GLOVE-NET: A MULTI SENSOR INTELLIGENT SOLUTION FOR GRASP CLASSIFICATION

Mini-Project Report submitted to the SASTRA Deemed to be University
in partial fulfillment of the requirements
for the award of the degree of

B. Tech. Electronics & Instrumentation Engineering

Submitted by

Balaji M
(Reg. No.: 126006058)
Premkumar T
(Reg. No.: 126006033)

May 2025

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

SCHOOL OF ELECTRICAL & ELECTRONICS ENGINEERING

THANJAVUR, TAMIL NADU, INDIA – 613 401

SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 of the UGC Act, 1956)

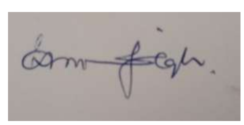
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

SCHOOL OF ELECTRICAL & ELECTRONICS ENGINEERING

THANJAVUR – 613 401

Bonafide Certificate

This is to certify that the report titled "**Glove-net: A Multi Sensor Intelligent solution for Grasp classification**" submitted in partial fulfillment of the requirements for the award of the degree of B. Tech. Electronics & Instrumentation Engineering to the SASTRA Deemed to be University, is a bona-fide record of the work done by **Balaji M (Reg. No.: 126006058) & Premkumar T (Reg. No.: 126006033)** during the Sixth semester of the academic year 2024-25, in the **School of Electrical & Electronics Engineering**, under my supervision. This report has not formed the basis for the award of any degree, diploma, associateship, fellowship or other similar title to any candidate of any University.



Signature of Project Supervisor :

Name with Affiliation : **Dr. Ghousiya Begum K, Asst. Professor-III**

Date : **08-05-2025**

Project *Viva voce* held on ________________

Examiner 1 **Examiner 2**

SCHOOL OF ELECTRICAL & ELECTRONICS ENGINEERING

THANJAVUR – 613 401

Declaration

We declare that the report titled "**Glove-net: A Multi Sensor Intelligent solution for Grasp classification**" submitted by us is an original work done by me/us under the guidance of **Dr. Ghousiya Begum K, Asst. Professor-III, School of Electrical and Electronics Engineering, SASTRA Deemed to be University** during the Sixth semester of the academic year **2024-25**, in the **School of Electrical and Electronics Engineering**. The work is original and wherever I/we have used materials from other sources, I/we have given due credit and cited them in the text of the report. This report has not formed the basis for the award of any degree, diploma, associate-ship, fellowship or other similar title to any candidate of any University.

Signature of the candidate(s) :

Name of the candidate(s) : **Balaji M**
Premkumar T

Date : **08-05-2025**

Acknowledgements

We express our gratitude to honorable **Prof. R. Sethuraman**, Chancellor and **Dr. S. Vaidhyasubramaniam**, Vice Chancellor, SASTRA Deemed to be University for the opportunity of pursuing our engineering in this esteemed in situation and carrying out the project work.

We thank **Dr. R. Chandramouli**, Registrar, SASTRA Deemed to be University for granting permission and extending the facilities in carrying out this project.

We express our sincere thanks and gratitude to our respectable **Dr. K. Thenmozhi**, Dean, SEEE and **Dr. Sridhar K**, Dean, Research **,Dr. A. Krishnamoorthy**, Associate Dean-Academics SEEE, **Dr. V. Muthubalan**, Associate Dean-Research SEEE, **Dr. K. Vijayarekha**, Associate Dean-Student Welfare, **Dr. Manigandan. N.S,** Associate Dean-Infrastructure, SEEE SASTRA Deemed to be University, for their support in accomplishment of this work.

We would like to express our sincere gratitude to our guide, **Dr. Ghousiya Begum K, SASTRA Deemed to be University**, whose invaluable guidance and support enabled us to successfully complete this project. Her emphasis on experiential learning allowed us to grow through our mistakes, deepen our understanding of scientific concepts in power electronics and electric vehicles, and enhance our ability to analyze and interpret results.

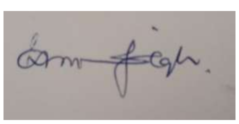
We also thank the project review panel members**, Dr. Sriranjani. R**, Sr. Asst.Professor and **Dr. Venkatesh. T**, Asst.Professor III, School of Electrical and Electronic Engineering, for their valuable comments and insights, which made this project better.

We would also like to thank our friends who willingly volunteered as subjects and cooperated with us in collecting real-time data for the project. Their support and participation were crucial to the successful completion of our work.

