AI-ASSISTED BRAIN COMPUTER INTERFACE FOR EEG-TO-TEXT COMMUNICATION

FYP GROUP : 43311

MUJTABA ABBAS (21034)
MEHAK SATTAR (21019)
AREEB ARFI (21022)
M. SAMEED (21029)

CONTENTS

INTRODUCTION

Communication is a fundamental aspect of human interaction, enabling individuals to express thoughts, emotions, and needs.

Advancements in neuroscience and artificial intelligence have opened new possibilities for assistive technologies, particularly in brain-computer interfaces (BCIs).

By translating neural signals into meaningful outputs, BCIs offer a revolutionary approach to bridging communication gaps for non-verbal individuals.

PROBLEM STATEMENT

Millions of individuals around the world face severe communication barriers due to various neurological or physical conditions. Existing assistive technologies are often costly, lack adaptability, and are not accessible to many users, limiting their real-world impact.

POSSIBLE SOLUTIONS

TRADITIONAL METHODS

Picture Exchange Communication Systems (PECS)

SCREEN ORIENTED DEVICE

Speech-Generating Devices (SGDs)

BCI DEVICES USING EEG

Electroencephalogram Based Systems

PROPOSED SOLUTION

The objective of this project is to develop a real-time, user-friendly communication system for individuals with communication barriers, utilising a wearable EEG headset. The system will capture brain signals and process them using AI models trained to classify common needs and emotions. The interpreted outputs will be displayed as words through a clean and intuitive interface.

OBJECTIVES

- Utilise a compact and efficient wearable EEG device to capture brain signals in real-time with good accuracy.
- Ensure the device is user-friendly, comfortable, and suitable for prolonged use.
- Design an interactive interface that translates processed EEG data into meaningful words.
- Integrate machine learning algorithms to enhance the accuracy of brain signal interpretation.

LITERATURE REVIEW

Chisco: An EEG-based BCI dataset for decoding of imagined speech

Zihan Zhang, Xiao Ding, Yu Bao, Yi Zhao, Xia Liang, Bing Qin & Ting Liu

Scientific Data **11**, Article number: 1265 (2024) | Cite this article

9773 Accesses | **6** Citations | Metrics

This study employs the g.tec Unicorn Hybrid Black+ headset to record EEG signals using eight-channel electrodes at a 250 Hz sampling frequency. Standard electrode positions were used, and properly fitted headcaps ensured accurate data collection. The recorded signals were then processed and analyzed using deep learning techniques for improved classification accuracy. **[2]**

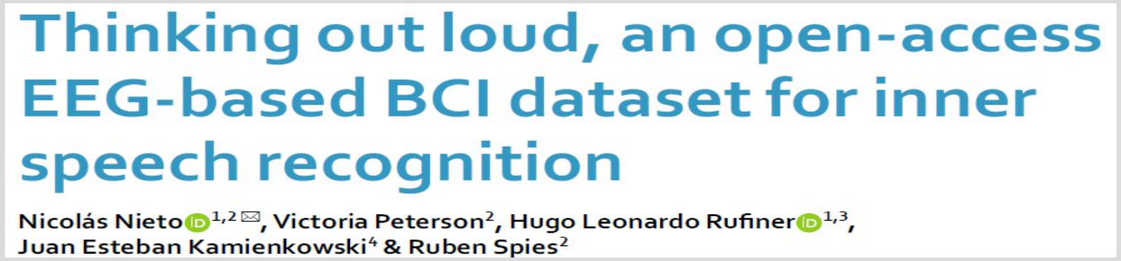
Imagined Speech Classification Using EEG and Deep Learning

by Mokhles M. Abdulghani, Wilbur L. Walters and Khalid H. Abed *

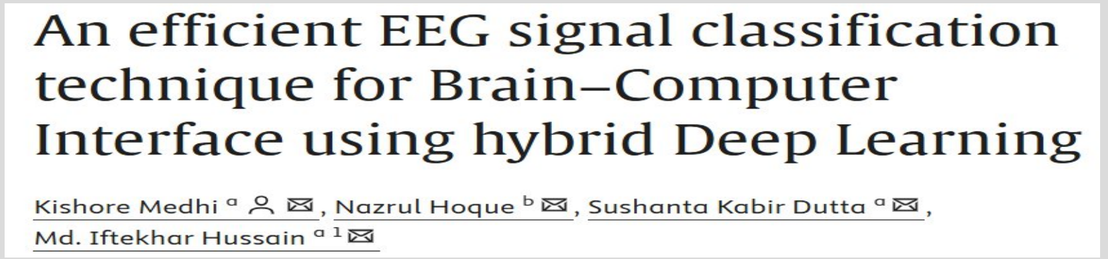
Department of Electrical & Computer Engineering and Computer Science, College of Sciences, Engineering & Technology, Jackson State University, Jackson, MS 39217, USA

BCIs enable communication by decoding neural activity into commands. Advances in deep learning have improved naturalistic paradigm BCIs, making them more user-friendly. While non-invasive imagined speech decoding holds great potential, progress is limited due to dataset scarcity and experimental challenges. **[1]**

LITERATURE REVIEW (CONT.)



The paper introduces a publicly available EEG dataset specifically designed for inner speech recognition (e.g., silently thinking words without speaking). It includes recordings from 15 participants imagining 6 different words. The dataset was collected using a 64-channel EEG system and is intended to support the development of brain-computer interface (BCI) applications that decode imagined speech. This resource enables researchers to explore and benchmark machine learning models for non-invasive neural decoding tasks. **[3]**



The system enhances the accuracy and real-time performance of brain-computer interfaces (BCIs) by integrating advanced signal processing techniques within the headset. This ensures precise brain signal detection while minimizing interference and noise, leading to more reliable communication for non-verbal individuals. By optimizing real-time processing, the system enables faster and more efficient translation of EEG signals into meaningful outputs, improving usability and overall effectiveness. **[4]**

