iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: V Month of publication: May 2025

DOI: https://doi.org/10.22214/ijraset.2025.70214

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Mansha

Jitesh Kumar Sahu[1], Pranjal Mahapatra[2], K Yashwant Rao[3], Sparsh Verma[4], Eshank Sharma[5], Ms. Sonia Wadhwa[6]
Department Of Computer Science, Government Engineering College Bilaspur

Abstract: "Mansha" is an advanced Brain-Computer Interface (BCI) system that enables direct communication between the brain and external devices using EEG technology, signal processing, and machine learning. Aimed at enhancing accessibility and independence, especially for individuals with disabilities, it offers intuitive, hands-free control by translating neural signals into actionable commands. The project prioritizes signal accuracy, processing speed, and adaptability, with applications ranging from assistive technologies and smart home automation to robotic control and neurorehabilitation. By addressing key challenges in BCI—such as signal variability and real-time responsiveness—Mansha contributes to a more inclusive, intelligent future in neurotechnology and AI.
Keywords: Brain-Computer Interface (BCI), EEG, neural signals, signal processing, machine learning, assistive technology

I. INTRODUCTION

In recent years, the fusion of neuroscience and artificial intelligence has driven major advancements in Brain-Computer Interface (BCI) technology. BCIs establish direct communication between the brain and external devices, eliminating the need for traditional input methods like touchscreens and voice commands. This technology is especially transformative for individuals with physical disabilities, enabling more inclusive and efficient human-computer interaction.

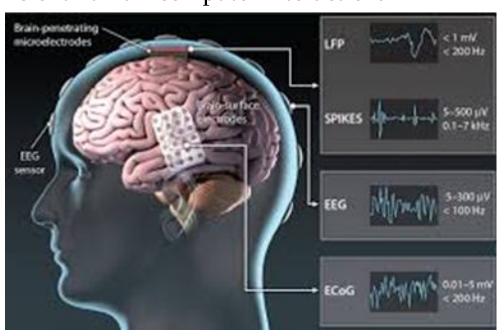


Fig. 1 Brain Computer Interface (bci) [9]

Project Mansha focuses on developing a BCI system that uses electroencephalography (EEG), signal processing, and machine learning to decode brain activity into actionable commands. By capturing and analyzing neural signals, the system enables hands-free control of computers, robotics, and smart home devices. The BCI pipeline involves EEG signal acquisition, feature extraction, classification, and device control, requiring interdisciplinary collaboration across neuroscience, AI, and engineering.
The applications of Project Mansha span assistive technologies, smart home automation, robotics, and healthcare. It enables individuals with motor impairments to control wheelchairs, prosthetics, and communication tools, while also allowing seamless interaction with home systems and robotic devices. In healthcare, it supports neurorehabilitation and brain monitoring. By enhancing the accuracy and adaptability of BCI systems, Mansha contributes to the evolution of neuroadaptive interfaces, promoting greater accessibility and quality of life.

II. LITERATURE SURVEY

The field of Brain-Computer Interface (BCI) has witnessed remarkable growth in recent years, driven by the convergence of neuroscience, signal processing, and artificial intelligence (Yuan & He, 2014) [1]. Early BCI systems primarily focused on basic motor imagery tasks and binary communication, offering limited real-world applicability. Seminal research by Wolpaw et al. and Birbaumer et al. laid the groundwork by demonstrating the feasibility of non-invasive BCI using electroencephalography (EEG) for communication and control (Mak & Wolpaw, 2009) [5]. These studies introduced protocols like Slow Cortical Potentials (SCP) and Event-Related Potentials (ERP), which have been widely adopted in clinical BCI applications for individuals with locked-in syndrome or severe paralysis (Daly & Wolpaw, 2008) [3].