And finally, we would like to acknowledge the appreciation and support that our parents provided to ensure we faced minimal obstacles throughout the project.

Abstract

Hand paralysis is a prevalent complication among stroke patients in India, significantly impacting their daily lives.This project proposes the development of Smart Rehabilitation Gloves tailored specifically for Indian patients, integrating cost-effective materials and widely available electronic components. The system comprises a sensory glove utilizing soft, flexible materials to ensure comfort and safety. The sensory glove incorporates flex sensors and MPU6050 inertial measurement units (IMUs) to measure gripping force, finger joint angles, and hand orientation. A monitoring system processes the sensor data to provide real-time feedback and assist patients in performing rehabilitation exercises accurately. Machine learning algorithms accurately recognize 7 predefined grasp motion such as Reaching, Holding, moving up, moving left, Moving Down, Releasing and Relaxing and task-oriented training with an accuracy exceeding 90%. Integrated IoT capabilities enable remote monitoring, personalized therapy adjustments. Designed to address the affordability and healthcare challenges unique to India, this system promises to enhance hand functionality, reduce therapy costs, and improve accessibility for home-based rehabilitation.



Signature of the Guide

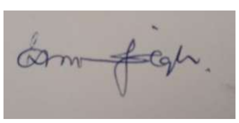
Student Reg. No: 126006058

Name: **Dr. Ghousiya Begum K, SEEE**

Name: **Balaji M**

Abstract

Hand paralysis is one of the most common and life-altering complications faced by stroke patients, often making everyday tasks incredibly challenging. To help address this, we present a wearable hand rehabilitation system that supports both mirror therapy and task-oriented therapy. The system includes specially designed gloves a sensory glove made from soft, flexible materials to ensure greater comfort and safety compared to traditional rigid rehab devices. The sensory glove, worn on the unaffected hand, is equipped with force and flex sensors that measure the gripping force and finger joint movements for precise motion detection. Machine learning algorithms accurately recognize 7 predefined grasp motion such as Reaching, Holding, Moving up, Moving left, Moving Down, Releasing and Relaxing and task-oriented training with an accuracy exceeding 90%. To make the rehabilitation process even smarter, machine learning techniques are used to recognize gestures captured by the sensory glove, enabling smoother and more responsive support for therapy tasks.



Signature of the Guide   Student Reg. No: 126006033

Name: **Dr. Ghousiya Begum K, SEEE**   Name: **Premkumar T**

Table of Contents

List of Figures

List of Tables

ABBREVIATIONS

LSTM	Long Short-Term Memory
GPIO	General Purpose Input/Output
ADC	Analog-to-Digital Converter
DAC	Digital to Analog Converter
SDA	Serial Data
SCL	Serial Clock
I2C	Inter-Integrated Circuit
PCB	Printed Circuit Board
MPU	Motion Processing Unit
VCC	Voltage at common collector
GND	Ground
AD0	I2C Slave address Pin

NOTATIONS

kΩ - Kilo Ohm
Hz – Hertz
V – Voltage
α – Weight lays on Gyroscope measurement

CHAPTER 1
INTRODUCTION

Recovering from a stroke can be an overwhelming journey—especially when it leaves lasting effects like hand paralysis. For many stroke survivors in India, the road to recovery is made even harder by limited access to affordable rehabilitation. Therapy centers are often far away, costly, or overburdened, and many patients simply don't get the consistent care they need. As a result, simple daily tasks—like holding a spoon or buttoning a shirt—can remain out of reach for far too long. This project introduces a Smart Rehabilitation Glove designed to make recovery more accessible, especially for Indian patients. The glove is made from soft, flexible materials that are comfortable to wear and affordable to produce. Inside, it's equipped with flex sensors and MPU6050 motion sensors that work together to track how the hand moves—measuring grip strength, finger positions, and overall orientation. As patients wear the glove and perform guided exercises, the system provides real-time feedback, helping them correct their movements on the spot. This immediate response ensures that each motion is done properly, which is essential for regaining muscle control and rebuilding strength. What makes this glove even more powerful is its built-in machine learning model, which can recognize seven key hand movements—such as Reaching, Holding, Moving, and Releasing—with over 90% accuracy. This allows it to support task-based training, which is not only more effective but also feels more meaningful to patients. Beyond the glove itself, the system includes IoT capabilities so that therapy can be monitored remotely. Doctors and therapists can track progress, adjust routines, and support patients—no matter where they are. This feature is especially valuable in India, where traveling for regular sessions isn't always possible. By combining smart technology with a deep understanding of local needs, this glove offers more than just innovation—it offers hope. It brings therapy into the home, reduces costs, and gives stroke survivors a real chance at regaining independence and improving their quality of life.