METHODOLOGY

1

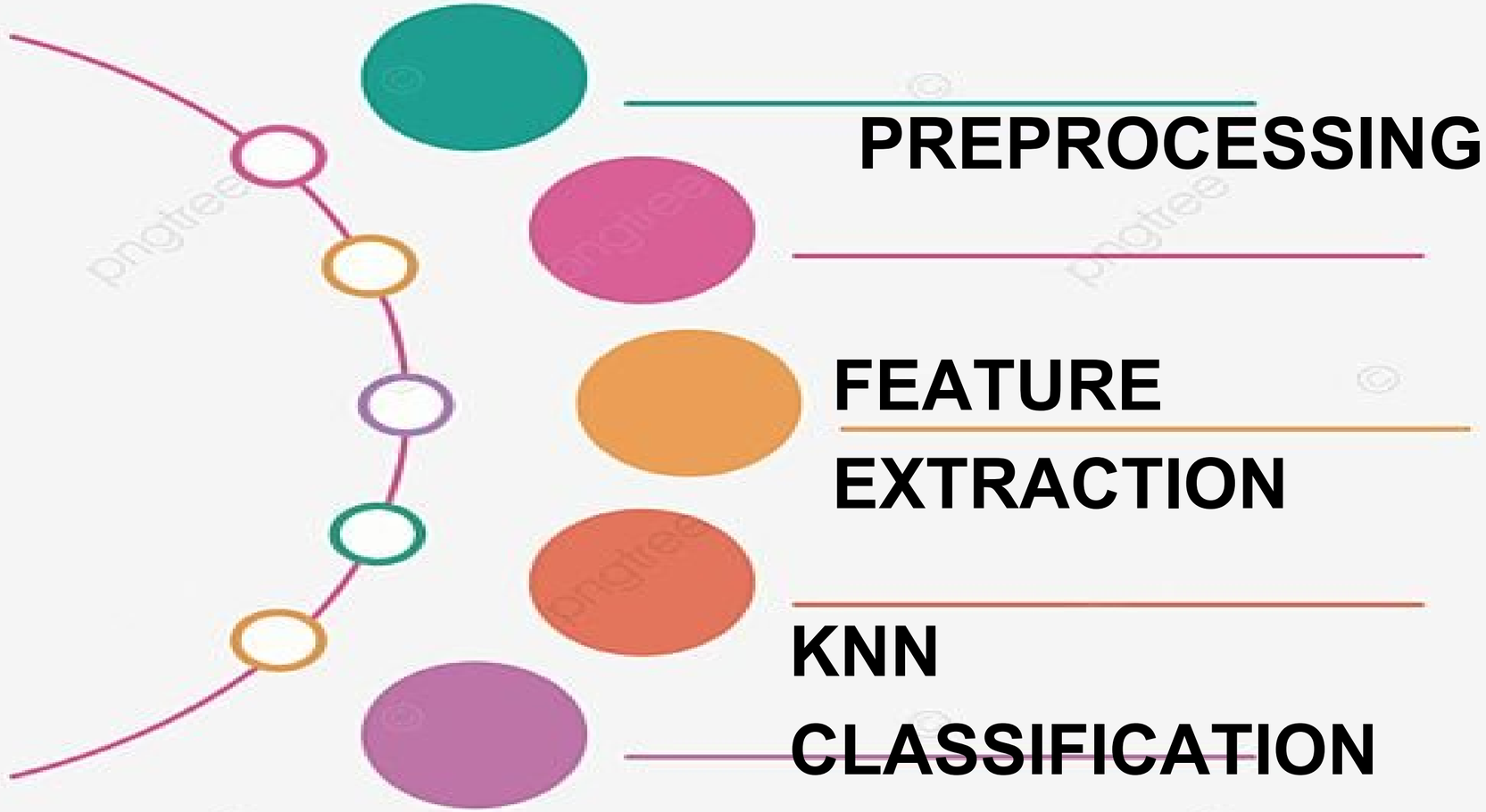
SYSTEM MODEL

2

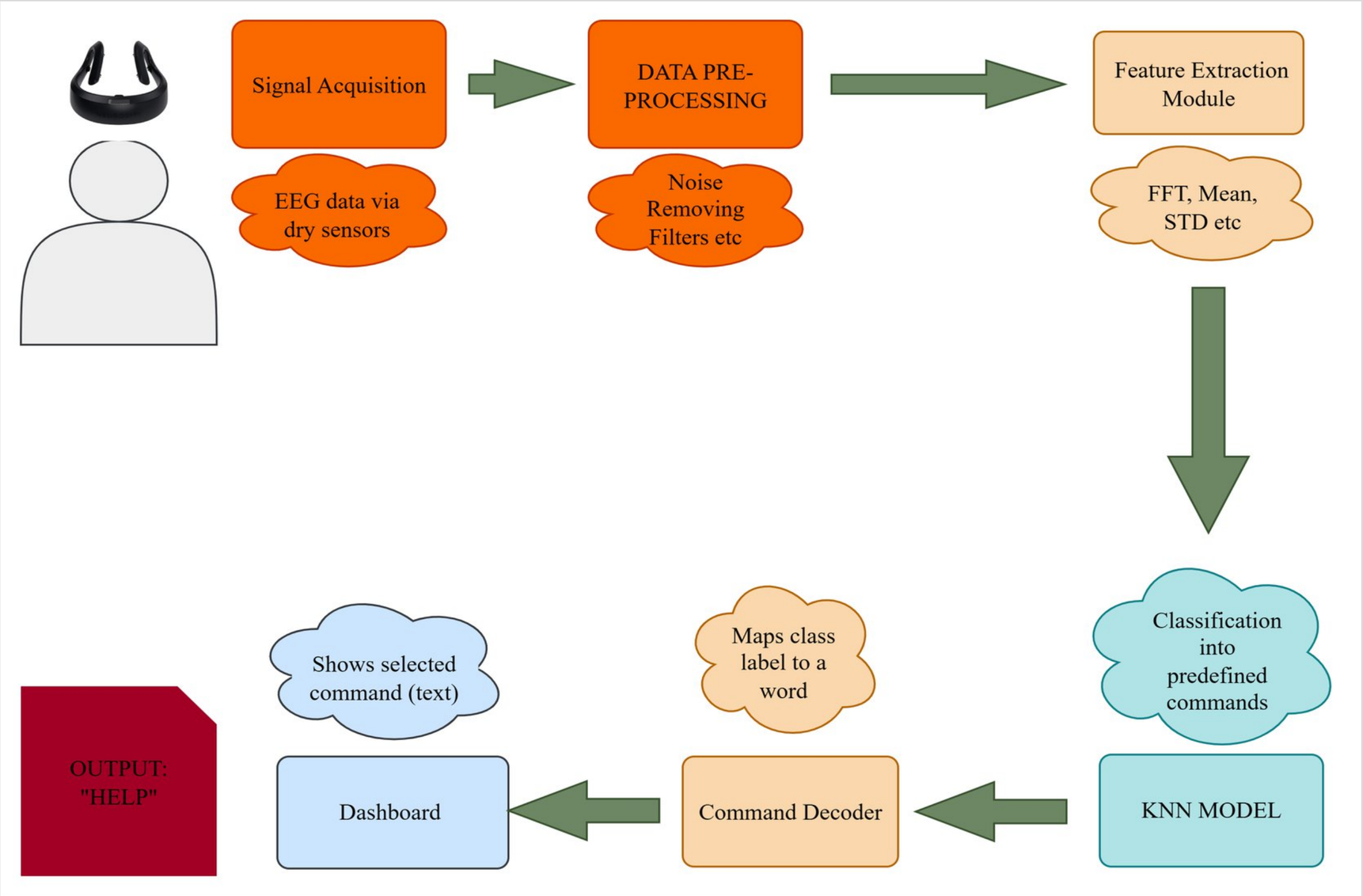
HARDWARE & SOFTWARE

3

SYSTEM ANALYSIS


EEG DATA
PREPROCESSING
FEATURE
EXTRACTION
KNN
CLASSIFICATION
OUTPUT

SYSTEM BLOCK DIAGRAM



EEG FREQUENCY BANDS

1. **Alpha Band (8–13 Hz)**: Represents a relaxed, calm state.
2. **Beta Band (13–30 Hz)**: Represents an alert, active thinking state.
3. **Gamma Band (30–50 Hz)**: Represents high cognitive processing states.

Brainwave Type	Frequency range	Mental states and conditions
Delta	0.1Hz to 3Hz	Deep, dreamless sleep, non-REM sleep, unconscious
Theta	4Hz to 7Hz	Intuitive, creative, recall, fantasy, imaginary, dream
Alpha	8Hz to 12Hz	Relaxed, but not drowsy, tranquil, conscious
Low Beta	12Hz to 15Hz	Formerly SMR, relaxed yet focused, integrated
Midrange Beta	16Hz to 20Hz	Thinking, aware of self & surroundings
High Beta	21Hz to 30Hz	Alertness, agitation
Gamma	30Hz to 100Hz	Motor Functions, higher mental activity

Table 1: Frequencies and Significance of Brainwaves

HARDWARE SELECTION

EMOTIV INSIGHT

OR

NEUROSITY CROWN?

WHY NOT EMOTIV INSIGHT?

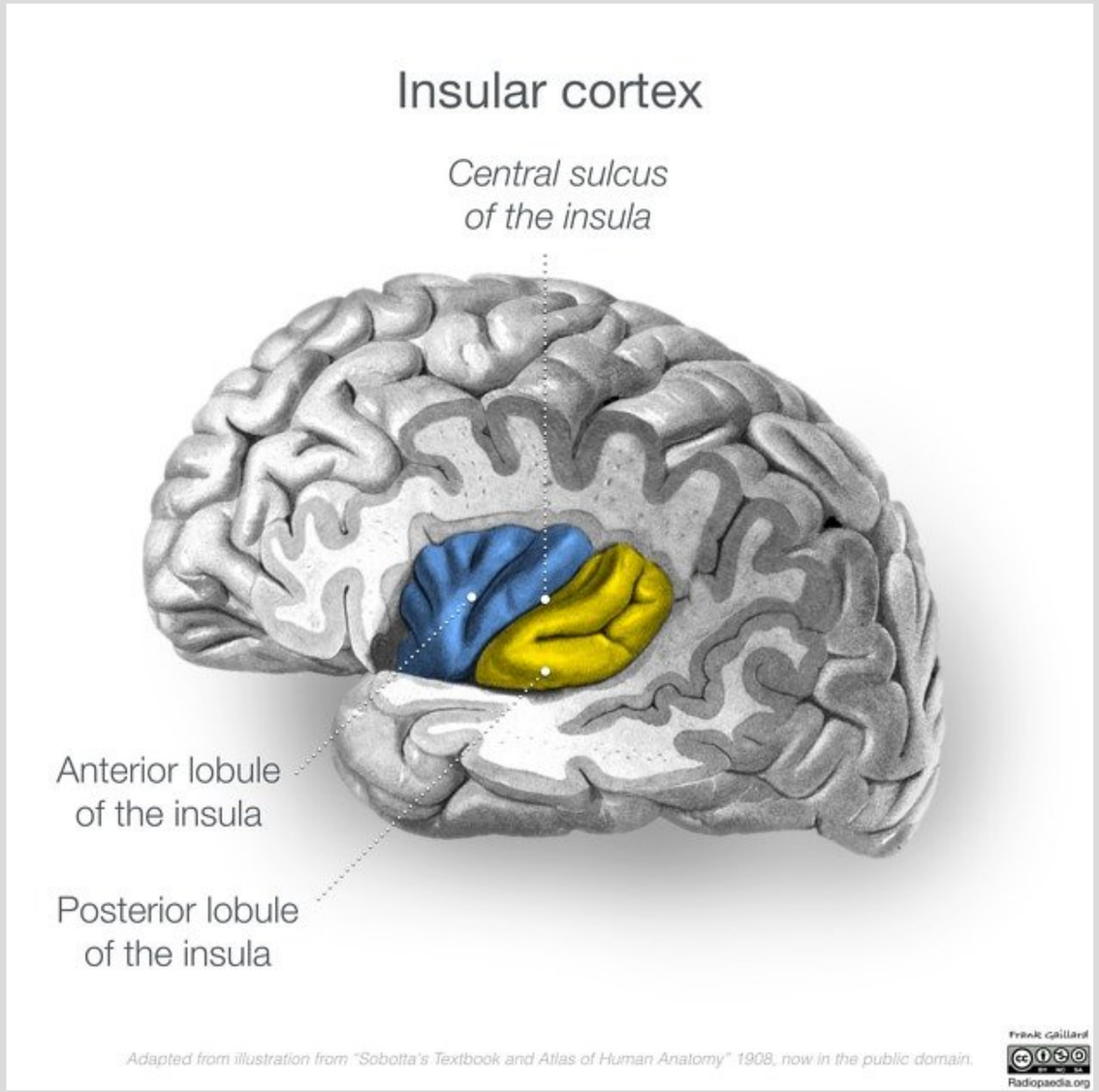
- 🔒 Restricted Raw Data Access – Requires paid SDK to access full EEG signals.
- □ Fewer EEG Channels – Only 5 sensors, leading to lower spatial resolution.
- ⚙ Limited Developer Flexibility – Not ideal for custom AI/ML pipelines.
- 🔄 Lower Signal Sampling Rate – Affects accuracy in real-time applications.
- 🚫 Not Targeted for BCI Applications – More focused on wellness and meditation use.
- 🔍 No Electrode at Left Anterior Insula – Missing coverage of key region for imagined speech.