With the advancement of machine learning and signal processing, newer studies have explored more complex classification techniques to improve signal interpretation accuracy (Bashashati et al., 2007) [4]. Feature extraction methods such as Common Spatial Patterns (CSP) (Blankertz et al., 2008) [7], Wavelet Transforms, and Power Spectral Density (PSD) analysis have enhanced the ability to distinguish neural patterns. Classification algorithms including Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), and more recently, Deep Learning models like Convolutional Neural Networks (CNNs) have shown significant promise in real-time EEG signal classification (Roy et al., 2019) [2]. These advancements have not only improved response speed and accuracy but also reduced the training time required for user adaptation.

The application domains of BCI have broadened significantly, extending from assistive technologies to smart environments, virtual reality, and industrial automation (Yuan & He, 2014) [1]. Studies such as those by Nicolas-Alonso and Gomez-Gil (2012) provide comprehensive reviews of BCI applications in smart home automation and mobility assistance. In such systems, users can control wheelchairs, prosthetic limbs, and even home appliances through mental commands. Similarly, integration of BCI with IoT platforms and robotic systems has enabled real-time, hands-free control in complex environments, offering immense potential for individuals with motor impairments.

Healthcare remains one of the most impactful areas for BCI research. Several studies highlight the role of BCI in neurorehabilitation, where repetitive mental tasks paired with feedback-based training can promote neural plasticity in stroke or spinal cord injury patients (Daly & Wolpaw, 2008) [3]. Projects like BrainGate and NeuroPace have demonstrated the viability of BCIs in restoring motor function or detecting epileptic seizures. These clinical successes, however, are often limited by issues such as signal variability, user fatigue, and non-standardized hardware protocols, highlighting a need for more adaptive and user-friendly systems (Mak & Wolpaw, 2009) [5].

Despite the progress, current BCI systems still face significant challenges in achieving widespread usability. Variability in EEG signals across users and sessions, noise sensitivity, lack of portability, and the need for long calibration periods are key barriers (He et al., 2020) [6]. Additionally, most systems focus on binary classification or a limited set of commands, constraining their applicability in dynamic real-world scenarios. Recent research calls for more robust, scalable, and personalized systems that can operate in real time with minimal training (Roy et al., 2019) [2].

Project "Mansha" addresses these limitations by focusing on improved signal fidelity, adaptive machine learning, and intuitive user interfaces. By incorporating advanced EEG processing and classification techniques, Mansha aims to offer a more practical and responsive BCI solution for diverse applications. Unlike traditional systems, it prioritizes user adaptability and cross-domain integration, making it suitable not only for assistive technology but also for smart automation, gaming, education, and high-stakes environments like aerospace and defense. This positions Mansha as a novel contribution to the evolving landscape of neurotechnology.

III. PROBLEM STATEMENT

Traditional human-computer interaction methods—such as touchscreens, keyboards, and voice commands—are not universally accessible, especially for individuals with motor impairments caused by conditions like spinal cord injuries, stroke, or neurodegenerative diseases.

These conventional input mechanisms rely on physical movement or speech, creating barriers for users who lack full motor control. This gap in accessibility not only limits the independence of affected individuals but also restricts their ability to engage with modern digital and smart environments effectively.

Brain-Computer Interface (BCI) technology offers a promising solution by enabling direct communication between the brain and external devices through the interpretation of neural signals. However, existing BCI systems often face several limitations, including low signal accuracy, high noise interference, delayed processing times, and difficulty adapting to individual user differences. Additionally, many systems require extensive training or calibration, making them impractical for everyday use. These challenges reduce the reliability, scalability, and real-world applicability of BCI technology across diverse users and settings.

Project "Mansha" aims to address these problems by developing an advanced, EEG-based BCI system that prioritizes accuracy, real-time responsiveness, and user adaptability. The project seeks to create a seamless and intuitive interface that enables hands-free control of devices in assistive technology, smart home automation, and healthcare. By integrating robust signal processing and adaptive machine learning models, Mansha targets the core limitations of current BCI systems and strives to enhance accessibility, usability, and the overall quality of human-computer interaction for individuals with motor impairments and beyond.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

IV. METHODOLOGY

The methodology of Mansha follows an Agile IoT development approach, combining iterative software development with structured hardware prototyping and real-time system integration to enable seamless brain-to-device interaction.