CHAPTER 2
LITERATURE REVIEW AND OBJECTIVE

2.1 REVIEW OF THE LITERATURE:

[1] This paper explores the challenges and innovations in a dual-glove system designed for stroke rehabilitation. The setup includes a motorized glove that aids in hand movement and a sensory glove that detects gestures. By combining flex and force sensors with machine learning, the system supports therapies like mirror therapy and task-based training. It enables real-time, at-home rehabilitation with the ability to recognize 16 detailed hand gestures with an impressive accuracy of 93.32%. Built using soft, flexible materials and equipped with robust IoT integration, the system sets a new benchmark in comfort, affordability, and gesture recognition accuracy for rehabilitation devices.

[2] This study presents a simple yet effective biofeedback device designed to track finger movement in stroke patients using flex sensors and an Arduino Nano. By converting real-time resistance readings into angular motion and displaying them through a C#-based graphical interface, the system offers a cost-effective solution for rehabilitation. The ability to visualize progress encourages patients to stay engaged and exercise regularly. While it doesn't automate or control movement like more advanced systems, its focus on tracking therapy improvements makes it particularly valuable in low-resource settings, where affordability and simplicity are key.

[3] This study introduces a soft rehabilitation glove powered by soft actuators, capable of supporting both flexion and extension, as well as abduction and adduction movements. To optimize pressure-driven motion, the system uses hybrid PID control in combination with finite element analysis. The glove is equipped with force and strain sensors and supports personalized grasping tasks tailored to the user's needs. Unlike bulky, rigid exoskeletons, this design is lightweight, safe, and highly adaptable. Its precise motion control and advanced actuation system make it ideal not only for rehabilitation but also for assisting users in everyday tasks.

[4] This paper addresses the challenges of accurately determining tilt angles using MPU6050 sensors in multirotor drones, especially under the influence of motor-induced vibrations. To reduce cumulative errors and vibration-related noise, the study proposes

customized FIR filter combined with an optimized complementary filter, enhanced through Gray Wolf Optimization (GWO). It also introduces a novel nonlinear correction method to counter tilt distortion caused by yaw movements. By effectively fusing data from the accelerometer and gyroscope, the approach significantly improves real-time angle estimation, offering better stability and accuracy in drone flight control.

[5] This study presents a Smart Glove system designed for therapy treatment, incorporating flex sensors to monitor finger bending and assist in physical rehabilitation. The system captures finger motion data to support guided exercises. While low-cost and functional, it lacks real-time adaptive feedback and advanced automation for personalized therapy.

[6] This paper proposes a rehabilitative embedded hand glove aimed at assisting paralyzed patients through sensor-based motion tracking. The glove is built with affordable components and targets basic motor recovery. However, the design lacks precise control algorithms and broader motion support, limiting its use in advanced rehabilitation or complex hand tasks.

[7] This research introduces a soft pneumatic actuator for hand rehabilitation, capable of generating safe and compliant assistance for finger movements. The actuator's performance is experimentally validated. However, the system requires external air supply units, making it less portable and harder to integrate into compact, wearable rehabilitation solutions.

[8] The study develops a soft robotic glove intended for both rehabilitation and daily assistance, using fluid-powered actuators to mimic natural hand motions. The glove allows users to perform tasks at home independently. Despite its effectiveness, the glove's cost and dependency on external pneumatic systems hinder widespread, low-cost deployment.

2.2 OBJECTIVE OF THE PROJECT:

1. Develop Cost-Effective Smart Rehabilitation Gloves:
   Design soft, sensor-integrated rehabilitation gloves using cost-effective materials and components to make hand therapy more accessible for stroke patients in low-income communities.
2. Accurately Measure Hand Movements:
   Utilize flex sensors and MPU6050 IMUs to accurately monitor finger movements, joint angles, and overall hand orientation during rehabilitation exercises.
3. IoT Integration and Data Transfer:
   Integrate WIFI-enabled ESP32 modules to stream real-time sensor data to cloud platforms, enabling remote monitoring and timely feedback from therapists.
4. Gesture Recognition Using Machine Learning:
   Train Machine learning models on motion data to classify predefined hand gestures, allowing for automated feedback to tailor rehabilitation routines to individual needs.