REASONS TO CHOOSE NEUROSITY CROWN

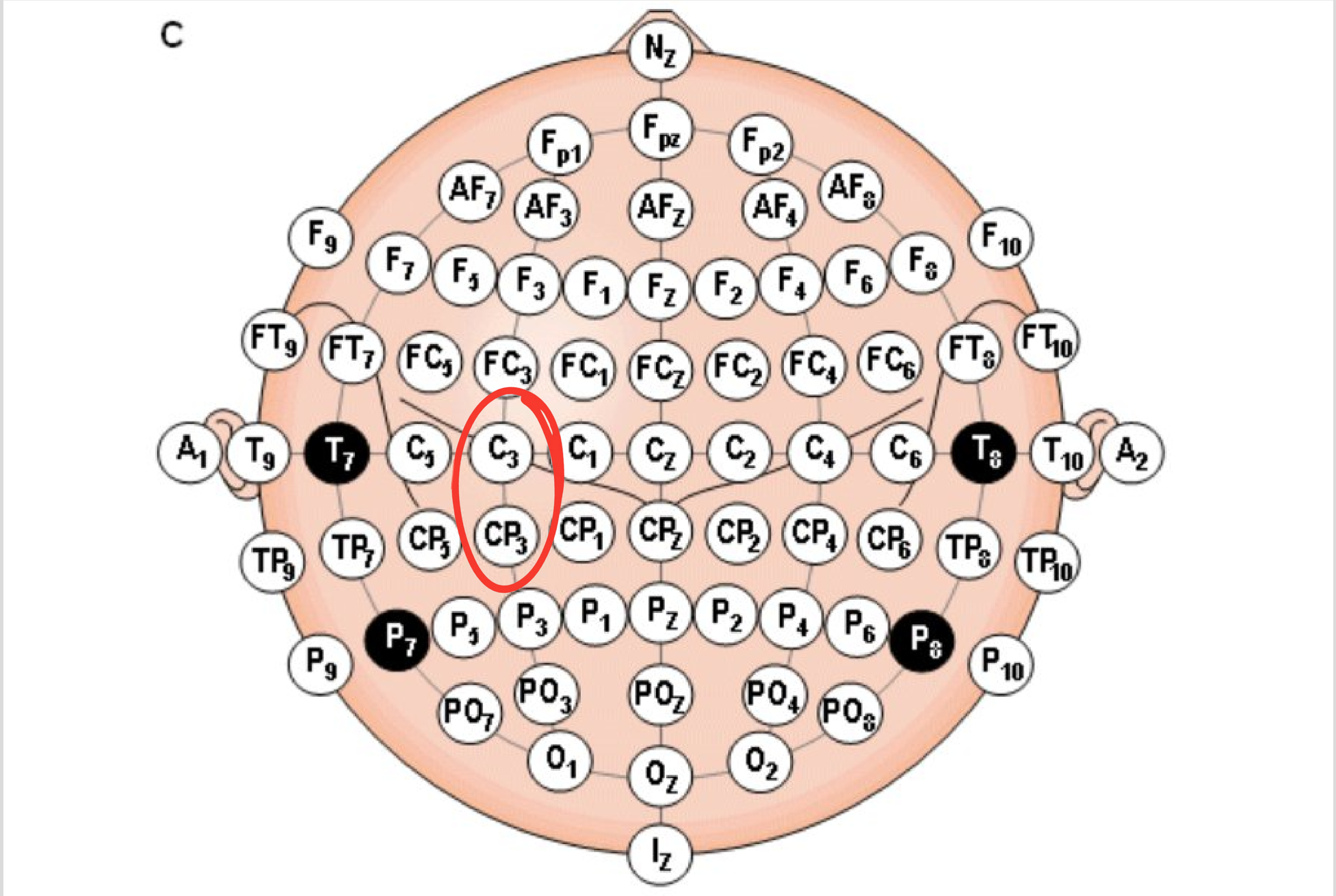
- 🧠 8-channel dry electrode EEG headset for real-time brainwave monitoring
- ⚡ 256 Hz sampling rate with 24-bit ADC ensures high signal quality
- 📶 Wi-Fi & Bluetooth connectivity for seamless data streaming
- 🔋 1-2 hours battery life, lightweight and ergonomic for long wear
- □ Covers motor cortex and frontal regions for mental command detection
- □ Developer-friendly with Python, JavaScript SDKs and cloud APIs
- 🔐 Secure login & personalised user profiles via Neurosity Cloud

THE LEFT ANTERIOR INSULA(LAI)

The Left Anterior Insula (LAI) is essential for emotion processing, interoception, and decision-making, making it crucial for our EEG-based nonverbal communication project. It helps recognize feelings like hunger, thirst, and discomfort while working with the frontal lobe to reinforce intent. Integrating LAI signals into our system enhances emotion recognition, intent detection, and word prediction accuracy, creating a more intuitive and responsive communication interface.



10-20 SYSTEM


C
NZ
FP1
FPZ
FP2
AF7
AF3
AFZ
AF4
AF8
F9
F7
F5
F3
F1
FZ
F2
F4
F6
F8
F10
FT9
FT7
FC5
FC3
FC1
FCZ
FC2
FC4
FC6
FT8
FT10
A1
T9
T7
C5
C3
C1
CZ
C2
C4
C6
T8
T10
A2
TP9
TP7
CP5
CP3
CP1
CPZ
CP2
CP4
CP6
TP8
TP10
P9
P7
P5
P3
P1
PZ
P2
P4
P6
P8
P10
PO7
PO3
POZ
PO4
PO8
O1
OZ
O2
IZ

