UNIVERSITY OF MARYLAND

Neuromechanics Laboratory

Intention Detection System Documentation

Directed by Dr. Li-Qun Zhang

Supervised by Giovanni Oppizzi

Written by Michael Penafiel

Contents

Introduction

The Electromyography (EMG) Intention Detection System is developed in the Neuromechanics Laboratory with the intended use for stroke rehabilitation and patient training. The person operating the program will be administering the test and will be referred to as the *user*. The person whose EMG is being recorded, and who will receive treatment is referenced as the *patient*. There are three primary functions of the application:

1. Collect EMG data from the patient using the MindRove Armband.
2. Train a model that represents a profile for the patient using a given data set.
3. Use the model to predict the intention of a patient via new data fed into the model, and relay the feedback to the Hand of Hope to assist the patient.

The Mindrove Armband is an 8-channel low-cost surface EMG (sEmg) system that can be connected to a computer running the Intention Detection program via Wi-Fi. sEMG data is streamed to the computer and processed automatically for the machine learning (ML) model. There are a few options for the user to parse the data before training a model. The model is an Artificial Neural Network (ANN) and will accept the EMG data as the input and produce the associated pose for the output.

Once a model has been trained and cross-validated, it can be used to test and train the patient. The user can create a series of poses that the patient must try and perform. For each pose, the system will collect a new data set from the patient and validate it. If it is the correct pose, the HoH will assist the patient. The patient must be intentional when creating the pose as this will provide a stronger reinforcement. An isometric contraction will produce a stronger EMG.

Technical Documentation

The application communicates with two systems to create a closed environment. It will stream data from the MindRove Armband as well as relay commands to the HoH robot.

Placing the Armband

A secure fit and clean contact between the leads of the armband and the skin will produce a high-quality signal with a low SNR. Before placing the armband on the patient, use an abrasive pad over the forearm to remove dead skin. After, use an alcohol wipe to clean the surface of the leads, as well as the skin of the patient. Be careful not to touch the contacts after cleaning them. Do not stretch the armband over the patient's forearm, as this may damage the wires connecting the leads.Undo the clasp to create enough slack in the armband that it goes over the patient's forearm with little to no resistance. Position the main lead of the armband over the ulna.

Figure 1: MindRove Armband

Take note of the placement of the armband, as placement in future sessions should be the same for each patient to minimize variance between sessions.

Connecting Armband

To connect with the armband, first turn on the device. The three LEDs will flash green, red, and blue before turning a steady green. Then search for the device in the available Wi-Fi networks. It will appear as “MindRove_ARB_<Serial>” where the last part is a serial code. An example of how appears is shown below. If a password is required, use *#mindrove*. Before using the system, make sure the armband is connected. Currently, there is no visual cue to verify the armband is connected from the user interface. It needs to be checked under the Network or Wi-Fi settings.