A. Methodology of Mansha: Neuroadaptive

1) NeuroAgile Development Lifecycle

Project Mansha adopts a NeuroAgile development lifecycle that merges Agile software practices with structured IoT system design. The approach emphasizes iterative prototyping, real-time validation, and continuous user feedback from requirement gathering to deployment. This ensures a smooth and adaptive development process that aligns hardware and software efforts for building an efficient brain-controlled interface.

2) Iterative and Adaptive Development

Mansha's methodology incorporates agile, sprint-based development cycles, allowing for rapid refinement of machine learning models and user interfaces. This iterative structure supports dynamic updates based on testing and user feedback, ensuring the system evolves continuously for better performance and usability.

3) Seamless Hardware-Software Integration

The project ensures parallel development and tight integration of hardware components (EEG acquisition systems) with software modules (signal processing, ML algorithms, and user interfaces). This cohesive process minimizes compatibility issues and accelerates testing, leading to a more stable and responsive system.

4) User-Centric Design and Validation

Mansha prioritizes user-centric design by involving end-users—particularly individuals with motor impairments—throughout the development cycle. Their input guides the design of intuitive controls and ensures the final product addresses real-world accessibility needs effectively.

5) Scalability and Flexibility

The system is designed with a modular architecture and compatibility with multiple communication protocols such as MQTT, BLE, and HTTP. This allows for easy customization, scalability across diverse platforms, and adaptability to emerging use cases or user-specific requirements.

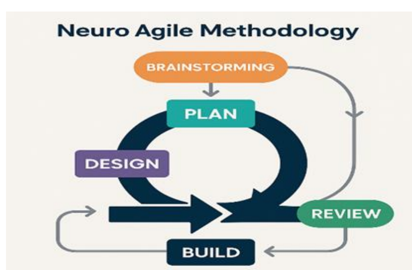


FIG. 2 NEURO AGILE METHODOLOGY

B. Stakeholder Engagement and System Design

Project Mansha is built on a robust and flexible system design that bridges neuroscience, artificial intelligence, and IoT technologies to create a seamless Brain-Computer Interface (BCI). The methodology behind Mansha prioritizes real-time performance, user accessibility, and adaptability, ensuring that both end-users and developers can benefit from its modular and scalable architecture. By combining neural signal processing with smart device control, and aligning development with stakeholder needs, Mansha delivers a powerful solution aimed at transforming assistive technology, healthcare, and human-computer interaction.

1) *User-Centric Problem Identification* - Engage with individuals facing motor impairments to understand their challenges with traditional device control. Define real-world use cases like smart home automation or assistive mobility based on user needs.
2) *BCI System Design & Component Selection* - Choose suitable EEG-based hardware (e.g., BioAmp EXG Pill, Arduino, electrodes) and ensure components are affordable, user-friendly, and safe for long-term, non-clinical use.
3) *Prototype Development & Signal Training* - Build an initial prototype in a controlled environment. Collect EEG data, process signals using Python, and train machine learning models to classify brainwave patterns into control commands.
4) *IoT Integration & Real-World Testing* - Connect the BCI system with smart devices using Bluetooth or Wi-Fi. Deploy and test in real environments to evaluate performance, latency, and ease of use. Ensure the system is responsive and safe.
5) *Feedback-Based Optimization* - Gather insights from user trials to refine signal accuracy, usability, and comfort. Update both hardware placement and software models for better performance in real-life scenarios.