DATA PLOTTING

Brainwaves

View Type

Raw Data

Data Points

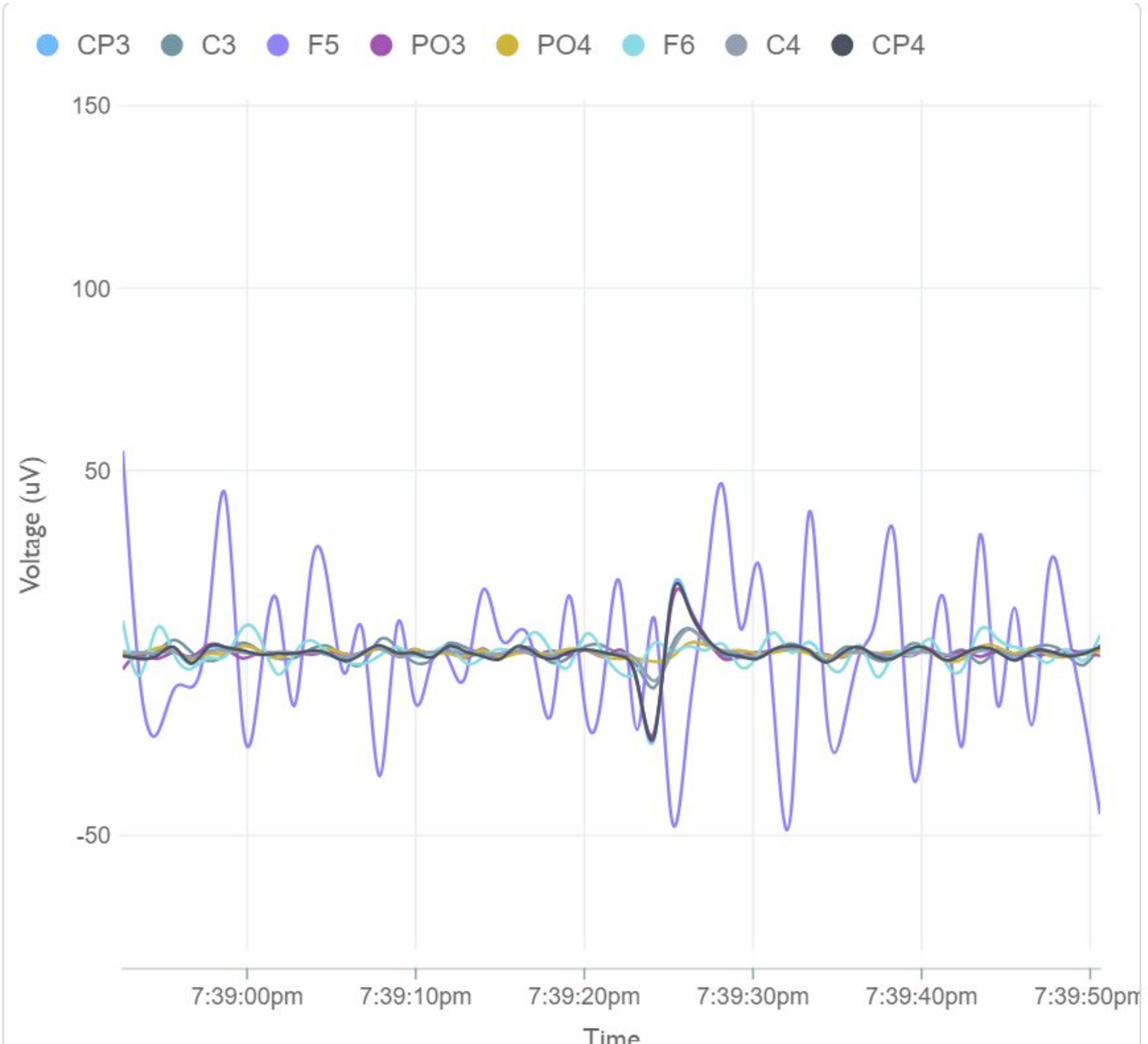
Average data points to seconds

Channels

CP3 C3 F5 PO3 PO4 F6 C4 CP4

Notch Filter

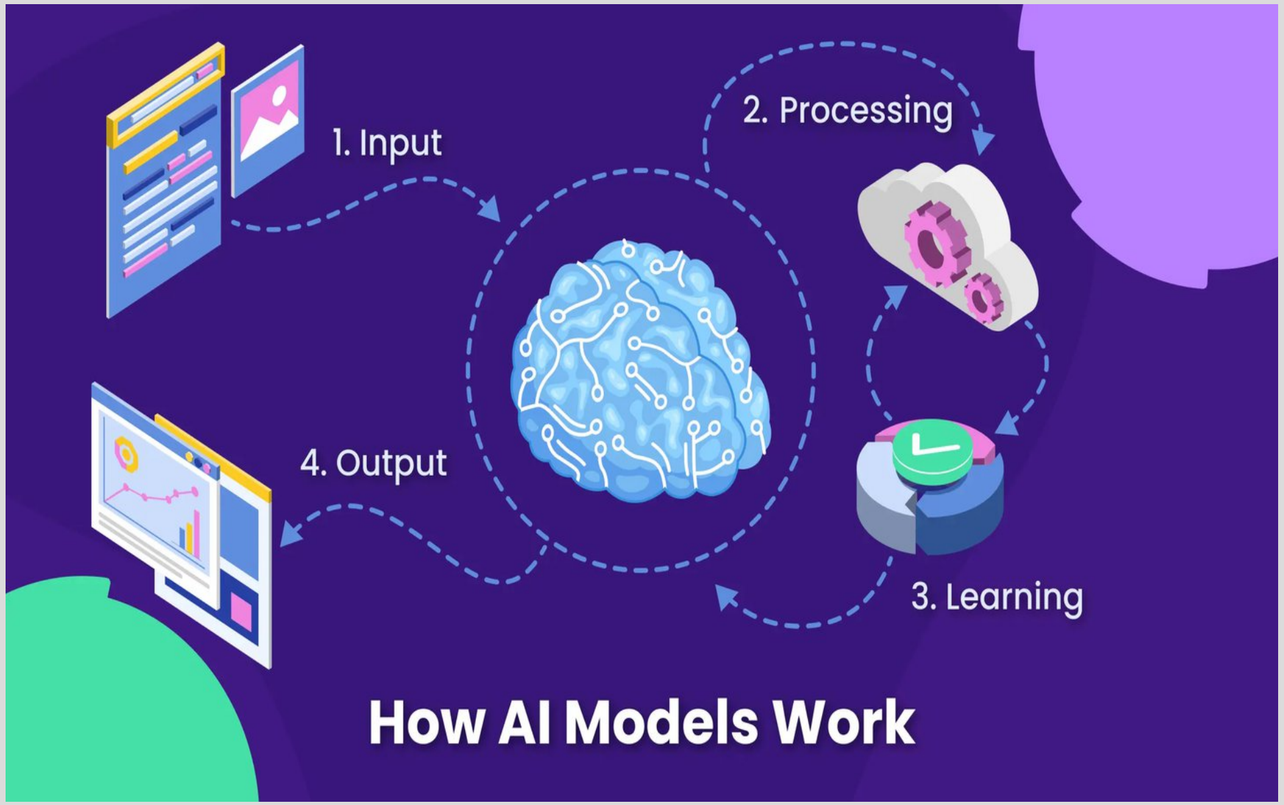
Bandpass Filter



SOFTWARE
VISUAL STUDIO CODE

Visual Studio Code (VS Code) is a lightweight yet powerful source code editor developed by Microsoft. It supports a wide range of programming languages and offers features like debugging, version control, extensions, and intelligent code completion. VS Code is ideal for developing web applications, including React and TypeScript dashboards.

AI MODELS



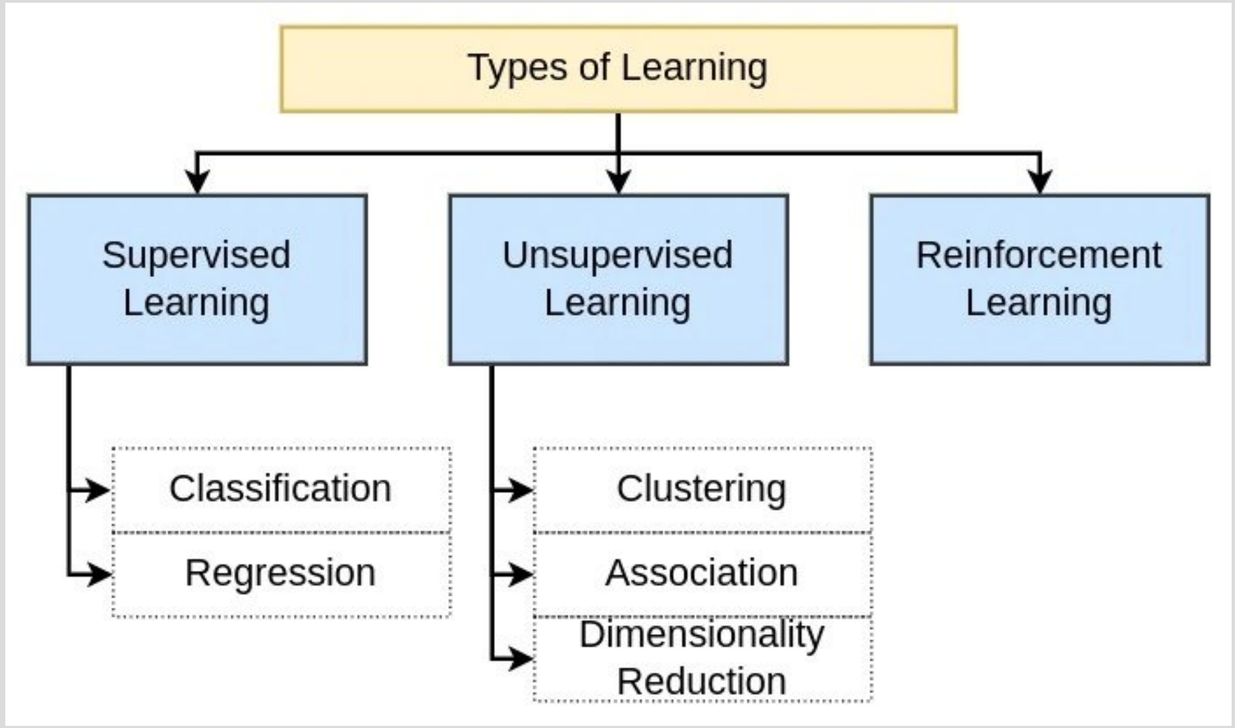
- AI models learn patterns from data to make intelligent decisions
- Used in fields like healthcare, robotics, finance, and brain-computer interfaces
- Divided into categories:
- 🎯 Traditional Machine Learning (e.g., KNN, SVM, Random Forest)
- 🧠 Deep Learning (e.g., CNN for images, LSTM for sequences)

LEARNING TYPES

- Supervised learns from labelled data
- Unsupervised finds hidden patterns in unlabeled data
- Reinforcement learns by rewards and penalties