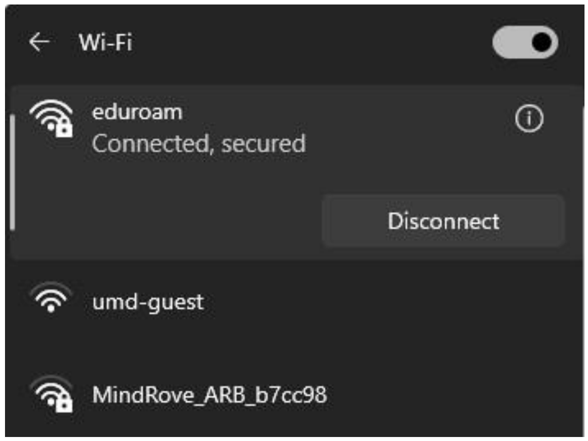
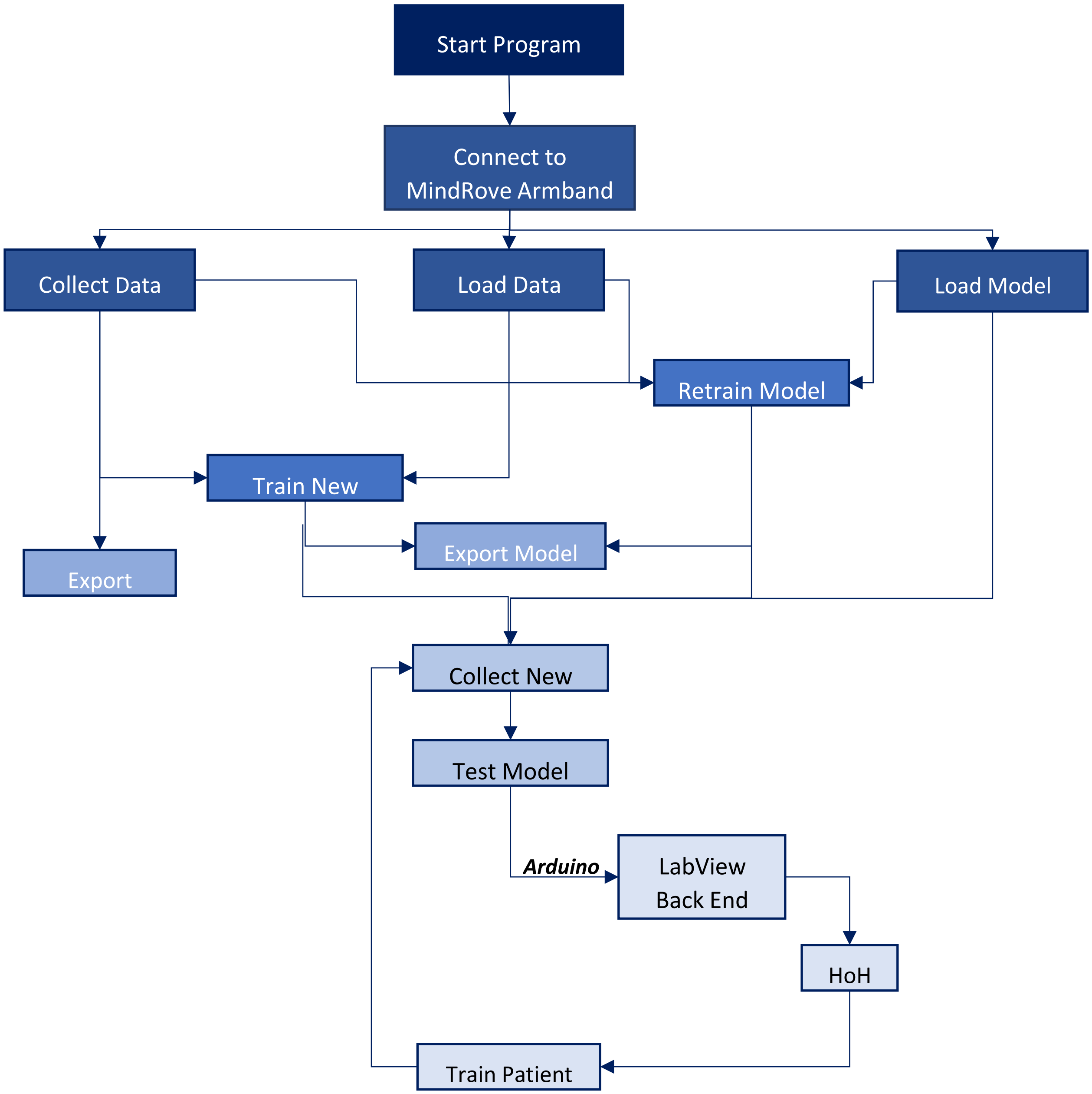


Figure 2: Example of the MindRove Armband appearing in the available Wi-Fi networks.

Hand of Hope Interface

The HoH is connected via a microcontroller that relays commands to a separate computer system with the LabView backend. When training a patient, data will be collected from the patient using the armband. This data is then processed and fed into the model which outputs the task. This is communicated to the microcontroller, which sends a signal to the LabView backend that codes for the position. The backend controls the HoH, which then assist the patient finish the task. moves to the same pose that the patient was trying to create.

Processes Flow Chart



User Guide

When the user opens the application, they will see the main tab in the window. This displays useful information at a glance, as well as direct access to the three primary functions. The status panel any relevant information about the patient, and the current session. This includes the total number of tasks that appear in a prompt, the subject name, the filename for collected or loaded EMG data and model, the interval in which data collection is taken, and the status of the MindRove and the port that connects with the HoH. Below the status panel, there are options to configure and connect the port.