CHAPTER 3
METHODOLOGY

The methodology provides a structured plan for the project, starting with a comprehensive literature review to establish a solid foundation. Based on the insights gained, appropriate sensors are selected and integrated into a smart glove. With the hardware set up, data is collected, cleaned, and pre-processed for analysis. Machine learning algorithms are then employed to identify various motion phases. Finally, the results are visualized through a web dashboard and connected via IoT, leading to the creation of a functional prototype.

3.1 OVERVIEW

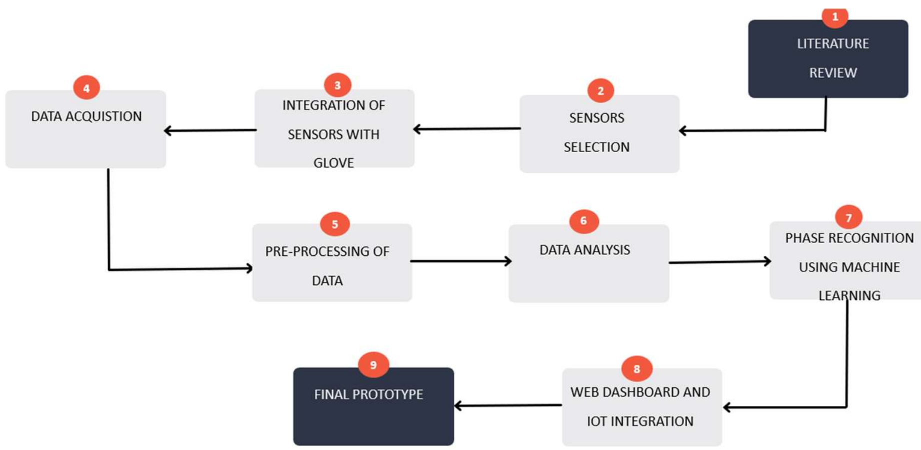


Figure 3.1. Flowchart illustrating the step-by-step methodology

3.2 BLOCK DIAGRAM AND DESCRIPTION

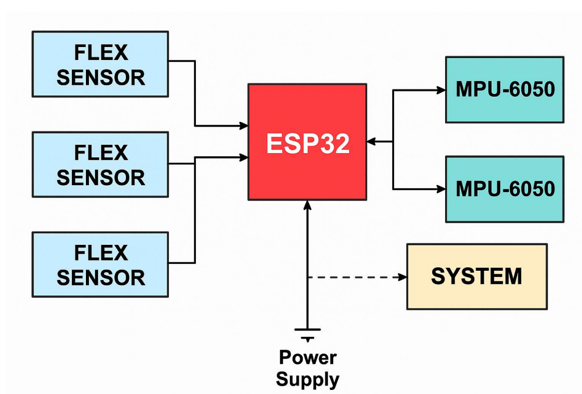


Figure 3.2. Block Diagram

The Figure 3.2 shows the block diagram of **three Flex Sensors** and **two MPU-6050 sensors** connected to an **ESP32** microcontroller. The **ESP32** is powered by an external **power supply** and establishes a **wireless connection** to a **system** (like a computer or server) for data communication. Arrows represent the data flow and connections between components

3.3 COMPONENTS USED

3.3.1 ESP32

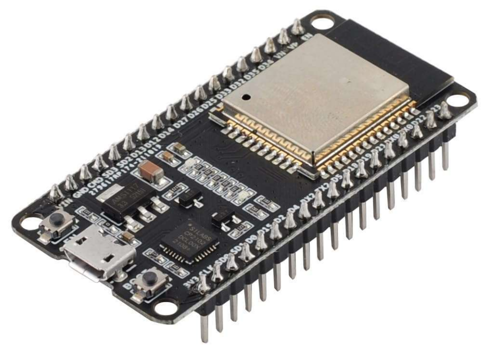


Figure 3.3. ESP32 Development Board

The Figure 3.3 show ESP32. The ESP32 is an affordable, energy-efficient microcontroller developed by Espressif Systems, offering both Wi-Fi and Bluetooth connectivity. It is powered by a dual-core Tensilica Xtensa processor and includes a variety of integrated peripherals such as ADCs, DACs, SPI, I2C, and UART. These features make it well-suited for IoT and embedded system applications. Compact yet powerful, the ESP32 supports real-time processing, wireless communication, and sensor integration, making it a popular choice for smart devices, wearables, and home automation projects.