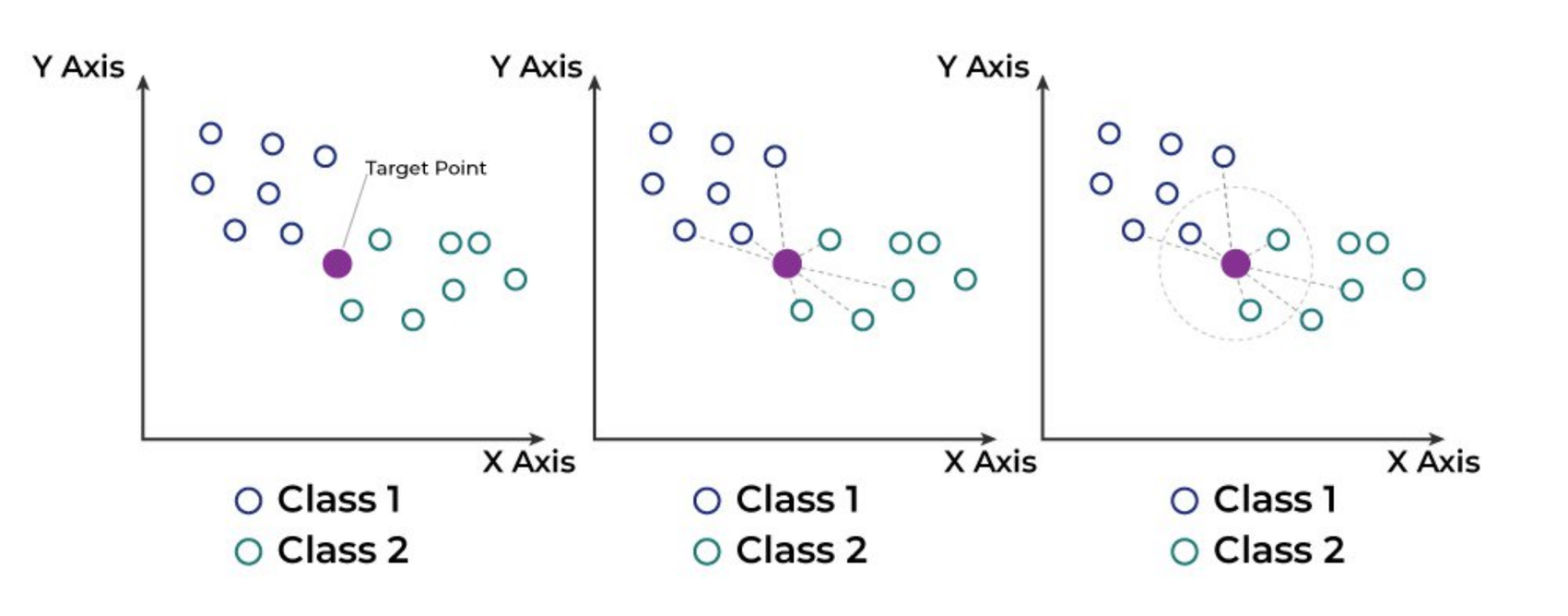
In this project, we used a KNN model to classify EEG brain signals into commands

AI enables real-time interpretation of brain activity, enhancing communication and control

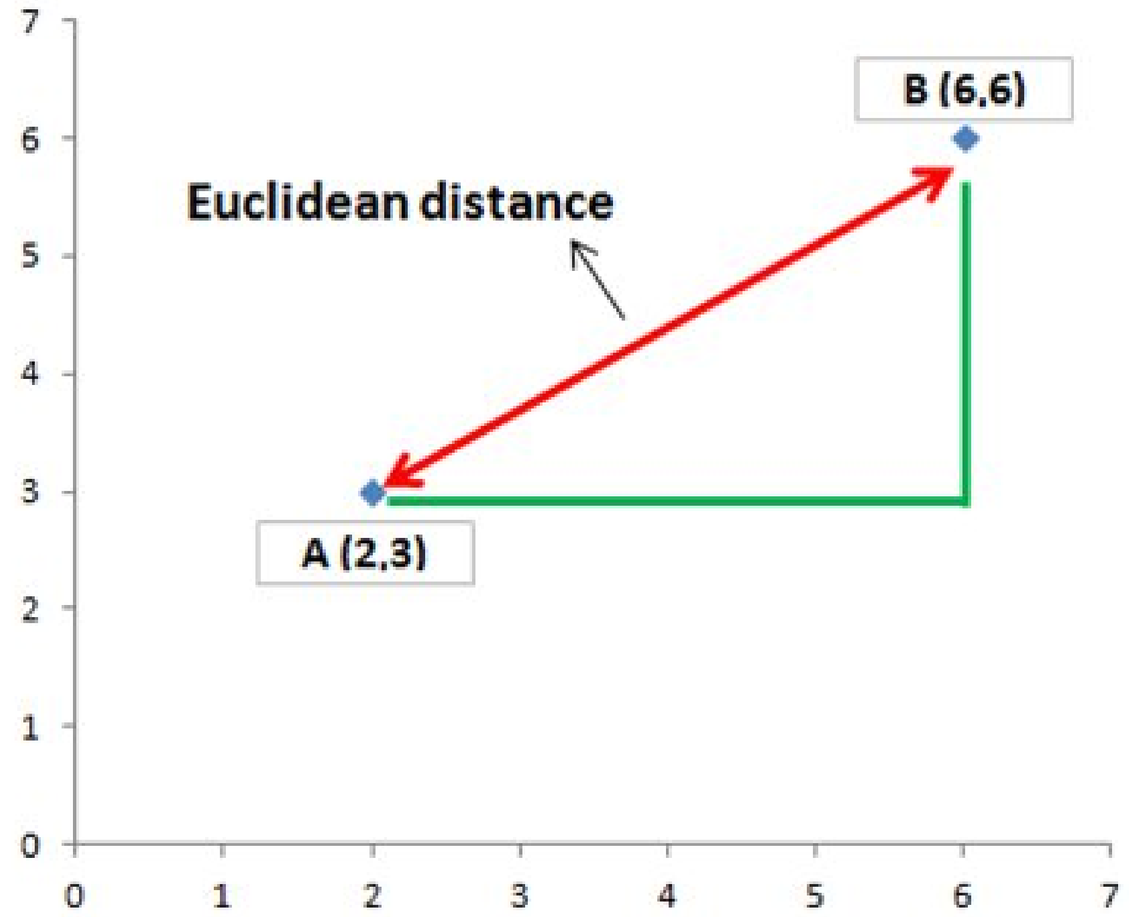


KNN CLASSIFICATION MODEL

- KNN compares the input feature vector with the training samples
- Finds K=X nearest neighbours using Euclidean distance
- Chooses the most common class among them

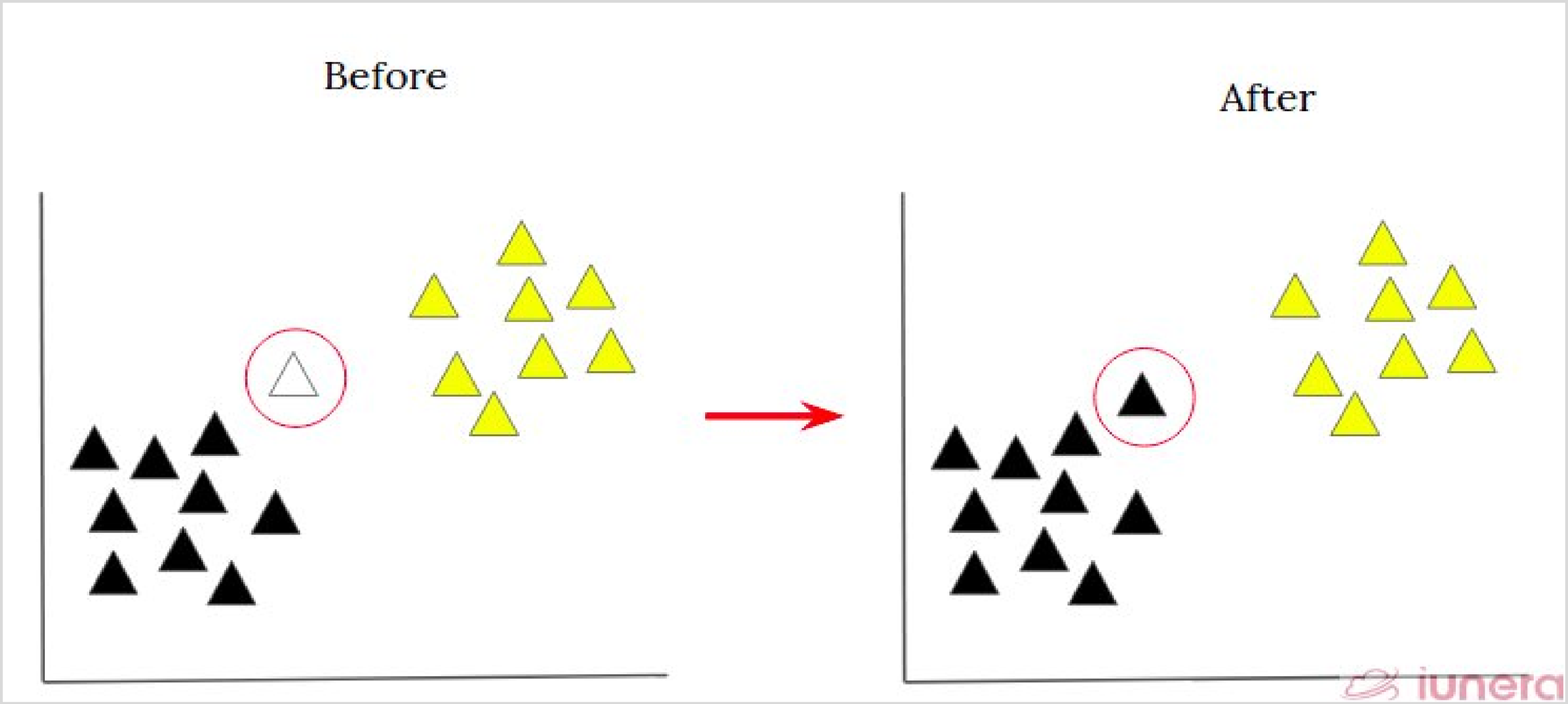


EUCLIDEAN DISTANCE

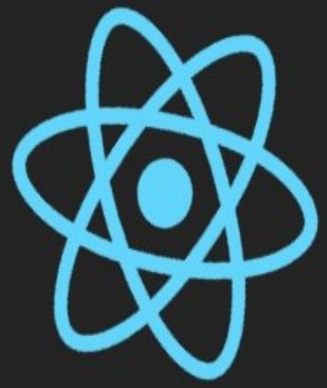


$$Euclidean\ distance\ (a, b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$$

KNN PREDICTION



DASHBOARD FRONT-END & BACK-END



TypeScript

TYPESCRIPT+ VITE+REACT

RESULTS AND DISCUSSION

- **SIMULATION RESULTS**
- **EXPERIMENTAL RESULTS**
- **COMPARISON**

SIMULATION: MODEL TRAINING AND TESTING

- Considered four models: CNN, LSTM, RF and KNN on the EEG dataset (10 words)
- Data shape: 70 samples per word (user-specific)
- Simulation Goal: Identify the best model for real-time imagined speech classification
- Models evaluated via training accuracy, validation performance, and generalisation.