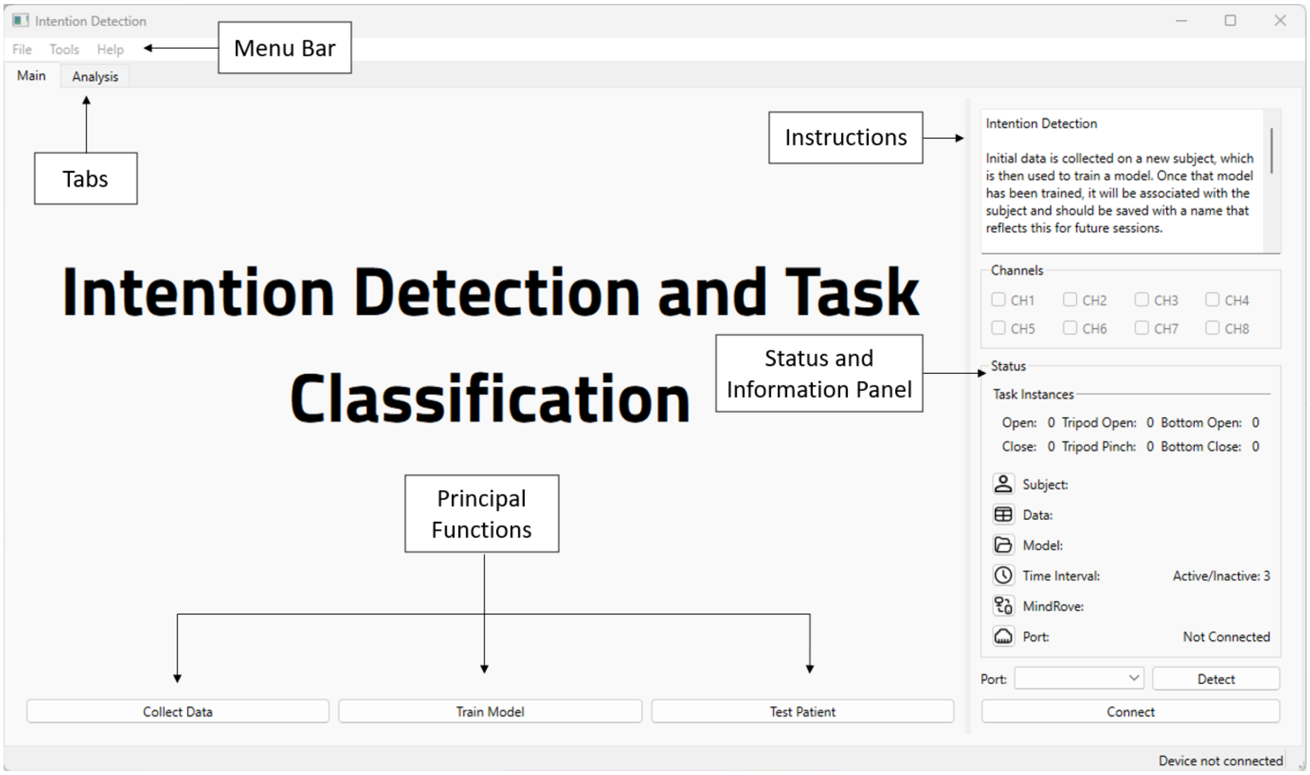


Figure 3: Main tab of the EMG Intention Detection program.

There are three primary functions that are located at the bottom of the main tab. The button on the left will initiate communication between the MindRove Armband and the program. This streams and processes the data, which includes operations to filter and rectify the data.

There are 6 tasks that are available in the current system. They can be seen in the table below. Each task is given an associated id that can be used to create a new sequence.

TASK NAME	ID
OPEN	1
CLOSE	2
TRIPOD OPEN	3
TRIPOD CLOSE	4
BOTTOM OPEN	5
BOTTOM CLOSE	6

Open 	**Tripod Open** 	**Bottom Open**
 **Close**	 **Tripod Pinch**	**Bottom Pinch**

Table 1: 6 Tasks as they appear in the program. Each task has an associated code that the program references.