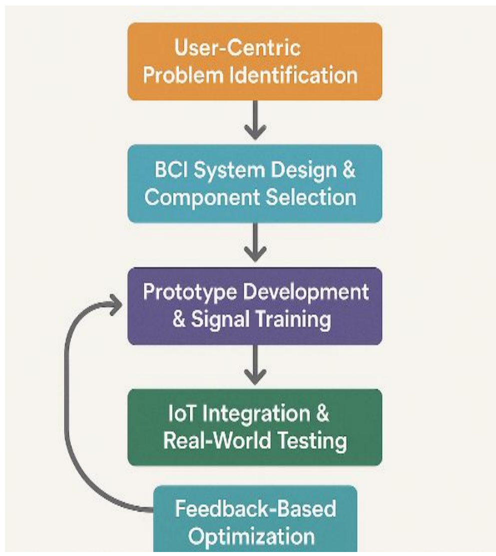


FIG. 3 STACKHOLDER ENGAGEMENT AND SYSTEM DESIGN

V. SYSTEM HARDWARE SPECIFICATION

A. *EEG-Based BCI Hardware Module*

The EEG-Based BCI Hardware Module is a specialized setup designed to acquire, process, and transmit neural signals for use in Brain-Computer Interface applications. It comprises components such as EEG headbands, biopotential amplifiers, microcontrollers, conductive electrodes, and signal conditioning accessories.

1) *Channel Brain BioAmp Band* - This is a wearable headband embedded with multiple electrodes specifically designed to capture EEG (electroencephalography) signals from the scalp. It plays a crucial role in detecting brainwave activity, which is then used to interpret user intent in the BCI system.
2) *Arduino Uno* - A microcontroller board based on the ATmega328P, the Arduino Uno is used to interface with the BioAmp EXG Pill and other hardware components in the system. It plays a key role in processing analog neural signals by converting them into digital data, which is then transmitted to a computer or processing unit for classification and device control.
3) *BioAmp Cable (100 cm)* - This long cable connects the BioAmp EXG Pill to the electrode bands or individual electrodes. The generous length allows comfortable positioning of equipment, avoiding strain or movement restriction during usage.
4) *Jumper Cables* - These flexible wires are used to make quick, reliable connections between different electronic components such as the EXG Pill, the Maker Uno, and other modules. They are essential for setting up and modifying the circuit during development and testing.
5) *Electrode Gel* - This conductive gel enhances the contact between the electrodes and the skin by lowering resistance. It ensures that electrical signals from the brain or body are transmitted with minimal loss or distortion, improving signal clarity.
6) *Repositionable Gel Electrodes* - These are reusable electrodes that can be adjusted and reapplied multiple times. They are ideal for prototype testing, repositioning during calibration, or multi-session studies where consistent placement is needed.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

FIG 4. EEG APPLICATION

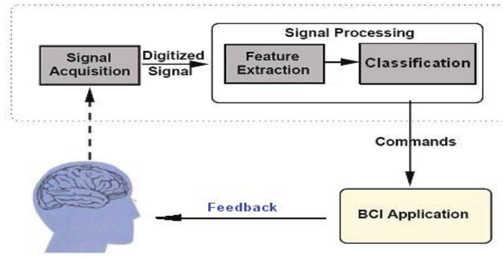


FIG 5. BCI BLOCK DIAGRAM [12]

VI. SYSTEM SOFTWARE SPECIFICATION

The BCI Software Framework in Project "MANSHA" is a cohesive set of tools and technologies used to process and interpret neural signals for real-time device control. It starts with firmware on the Arduino Uno, capturing EEG data from the BioAmp EXG Pill and sending it to a computer. Python is used for signal filtering, feature extraction, and classification using libraries like NumPy, SciPy, and scikit-learn. Custom GUIs built with Tkinter or PyQt enable user interaction and signal visualization. Communication interfaces such as PySerial and Bluetooth allow the system to control external devices, completing the brain-to-machine communication loop.

1) *Arduino IDE* - The Arduino IDE is used to program the Arduino Uno microcontroller. It facilitates the reading of analog EEG signals from the BioAmp EXG Pill and transmits the data to a computer using serial communication.
2) *Python* - Python is the central programming language for developing the BCI software pipeline. It enables real-time data acquisition, signal filtering, and noise removal.
3) *Machine Learning Frameworks* - To decode brain activity into actionable commands, scikit-learn is used to implement machine learning models such as SVM or decision trees. These models classify EEG patterns based on training data.
4) *Graphical User Interface (GUI)* - A GUI developed in Python using libraries like Tkinter, PyQt, or Streamlit allows for real-time visualization of brainwave signals. The interface provides user feedback and control options, helping users monitor their mental commands during training and testing phases.
5) *Signal Visualization and Plotting* - To better understand and debug brainwave patterns, visualization libraries such as Matplotlib and Seaborn are used. These tools enable the graphical representation of raw and filtered EEG signals, offering insights into neural responses during system interaction.
6) *Communication Interfaces* - For transmitting control signals to external systems, PySerial is used for wired communication through USB. In scenarios where wireless control is needed, technologies such as Bluetooth (using PyBluez) or Wi-Fi (using MQTT, Socket, or HTTP protocols) are employed, enabling flexible and seamless integration with smart devices and assistive tools.