PYTHON LIBRARIES

LIBRARY	PURPOSE
1. neurosity	EEG data streaming from Neurosity headset.
2. joblib	Load trained KNN model
3. numpy	Numerical operations, array handling
4. scipy.fft	Frequency-domain feature extraction (FFT)
5. dotenv, os	Secure handling of device credentials

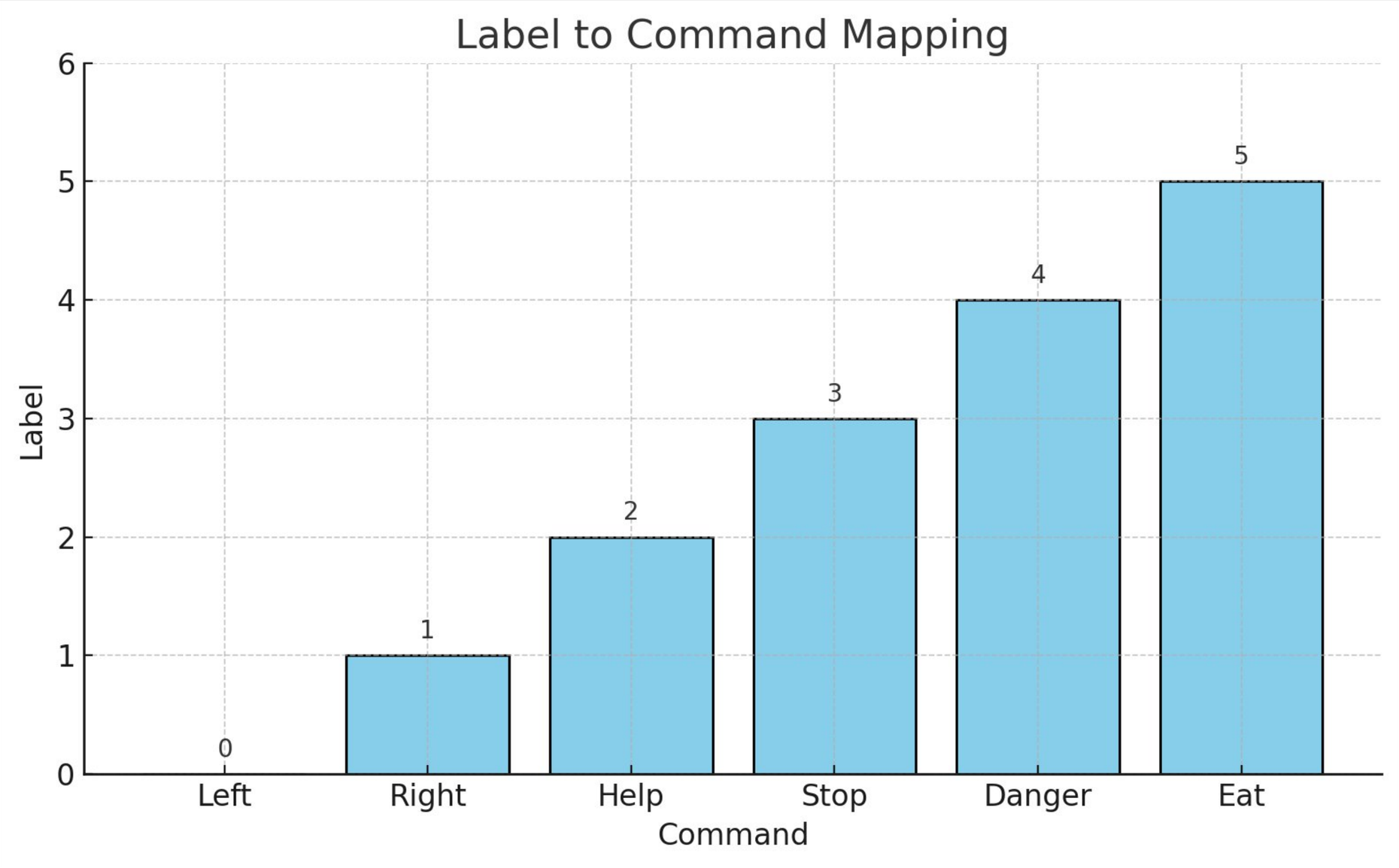
KNN
BEST FOR LABELED DATA

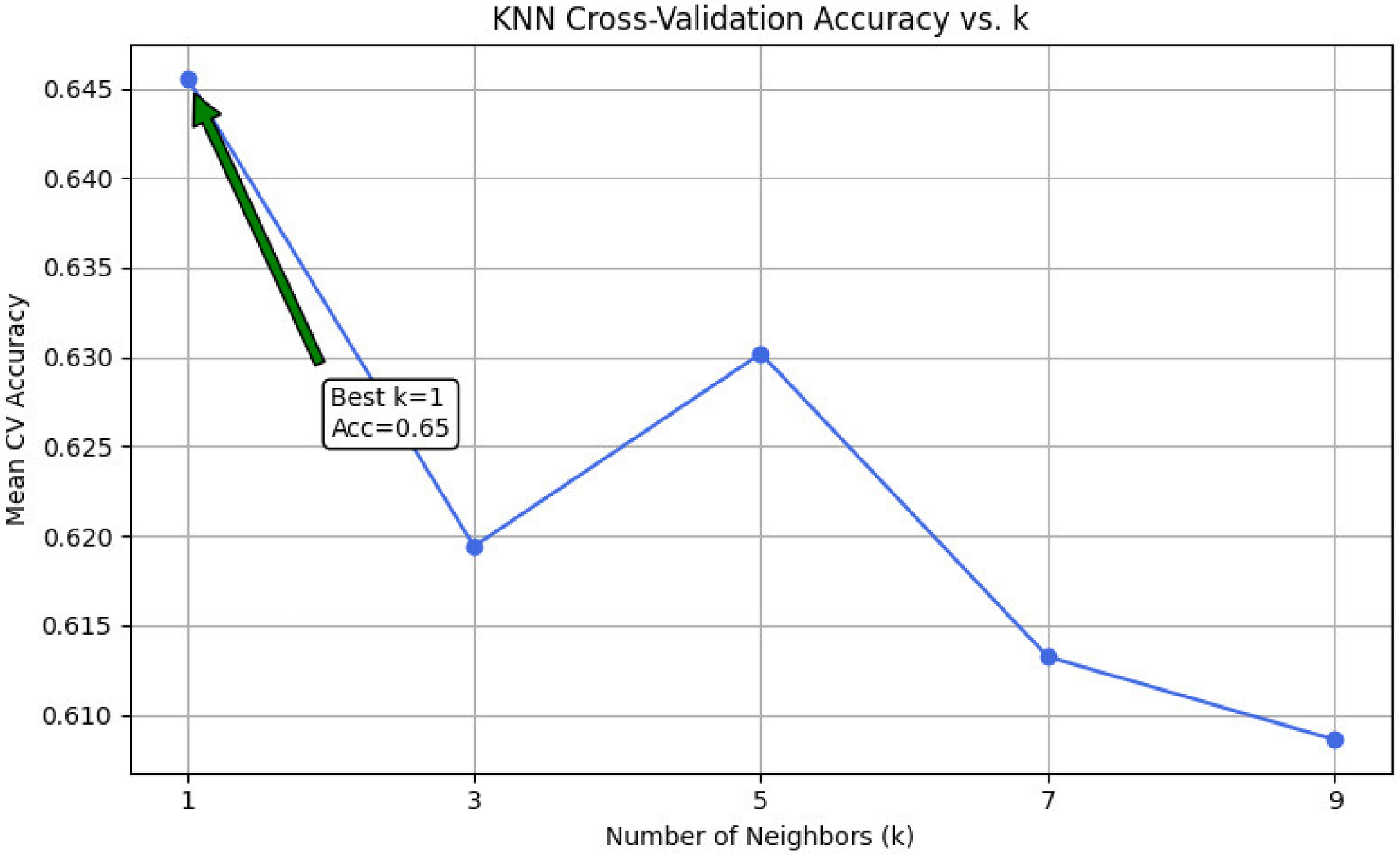
- Trained on a 10-word dataset with 70 samples/word
- Simpler, non-parametric model
- Fast, lightweight, and robust to overfitting
- Great performance with small, personalised datasets