- Working Voltage: 3.0V to 3.6V (typically 3.3V)
- GPIO Pins: 34 (multi-purpose – digital I/O, ADC, DAC, etc.)
- ADC Channels: Up to 18 (12-bit resolution)
- DAC Channels: 2 (8-bit resolution)
- Operating Current: ~80 mA average during active use

3.3.2 FLEX SENSOR

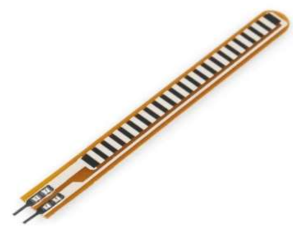


Figure 3.4. Flex Sensors

The Figure 3.4 show Flex Sensors .Flex sensors are thin, flexible resistive strips that alter their resistance as they bend. Typically, their resistance starts around 10kΩ when straight and increases to 30–50kΩ as they bend. Operating at voltages between 3.3V and 5V, they provide analog output signals via a voltage divider. Due to their durability, low power consumption, and ease of integration with microcontrollers like the ESP32, flex sensors are commonly used in wearable technology, robotics, and rehabilitation devices. They are particularly effective for tracking finger movements and joint angles. Flat Resistance: ~10 kΩ when unbent.

- Bent Resistance: Can go up to 30–50 kΩ
- Bend Angle Range: Typically, 0° to 180°.
- Output Type: Analog voltage
- Response Time: <100 milliseconds

3.3.3 MPU6050

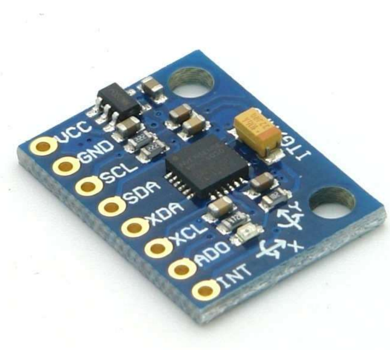


Figure 3.5. MPU6050 Sensor

The Figure 3.5 shows MPU6050. The MPU6050 is a compact 6-axis motion sensor that combines a 3-axis gyroscope (measuring rotational velocity) and a 3-axis accelerometer (detecting linear acceleration) using MEMS technology. It features an onboard Digital Motion Processor (DMP) for real-time motion tracking and sensor fusion, along with a 16-bit ADC for high-precision data conversion. Commonly used in robotics, drones, and wearables, the MPU6050 provides accurate orientation, acceleration, and rotational data via I²C communication, enabling applications like gesture recognition and stabilization systems. Its configurable I²C address also allows for seamless integration of multiple modules in a single system

- Operating Voltage: 2.3V to 3.4V
- Communication Interface: I²C (Inter-Integrated Circuit)
- Sensor Resolution: 16-bit ADC
- Power Consumption: Very low power usage

3.4 HARDWARE SETUP

3.4.1 INTERGRATION OF SENSORS WITH GLOVE

The integration of sensors into a soft, cotton-based glove specifically designed for precise hand motion tracking. The glove is made from breathable, flexible cotton fabric to ensure maximum comfort, allowing users to wear it for extended periods without discomfort. Flex sensors are embedded along each finger to measure the degree of bending by detecting changes in resistance as the fingers move.

Additionally, two MPU6050 module each combining a 3-axis accelerometer and a 3-axis gyroscope are strategically positioned to capture the orientation and motion of the entire hand. Color-coded wires connect all sensors to a central processing unit mounted on the glove, enabling seamless real-time data acquisition and feedback.