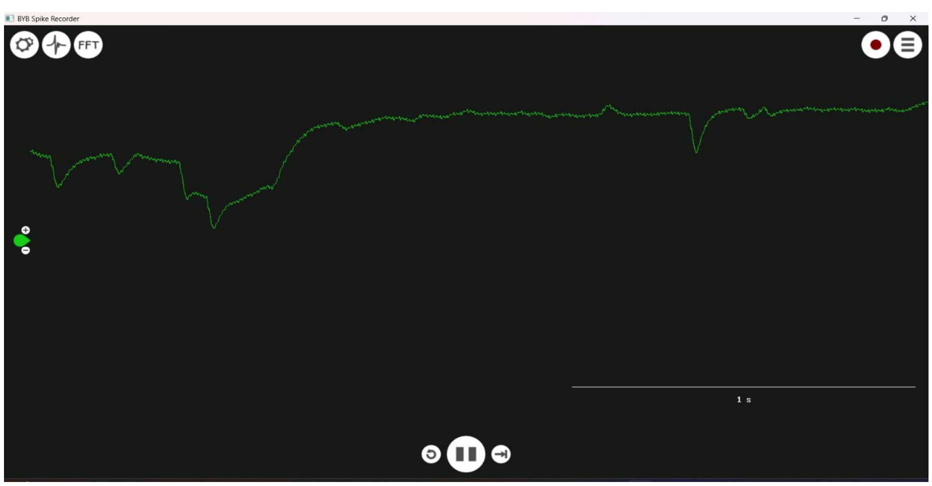


FIG 6. EEG SIGNAL

VII. LIMITATIONS

Although EEG-based brain-computer interface (BCI) systems provide an innovative means of device control through brain signals, they face several limitations that can affect their accuracy, functionality, and user experience. These challenges must be acknowledged to develop more reliable and effective systems. The main limitations include:

A. *Signal Noise and Interference*

1) EEG signals are highly sensitive to various forms of interference.
2) Muscle movements, electrical noise from surrounding devices, and poor electrode contact can distort the signals.
3) This interference can lead to inaccurate readings and misclassification of brain signals, reducing system reliability.

B. *Limited Command Set*

1) EEG has low spatial resolution, making it difficult to distinguish between subtle brainwave patterns.
2) As a result, the system can only recognize a small set of mental commands.
3) This limits the complexity and flexibility of device control.

C. *User Training and Adaptability*

1) Users typically need significant training to produce consistent and distinguishable brainwave patterns.
2) Performance varies from person to person, with some users requiring extended practice.
3) Without proper training, achieving reliable control may be difficult for many individuals.