Collecting

To collect data from the patient, the user must create a new task sequence. There are 5 different ways to create a new sequence of tasks.

1. **Create a New Task Sequence**: This function can be found under Tools>Create Task Sequence. This opens a new dialog where the user can select the number of repetitions for each task. The total number of tasks is displayed in the bottom left corner of the window. The user can choose to shuffle the tasks randomly. If this option is not selected the tasks will appear in the following order: Open, Close, Tripod Open, Tripod Pinch, Bottom Open, Bottom Pinch.

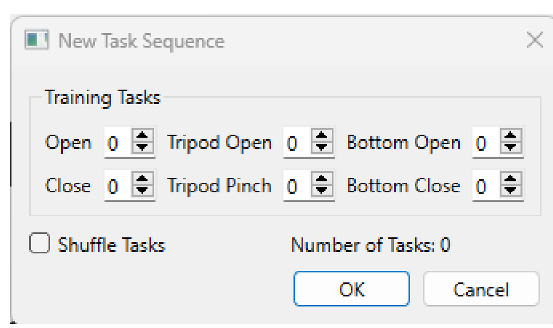


Figure 3: Create new task sequence. Includes an option to shuffle and total tasks.

2. **Create Custom Task Sequence**: This function provides full control of the order the tasks are made. The user can input up to 30 tasks using their id, separated by a space.
3. **Create Randomized Task Sequence**: If the user wants to create a random sequence to train the patient, the can do so using the Randomized Task Sequence found under Tools>Create Task Sequence. There is an option to assign a length, up to 30 tasks long. If this option is not selected, a random length is chosen.
4. **Create Training Parameters**: If the user is interested in assigning an equal number of repetitions for each task, they can do so using the Training parameters found under Tools. The user can select each task they want to include as will as the number of repetitions, up to 30 for each task. This results in a total of up to 180 tasks. The tasks are automatically shuffled.

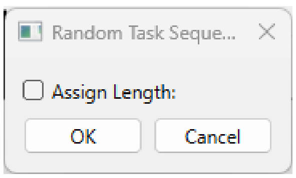


Figure 4: Random Task Sequence Dialog

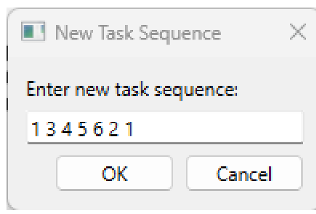


Figure 5: Create custom task sequence. This option gives the most freedom over order of tasks.

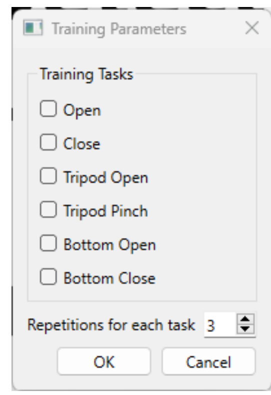


Figure 6: Create training parameters. This dialog provides options to choose the tasks and number of repetitions.

Once the task sequence has been created, the user can begin collecting EMG data from the patient. When **Collect Data** is clicked, there is a 5 second countdown to prepare the patient. The collection phase always begins with rest, and there is a rest period in between each task. The patient should create the task as they appear on the screen. The tasks are set to a 3 second interval, which is displayed in the top right corner of the screen. This gives the user and patient an indication when the next task will appear. The interval can be changed by either clicking the time icon in the status panel or under Tools>Set Time Interval. This opens a dialog with the option to change both the active interval, i.e. the period of active contraction, and the inactive interval, i.e. the period of rest. The current time interval can also be seen in the dialog.

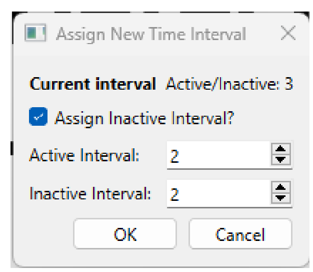


Figure 7: Change time interval. Assign different intervals for active and inactive periods. The current intervals are displayed here as well.