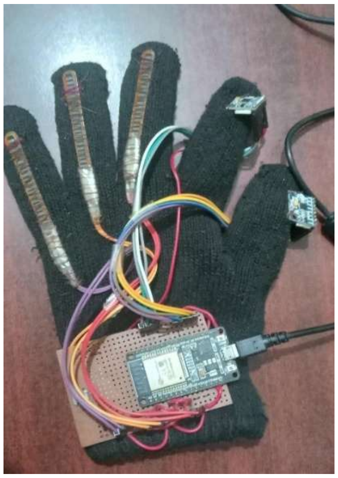


Figure 3.6 & 3.7. Integration of sensors with glove

The Figure 3.6 & 3.7 shows the setup that provides detailed insight into both individual finger movements and overall hand posture, making it highly effective for rehabilitation, gesture recognition, and other interactive applications. The glove strikes a balance between precision and comfort, ensuring accurate tracking without compromising user experience

3.4.2 CIRCUIT DIAGRAM AND CONNECTION

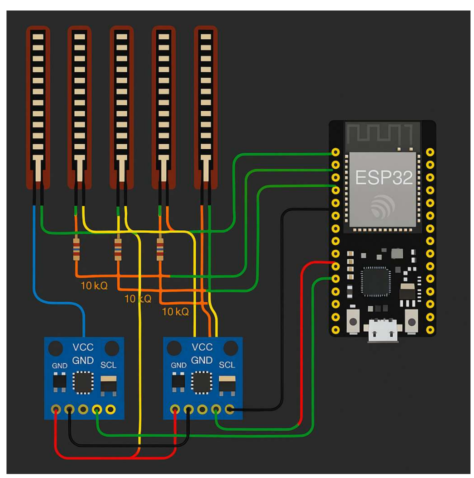


Figure 3.8. Circuit Diagram

The Figure 3.8 displays the circuit diagram connection. The ESP32 microcontroller connected to three flex sensors and two I2C modules. The flex sensors detect bending, and the ESP32 processes this data for applications like gesture recognition or smart wearables, using both analog and digital signals.

Flex Sensor Connections

The black glove is equipped with flex sensors on each finger that pick up on bending movements. These flexible, resistive sensors operate by altering their resistance whenever they are bent. When straight, a flex sensor has about 32.5kΩ resistance.When bent at 90°, resistance increases to approximately 80k. Each flex sensor is connected in a voltage divider configuration. One end of each flex sensor connects to the ESP32's 3.3V power supply .The other end connects to both a 10kΩ resistor and an analog input pin The 10kΩ resistor connects to ground, completing the voltage divider circuit This setup produces different voltage levels that the ESP32's analog-to-digital converter (ADC) can pick up. As the fingers move and bend, the resistance of the sensors shifts, which changes the voltage sent to the analog pin.

MPU6050 Sensor Integration

The glove appears to have two MPU6050 sensors connected to the ESP32 via I2C protocol: Both sensors share the same I2C bus connecting to:

- SDA (Serial Data) → GPIO 21 on ESP32
- SCL (Serial Clock) → GPIO 22 on ESP32

To prevent address conflicts the second sensor's address is changed to 0x69 by Connecting the AD0 pin of the second MPU6050 to 3.3V

The complete MPU6050 connections are:

- VCC → 3.3V
- GND → GND
- SDA → GPIO 21
- SCL → GPIO 22
- AD0 → 3.3V (only for second sensor)

3.4.3 ASSEMBLY AND ENCLOSURE

Assembly

PCB and Mounting Method

To keep everything organized, a small PCB is placed at the wrist section of the glove. It serves as a central hub, neatly bringing together the ESP32 microcontroller and all the sensor connections. The sensor wires are soldered onto the PCB, creating a clean and efficient setup.

Wiring and Soldering

- Each flex sensor is connected to the PCB through wires that are soldered to specific pins (32, 34, and 35) on the ESP32.
- A 10kΩ resistor is used in series with each flex sensor to form a voltage divider. This
- setup allows the ESP32 to accurately measure changes in resistance as varying analog voltage levels.
- Additional connections are made for the MPU6050 sensors via the I2C interface, linking the SDA and SCL lines to the PCB.

Attachment to the Glove

- The flex sensors are carefully stitched along the fingers of the glove using sewing thread.
- They are positioned to follow the natural bending points of the fingers, particularly at the knuckles.
- To keep the sensors secure without restricting finger movement, the thread is gently looped over each sensor above the joints.
- The PCB assembly itself is sewn onto the wrist area, ensuring that the whole system remains compact and tidy.

With this design, the glove is capable of capturing detailed finger movements through the flex sensors, while the MPU6050 sensors monitor the overall hand orientation and motion. Together, they form a complete and effective hand motion tracking system.

Enclosure

PCB Mounting:

- The PCB is mounted at the wrist area of the glove, secured using a combination of sewing thread and double-sided adhesive tape. This setup not only fixes the PCB firmly in place but also acts as a basic enclosure, providing both protection and user comfort.
- The enclosure safeguards the electronic components from exposure to sweat and mechanical impacts, while also preventing direct contact between the circuitry and the user's skin.