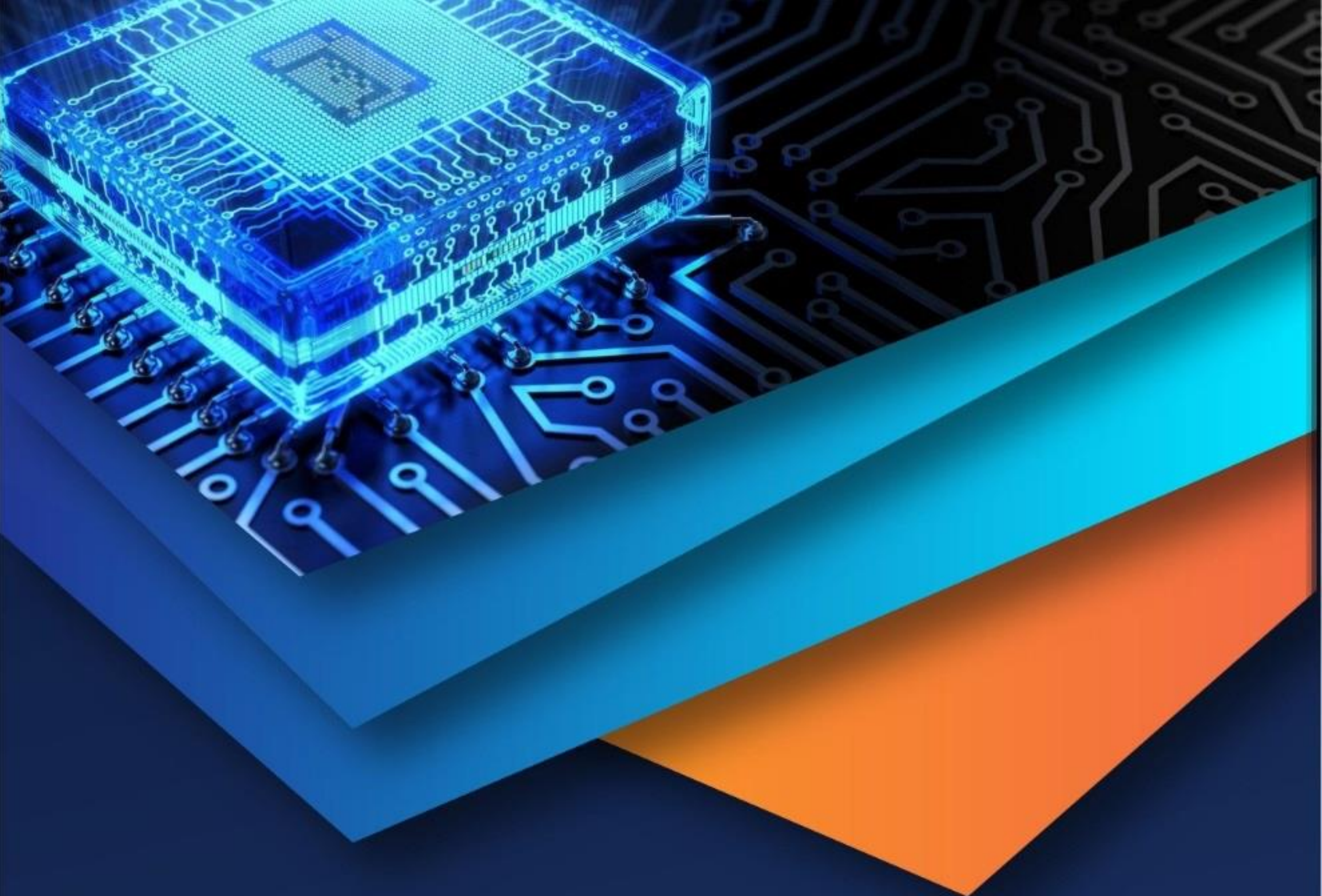
VIII. ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our colleagues for their invaluable support, guidance, and encouragement throughout this research. Their insights, feedback, and collaborative spirit were crucial in shaping the direction of this project. We also extend our deepest thanks to our professors for their unwavering support and mentorship. Their expert advice and encouragement were essential in refining our ideas and driving the success of this research. Special thanks are due to the institutions and technological advancements that provided the tools and resources necessary for the development and testing of our prototype. Additionally, we would like to acknowledge the contributions of all the authors whose research served as a foundation for this work. Their scholarly contributions were a constant source of inspiration and provided critical perspectives that guided us through the challenges of this study. Finally, we express our appreciation to all those who have directly or indirectly contributed to the completion of this project. Without their help, this work would not have been possible.