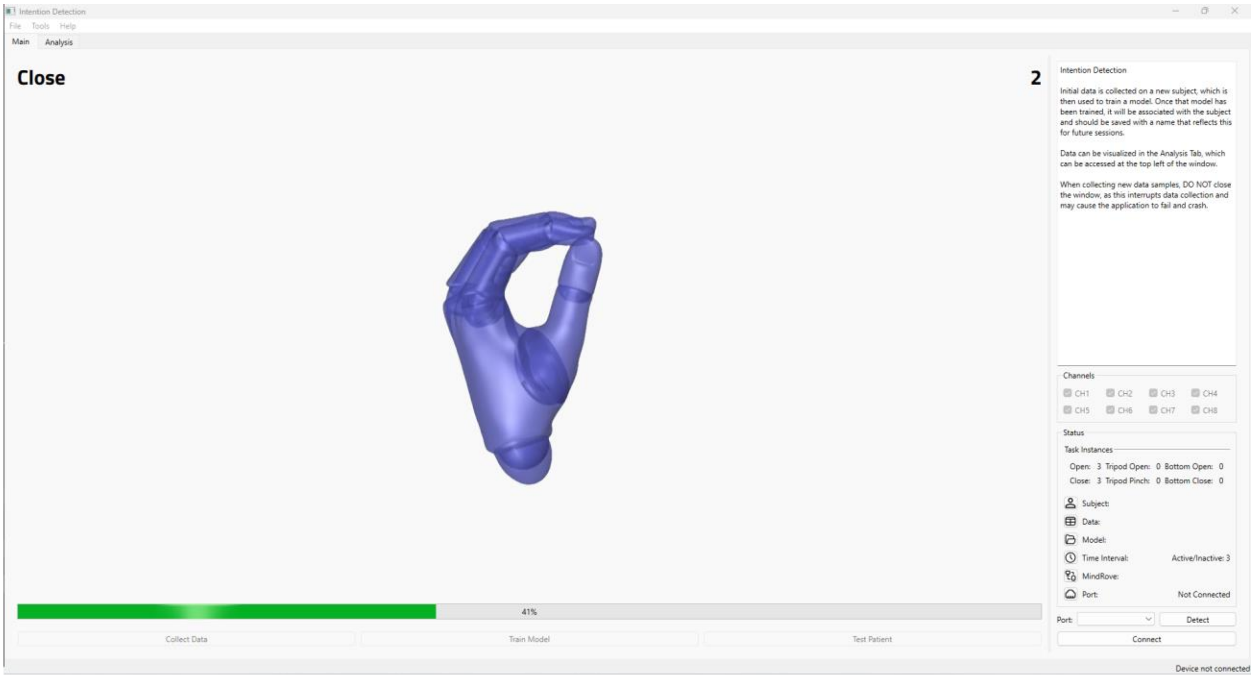


Figure 4: Example of the UI while collecting data. The interval is displayed in the top right pf the window, and the current pose is displayed in the top left. Progress can be viewed at the bottom of the screen.

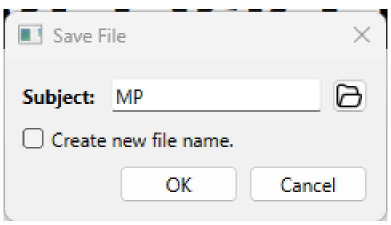


Figure 8: Dialog option that appears to save data.

When the collecting phase is complete, a dialog pops up, containing two fields. The first is the subject name. If the subject folder was created or opened before collecting data, it will appear here. Otherwise, the user can click the folder icon to access or create a new patient folder. The second field is a checkbox indicating whether the user wants to create a new filename. If this option is selected it opens the file explorer where it can be saved in the patient folder. Otherwise, it saves it in the patient folder using the timestamp. Any instance of saved data will create three files:

1. Raw data file, containing the filtered data
2. Rectified data file, containing the RMS envelope, i.e., absolute value of the signal.
3. Normalized data file, containing the processed data. Signal is normalized from 0 to 1. This is the data used to train the model.

Training the Model

Before training the model, the user must verify that a quality signal is being recorded from each channel. The data can be viewed in the Analysis Tab which can be accessed in the top left of the window. This contains plots of the 8 channels from the armband, as well as a plot of the poses. The normalized data is displayed once it has been collected from the patient and will appear as shown in **Figure 9**.