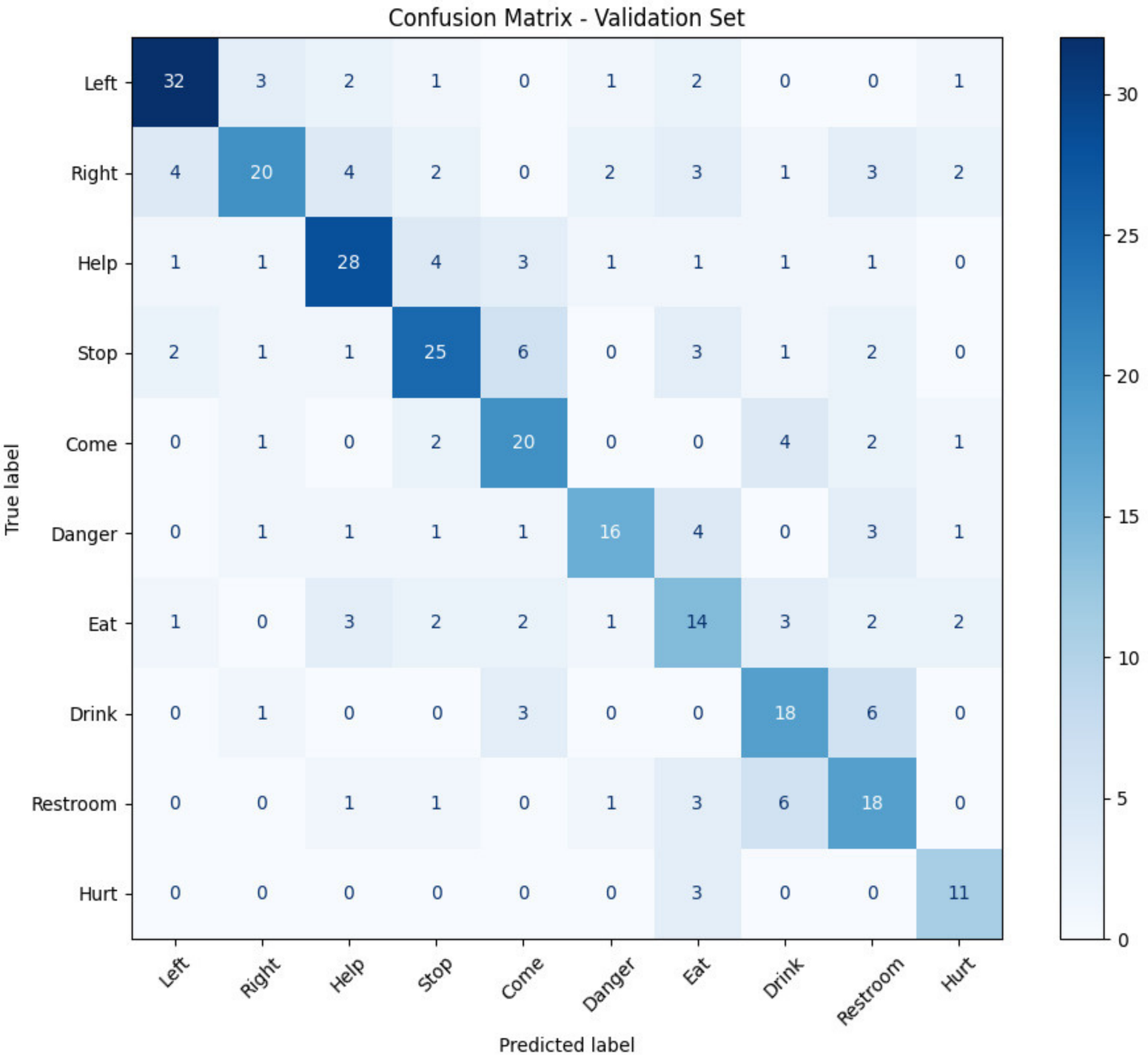
REAL-TIME TESTING WITH NEUROSITY CROWN

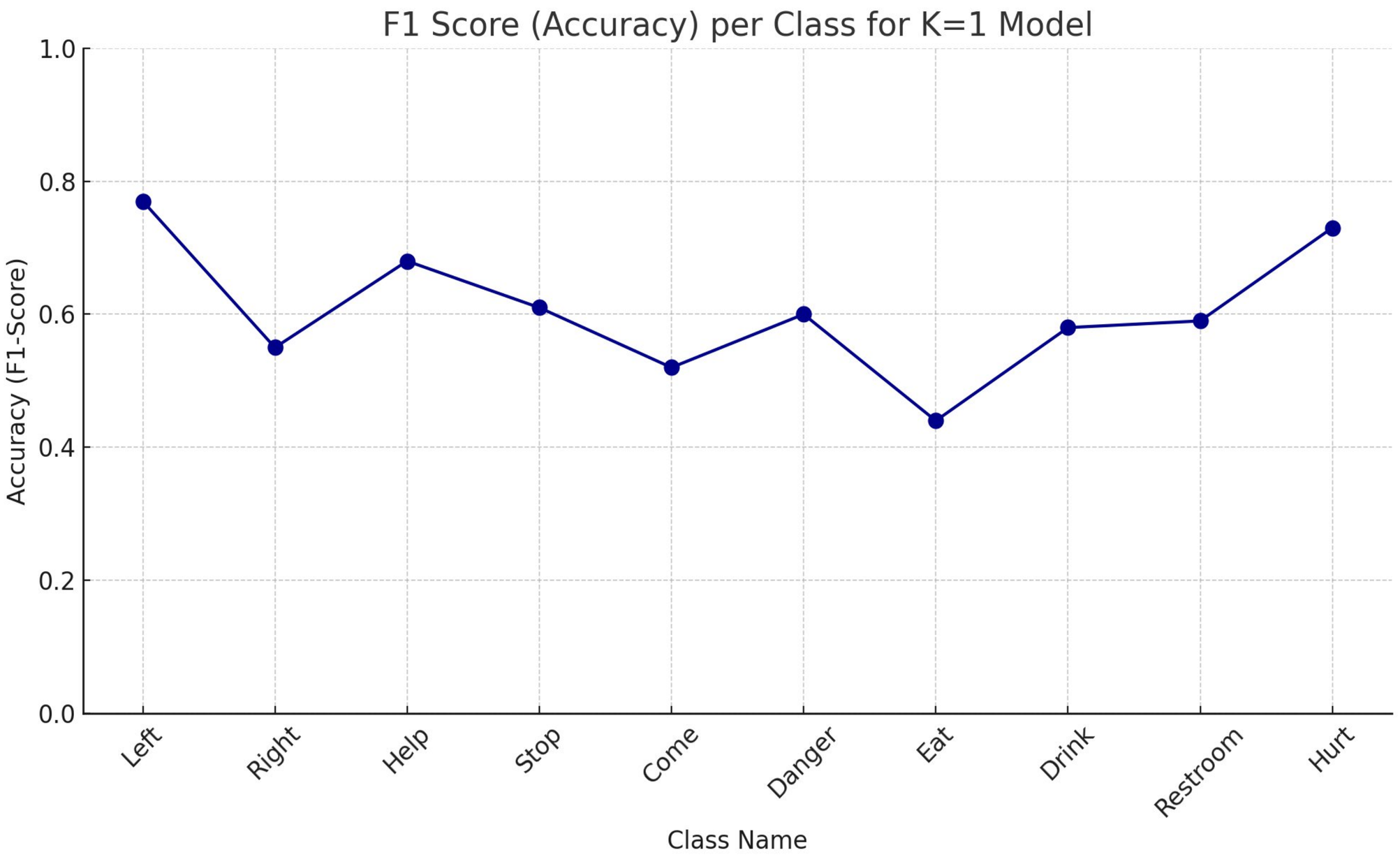
- Integrated model with live EEG stream (via Neurosity SDK)
- Used TypeScript + React frontend for live prediction display
- KNN achieved accurate and consistent predictions in real-time
- UI successfully mapped brain signals to 10 commands with low latency

