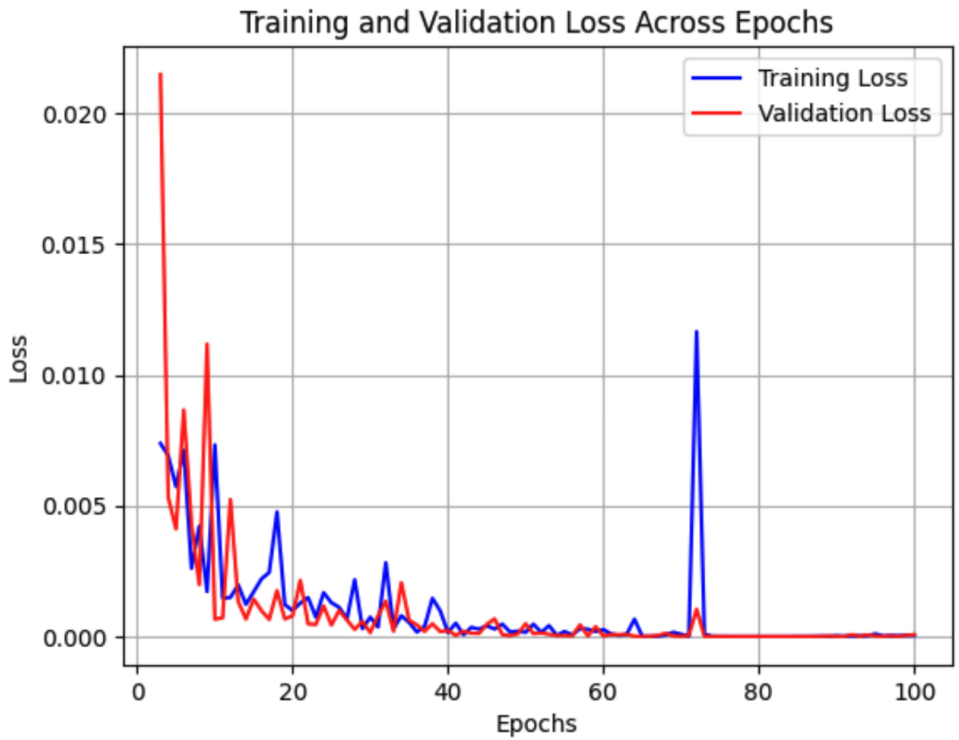


Figure: Train MSE: 0.0004332350 Test MSE: 0.0035434817

Transformer

- The Transformer model is a type of neural network architecture that is designed to learn **complex long time dependencies** the data efficiently.



Interface Development

- The GUI will allow users to interact with the data and see the results of the preprocessing and prediction steps.
- The GUI is built using the `PyQt5` library.

GUI Demo

GUI

Takeaways

- Biological data is hard to work this
- Data preprocessing is a pain
  - Domain specific
  - 446/546 doesn't teach this!
- Median Filter: Take the median in a specific time interval. Smoothing data too much, especially for the joints velocity data point that are significant but subtle.

Next Steps

- GUI Improvements
  - Add EEG and Velocity data preparation modules.
  - Add models to the interface.
  - Add hover-over descriptions for better user guidance.
  - Add more labels for easier interpretation of data.
  - Fix some labels for clarity and accuracy.
- Models/Data Processing Enhancements
  - Improve the performance of LSTM models.
  - Implement machine learning with other trials or subjects to generalize the model.
  - Scale velocity data backup and integrate to compare with joint data for enhanced analysis.

References

[1] He, Yongtian, et al. "A Mobile Brain-Body Imaging Dataset Recorded during Treadmill Walking with a Brain-Computer Interface." Scientific Data, vol. 5, no. 1, 24 Apr. 2018, https://doi.org/10.1038/sdata.2018.74. Accessed 2 Apr. 2020.

[2] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A. Engemann, Daniel Strohmeier, Christian Brodbeck, Roman Goj, Mainak Jas, Teon Brooks, Lauri Parkkonen, and Matti S. Hämäläinen. MEG and EEG data analysis with MNE-Python. Frontiers in Neuroscience, 7(267):1–13, 2013. doi:10.3389/fnins.2013.00267.

[3] Tangri, R. Robust Statistics: The Influence Function. Medium. https://rohan-tangri.medium.com/robust-statistics-the-influence-function-d71ac687d046.

[4] Loehle, Craig. "Estimating Climatic Timeseries from Multi-Site Data Afflicted with Dating Error." Mathematical Geology, vol. 37, no. 2, Feb. 2005, pp. 127–140, https://doi.org/10.1007/s11004-005-1305-6.

References

[1] Belkhiria, Chama, and Vsevolod Peysakhovich. "Electro-Encephalography and Electro-Oculography in Aeronautics: A Review over the Last Decade (2010–2020)." Frontiers in Neuroergonomics, vol. 1, 21 Dec. 2020, https://doi.org/10.3389/fnrgo.2020.606719.