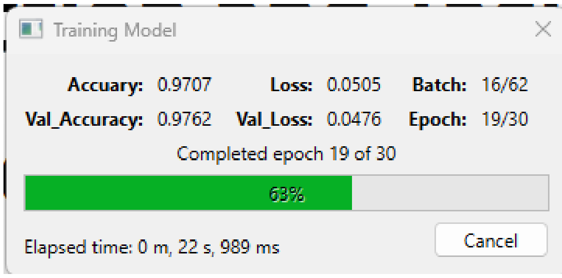
If any channel does not have the desired shape, or if a significant number of spikes occur that skew the shape of the signal, it can be excluded when training by selecting the checkboxes in the right of the window. The selected channels can also be viewed and selected in the main tab. Spikes that occur may distort the signal and decreases the accuracy when training the model. This can be limited by cleaning the leads and skin of the patient, and by ensuring a snug fit so that the leads maintain contact. If the user is still unable to obtain a clean signal from the patient from a channel, it can be excluded from the training. This is not the ideal solution, because subsequent sessions using the model will need to have the same channel configuration. For, example, if channel 7 was excluded before the model was trained, then it can no longer be used in future sessions.

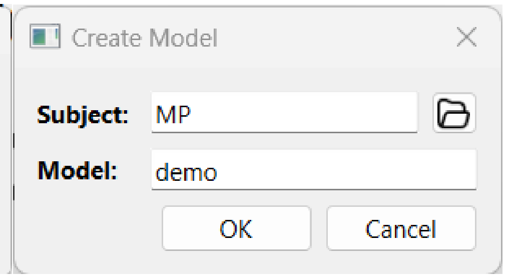
Figure 9: Example of the Analysis Tab displaying rectified data. The data that appears is a clear example of how it should look when training the model.

Once the channels have been selected, the model can be trained. It is an automated process and will begin when pressing **Train Model**. This is a background task and will take several minutes. A dialog opens that will display the progress of the model, and includes several parameters that can be monitored. These include model accuracy, loss, validation accuracy, validation loss, number of batches/ epochs, and total epochs. The total progress and elapsed time can also be monitored. When training is completed, a final dialog opens with options to name the trained model.

When the model is training, a general rule of thumb is that accuracy and validation accuracy should increase with each completed epoch, while loss and validation loss should decrease. If this behavior is not exhibited, the training session can be exited early by clicking cancel. This will end training and progress to the final dialog, which can be canceled from.

NOTE: Do not use the exit button to end the training session, as this will not close background tasks.





Visualization

Once the model has been trained, there are four ways to visualize the performance of the model. These are not the final indicator that a model is optimized, but they provide importance information at a glance. All of these visualization methods appear in the Analysis tab.

Deep-Learning Results

This is the result of a prediction using a sample of the data that the model was not trained on. The model creates a prediction for each timepoint, which is plotted in the temporal domain.

Visualization of Cross-Validation

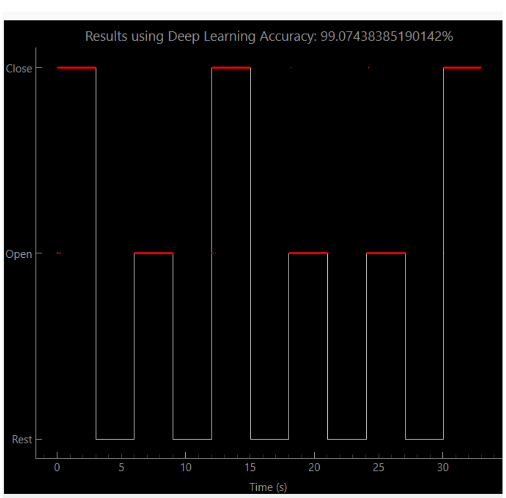
Plot of the accuracy and loss for training and testing data across all epochs. Accuracy should steadily improve while loss declines. If loss levels off at a value higher than 0.2, while accuracy improves, the model may be overfitting.