REFERENCES

[1] Yuan, H., & He, B. (2014). Brain–computer interfaces using sensorimotor rhythms: Current state and future perspectives. IEEE Transactions on Biomedical Engineering, 61(5), 1425–1435. https://doi.org/10.1109/TBME.2014.2313861
[2] Roy, Y., Banville, H., Albuquerque, I., Gramfort, A., Falk, T. H., & Faubert, J. (2019). Deep learning-based electroencephalography analysis: A systematic review. Journal of Neural Engineering, 16(5), 051001. https://doi.org/10.1088/1741-2552/ab260c
[3] Daly, J. J., & Wolpaw, J. R. (2008). Brain–computer interfaces in neurological rehabilitation. The Lancet Neurology, 7(11), 1032–1043. https://doi.org/10.1016/S1474-4422(08)70223-0
[4] Bashashati, A., Fatourechi, M., Ward, R. K., & Birch, G. E. (2007). A survey of signal processing algorithms in brain–computer interfaces based on electrical brain signals. Journal of Neural Engineering, 4(2), R32–R57. https://doi.org/10.1088/1741-2560/4/2/R03
[5] Mak, J. N., & Wolpaw, J. R. (2009). Clinical applications of brain-computer interfaces: Current state and future prospects. IEEE Reviews in Biomedical Engineering, 2, 187–199. https://doi.org/10.1109/RBME.2009.2035356
[6] He, H., Wu, D., et al. (2020). Transfer learning for brain–computer interfaces: A Euclidean space data alignment approach. IEEE Transactions on Biomedical Engineering, 67(2), 399–410. https://doi.org/10.1109/TBME.2019.2913926
[7] Blankertz, B., Tomioka, R., et al. (2008). Optimizing spatial filters for robust EEG single-trial analysis. IEEE Signal Processing Magazine, 25(1), 41–56. https://doi.org/10.1109/MSP.2008.4408441
[8] Brain-Machine Interface Technologies - Bannari Amman Institute of Technology
[9] Arduino. (n.d.). Arduino Uno Rev3 documentation. Arduino.cc. https://docs.arduino.cc/hardware/uno-rev3
[10] Upside Down Labs. (n.d.). BioAmp EXG Pill: Open-source biopotential sensor. GitHub. https://github.com/upsidedownlabs/BioAmp-EXG-Pill
[11] Python Software Foundation. (n.d.). Python 3 documentation. https://docs.python.org/3/
[12] The block diagram of a BCI system-Research gate
[13] Harris, C. R., et al. (2020). Array programming with NumPy. Nature, 585, 357–362. https://numpy.org/doc/stable/
[14] Virtanen, P., et al. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. Nature Methods, 17, 261–272. https://docs.scipy.org/doc/scipy/

[15] Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830. https://scikit-learn.org/stable/
[16] Abadi, M., et al. (2016). TensorFlow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 265–283. https://www.tensorflow.org/
[17] Paszke, A., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems, 8026–8037. https://pytorch.org/docs/stable/
[18] Python Software Foundation. (n.d.). Tkinter GUI library. https://docs.python.org/3/library/tkinter.html
[19] PySerial Developers. (n.d.). PySerial: Python serial port extension. https://pyserial.readthedocs.io/en/latest/
[20] PyBluez. (n.d.). PyBluez: Bluetooth for Python. GitHub. https://github.com/pybluez/pybluez
[21] Streamlit Inc. (n.d.). Streamlit documentation. https://docs.streamlit.io/
[22] MQTT. (n.d.). MQTT protocol documentation. MQTT.org. https://mqtt.org/documentation/
[23] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90–95. https://matplotlib.org/stable/contents.html
[24] Waskom, M., et al. (2021). Seaborn: Statistical data visualization. Journal of Open Source Software, 6(60), 3021. https://seaborn.pydata.org/
[25] InfluxData. (n.d.). InfluxDB time series database. https://docs.influxdata.com/influxdb/
[26] Espressif Systems. (n.d.). ESP32 documentation.
[27] https://docs.espressif.com/projects/esp-idf/en/latest/esp32/
[28] Upside Down Labs. (n.d.). BioAmp EXG Pill: Open-source biopotential sensor. GitHub. https://github.com/upsidedownlabs/BioAmp-EXG-Pill



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)