CLASS LABELS

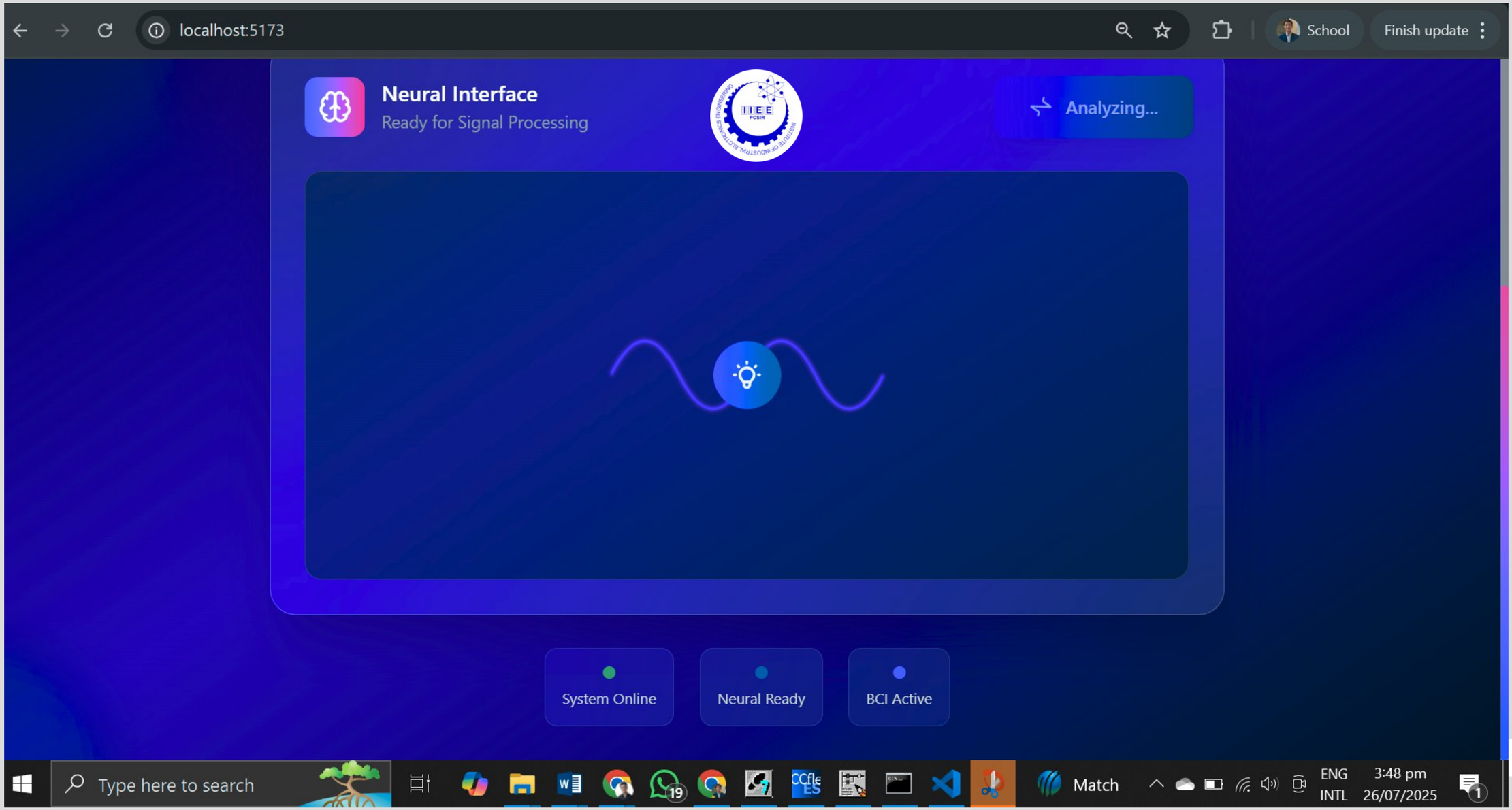



KNN Cross-Validation Accuracy vs. k
Mean CV Accuracy
0.645
0.640
0.635
0.630
0.625
0.620
0.615
0.610
Best k=1
Acc=0.65
1
3
5
7
9
Number of Neighbors (k)


Confusion Matrix - Validation Set
True label
Predicted label
Left
Right
Help
Stop
Come
Danger
Eat
Drink
Restroom
Hurt
32 3 2 1 0 1 2 0 0 1
4 20 4 2 0 2 3 1 3 2
1 1 28 4 3 1 1 1 1 0
2 1 1 25 6 0 3 1 2 0
0 1 0 2 20 0 0 4 2 1
0 1 1 1 1 16 4 0 3 1
1 0 3 2 2 1 14 3 2 2
0 1 0 0 3 0 0 18 6 0
0 0 1 1 0 1 3 6 18 0
0 0 0 0 0 0 3 0 0 11
30
25
20
15
10
5
0


F1 Score (Accuracy) per Class for K=1 Model
Accuracy (F1-Score)
1.0
0.8
0.6
0.4
0.2
0.0
Left
Right
Help
Stop
Come
Danger
Eat
Drink
Restroom
Hurt
Class Name

MODEL DASHBOARD



FINAL THOUGHTS ON RESULTS

- Accuracy was average because imagined words have weak and noisy brain signals.
- KNN worked by comparing patterns using the distance between signals.
- RF can be used for better accuracy with a bigger dataset.
- The system can be improved with additional training and more effective signal filtering. (OpenBCI)

CONCLUSION & RECOMMENDATIONS

- Developed a BCI system using Neurosity Crown and a K-Nearest Neighbours (KNN) model.
- Successfully classified EEG signals into 10 imagined commands in real-time.
- Demonstrated the potential of combining AI with neurotechnology for communication assistance.
- Model performance was limited by the small, internally collected dataset.
- Recommend using a larger, external dataset to enable advanced models (e.g., CNN, LSTM) and improve accuracy.

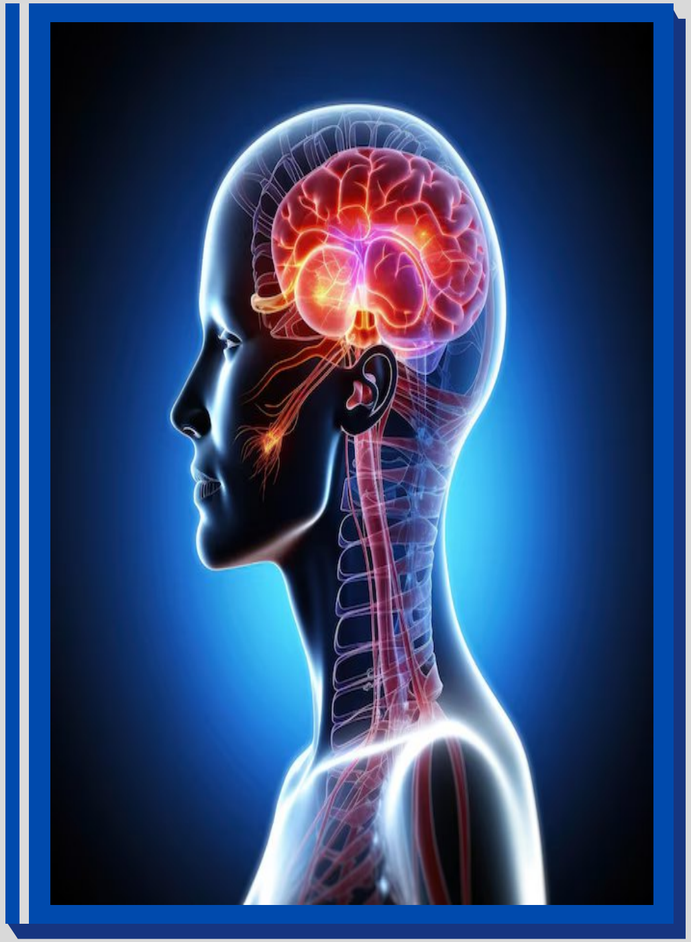
REFERENCES

1. Medhi, K., Hoque, N., Dutta, S. K., & Hussain, M. I. (2022). An efficient EEG signal classification technique for Brain–Computer Interface using hybrid Deep Learning. Biomedical Signal Processing and Control, 78, 104005.

2. Zhang, Z., Ding, X., Bao, Y., Zhao, Y., Liang, X., Qin, B., & Liu, T. (2024). Chisco: An EEG-based BCI dataset for decoding of imagined speech. Scientific Data, 11(1), 1265.

3. Nieto, N., Peterson, V., Rufiner, H. L., Kamienkowski, J. E., & Spies, R. (2022). Thinking out loud, an open-access EEG-based BCI dataset for inner speech recognition. Scientific data, 9(1), 52.

4. Medhi, K., Hoque, N., Dutta, S. K., & Hussain, M. I. (2022). An efficient EEG signal classification technique for Brain–Computer Interface using hybrid Deep Learning. Biomedical Signal Processing and Control, 78, 104005.



THANK**YOU**

CODE EXAMPLES FOR IN-DEPTH QUESTIONS

DATASET COLLECTION

main.py > ...

```python

# Stop the stream from the main thread (safe!)
unsubscribe()

buffer = buffer[:16]  # Only keep the first 16 events

output = {}
for i, event in enumerate(buffer):
    label = f"{int(i * 62.5)}ms"
    output[label] = event  # shape: 8 x 16 per event

# Save to files
with open("dataset/mujtaba/drink/mujtaba_drink45.json", "w") as f:
    json.dump(output, f, indent=2)

print("✅ EEG data saved with labeled time windows.")
```

```
(venv) C:\Users\Mujtaba\Desktop\neuro>venv\Scripts\deactivate
C:\Users\Mujtaba\Desktop\neuro> main.py
```

ACCURACY RESULTS PYTHON CODE

```python
accuracy = accuracy_score(y_val, y_pred)
print(f"\n🎯 Validation Accuracy with k={best_k}: {accuracy:.4f}")

print("\n📊 Classification Report:")
class_names = [
    "Left",
    "Right",
    "Help",
    "Stop",
    "Come",
    "Danger",
    "Eat",
    "Drink",
    "Restroom",
    "Hurt",
]
print(classification_report(y_val, y_pred, target_names=class_names))


print("\n🔍 Cross-validation results:")
for k in k_values:
    knn = KNeighborsClassifier(n_neighbors=k, metric="manhattan")
    scores = cross_val_score(knn, X_train, y_train, cv=5)
    mean_score = np.mean(scores)
    cv_scores.append(mean_score)
    print(f"  k={k}: mean accuracy = {mean_score:.4f}")
```

CODE FOR CONFUSION MATRIX PLOTTING (MATPLOTLIB)

```
# ── Plot and show confusion matrix ──
cm = confusion_matrix(y_val, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=class_names)
fig, ax = plt.subplots(figsize=(10, 8))
disp.plot(ax=ax, cmap="Blues", xticks_rotation=45)
plt.title("Confusion Matrix - Validation Set")
plt.tight_layout()
plt.savefig("confusion_matrix_validation.png")
plt.show()
print("📊 Confusion matrix saved as 'confusion_matrix_validation.png'")
```