Normalized Confusion Matrix

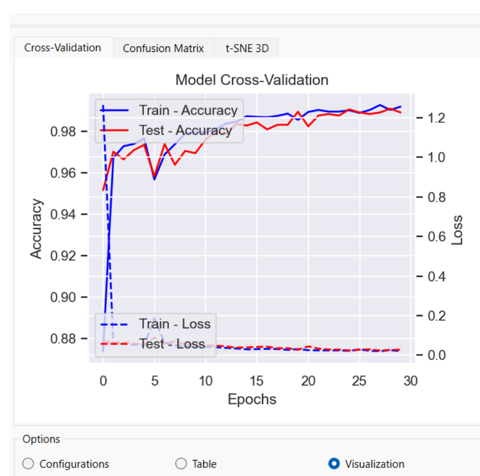
Prediction summary of sample set normalized across true labels. The diagonal represents the instances of correctly predicted labels. A perfect confusion matrix is represented by an identity matrix.

t-Distributed Stochastic Neighboring Embedding (tSNE)

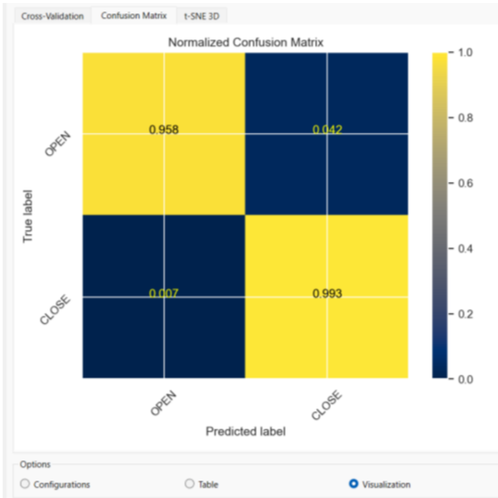
Non-linear dimensionality reduction algorithm used to explore and visualize the data. It reduces the dimensions of the data from 8 (i.e. the channels of the armband) into something observable by humans. This feature reduces the data to 3 dimensions and can be used to observe the formation of clusters, or regions, where predictions occur. Overlap in regions represent conditions where the model fails to distinguish between classes.



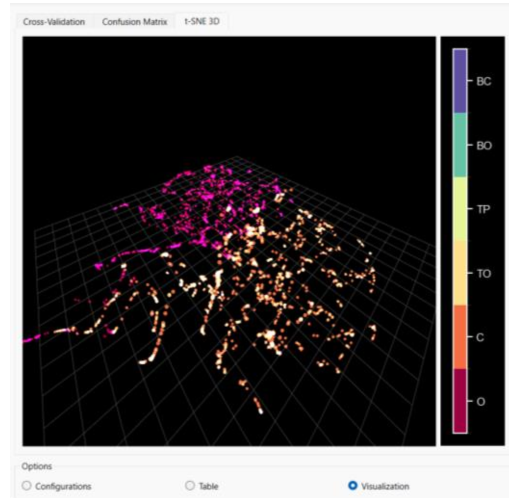
A)



B)



C)

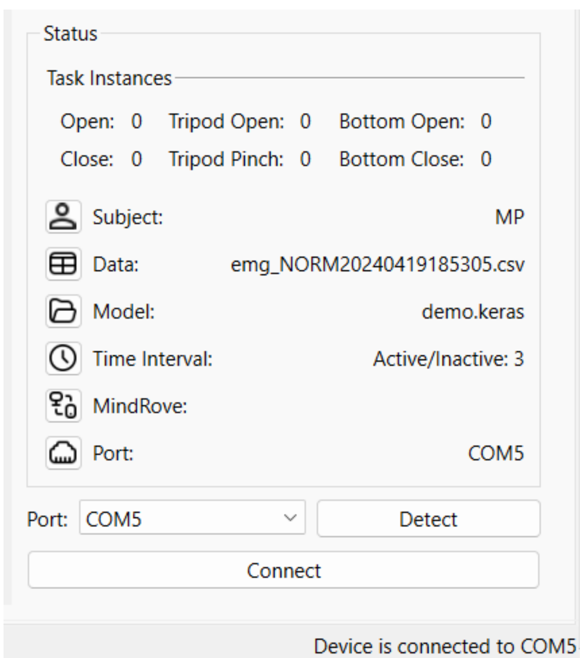


D)

Figure 10: Example visualization using a two-task sample. A) Results of Deep-Learning in Temporal Domain B) Cross-model validation with Accuracy and Loss C) Normalized Confusion Matrix D) 3D t-SNE

Training the Patient

The final task is to test the patient using the newly built model. This feature is still in development. The purpose is to test the model against the patient,



Function Reference

Collecting data

See **Collecting**, pg 8

Training model

See **Training the Model**, pg 10

Testing patient

This function is in development

Getting Started

This feature provides guidelines within the software for the patient to follow. It is found under the Help menu. This opens a separate window that explains the purpose of the system, shows the 6 tasks, provides a demo, and lets the patient practice on a preset task sequence. The user should guide and explain each slide that appears.

1. Title Window: Explains the purpose of the Intention Detection System and what the patient can expect to see and do.
2. Tasks: Let's the patient visualize the 6 tasks.
3. Demo: This slide provides a snapshot of the collecting phase in progress. There are labels so the patient knows what each part is.
4. Practice: This slide lets the patient perform against a preset task sequence. It is preset with the following sequence: 1, 3, 4, 2, 5, 6. It can be stopped early and reset if the patient wants to start over.

Assigning a new task sequence

A large task sequence can be easily created from the function New Task Sequence, found under Tools. The user can select the number of repetitions for each task. There is an option to shuffle the tasks; if this option is not selected, the tasks will appear in order of their id.

Assigning a custom task sequence

If the user wants full control of the task sequence, they should use the Create Custom Task Sequence function, found in Tools. The user can input tasks by the associated id, separated by a space.

Assigning a random task sequence of a defined length

A random prompt can be created from the Tools menu, Create Task Sequence>Randomized Task Sequence. To assign a length, select the option. This provides an option to select the length, up 30 tasks.

Assigning a task sequence of random length

A task sequence of a random order and length can be selected from the Tools menu, Create Task Sequence>Randomized Task Sequence. There is an option to assign length. This option can be ignored to assign a prompt of a random length.

Setting training classes

The training parameters are used to quickly acquire a task sequence. It is accessed from the Tools menu on the Menu bar. This opens a dialog with options to select the tasks. The number of repetitions can be choses, which is applied to all the selected tasks. Up to 30 repetitions can be made.

Setting time interval

The time interval during the collecting phase can be updated via two methods. The first is the Time icon from the Status panel. The second is from the menu bar under the Tools.

Clearing data

If the data is insufficient, it can be cleared from the File menu.

Clearing model

If the model is insufficient, it can be cleared from the File menu

Assigning channels

When data is initially collected, all 8 channels are displayed and selected to be fed into the model. If a channel has an insufficient EMG signal or there are significant outliers, it can be omitted from the training phase. Ideally, all 8 channels should be used, because subsequent sessions using the model will need to have the same configurated channels. For example, a model trained on a data set omitting channel 7, it can no longer be used.

Displaying raw data

The raw data can be displayed by selecting the option in the Datatype panel on the Analysis tab. It is filtered but unprocessed.

Displaying rectified data

Rectified data can be displayed by selecting the option in the Datatype panel on the Analysis tab.

Displaying normalized data

Normalized data can be displayed by selecting the option in the Datatype panel on the Analysis tab. This is the processed data that is used to train the model.

Connecting to LabView backend

The HoH is connected via COM port and needs to be connected before the system can relay messages to the HoH. Ports must be detected before the system can connect. This is done by plugging the microcontroller then pressing the detect button in the status panel. The port can then be selected the dropdown menu. The connect button will connect the system with the HoH.

Send Feedback

A Google Form providing feedback can be found under the **Help** menu. This can be used to report issues, bugs, or provide suggestions to improve the performance and functionality of the system.

Troubleshooting

- If no data is collected from the MindRove, make sure it is connected via WiFi. Otherwise, turn the device off and on again.

Future Iterations

Many features that would improve the experience have not been implemented yet. In subsequent versions of the system, the following will be included:

- Interrupt during the collection phase
- Synchronous timing between the task countdown and display
- Robust filtering that removes outliers and spikes in EMG
- Functions to save visuals as PNG, jpeg, or another format
- Mode selection that provides different prompts to train the patient

Resources

The following links provide the documentation for the MindRove Armband.

1. https://mindrove.com/armband/
2.https://docs.mindrove.com/Troubleshoot.html