Wearable Enclosure Features

In wearable devices, the enclosures are made to be as lightweight and comfortable as possible, so they can be worn for long periods without causing discomfort. They are carefully designed to fit the shape of the body and move naturally with it, making sure the device feels almost unnoticeable during use.

Protection and Aesthetics

- Enclosures do more than just protect the electronics, they also enhance the overall look and feel of the device. A neat wrist-mounted enclosure, for example, can make the glove appear more polished and professional, moving it beyond the look of an early prototype.

3.5 REAL TIME DATA ACQUISITION

3.5.1 PROTOCOLS FOLLOWED

The Figure 3.9 to 3.15 depict a person wearing a sensor-equipped glove while interacting with an object, likely in a robotic or prosthetic hand experiment. The sequence captures key actions—reaching, holding, moving up, moving left, moving down, releasing, and relaxing—showing how the glove tracks and responds to hand movements.

3.9. Reaching

3.10. Holding

3.11. Moving Up

3.12. Moving Left

3.13. Moving Down

3.14. Releasing

3.15. Relaxing

3.5.2 DATA VISUAIZATION IN SERIAL MONITOR

Figure 3.16. Obtained sensor readings in Serial Monitor

The Figure 3.16 shows the serial monitor that shows real-time sensor data from the smart glove project. It displays ADC readings and corresponding angle measurements from three flex sensors, along with gyroscope data from two MPU6050 sensors connected at I2C addresses 0x68 and 0x69. This output confirms that the ESP32 is successfully reading, processing, and streaming data from all the sensors.

Fig 3.17. Objects Used

The Figure 3.17 shows a variety of everyday items neatly laid out on the floor—backpack, notebook, bottles, mugs, phones, utensils, toothbrush, coin, keychain, ball, and kettle—highlighting the diversity of daily essentials in a simple, organized setup.

3.5.3 RAW SENSOR DATA ACQUSITION USING PYTHON

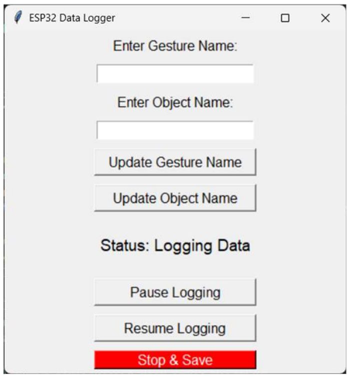


Fig 3.18. Data Logger GUI using Python

3350	0	3350	0	4095	90	-0.4	-1.17	0.53	-134.58	6.8	105.78	-4.98	-0.26	1.49	-9.62	-4.01
3350	0	3350	0	4095	90	5.71	0.85	-3.81	-133.37	6.98	104.97	-5.58	-1.4	0.05	-9.69	-4.03
3350	0	3350	0	4095	90	-1.61	0.37	1.95	-133.71	7.05	105.39	-5.36	-3.66	-3.32	-9.75	-4.07
3350	0	3350	0	4095	90	-1.94	1.66	2.83	-134.12	7.41	105.99	-3.63	-2.29	-2.24	-9.8	-4.1
3350	0	3350	0	4095	90	-3.27	1.71	1.25	-134.82	7.77	106.26	-4.71	-0.73	-0.9	-9.85	-4.11
3350	0	3350	0	4095	90	-0.89	2.61	3.07	-135.01	8.33	106.91	-3.68	-0.85	-1.66	-9.9	-4.12
3350	0	3350	0	4095	90	-3.56	1.27	1.38	-135.77	8.6	107.21	-3.85	-0.51	-0.43	-9.94	-4.12
3350	0	3350	0	4095	90	-2.38	2.17	1.69	-136.27	9.06	107.57	-5.29	-0.79	-0.85	-10.01	-4.13
3350	0	3350	0	4095	90	-2.3	2.65	1.69	-136.76	9.62	107.93	-3.76	-0.33	-1.74	-10.05	-4.14
3350	0	3350	0	4095	90	-2.9	-0.1	2.28	-137.38	9.6	108.41	-4.19	-1.47	-1.27	-10.1	-4.16
3350	0	3350	0	4095	90	-2.79	-0.27	1.92	-137.98	9.55	108.82	-4.4	-2.42	0.15	-10.16	-4.18
3350	0	3350	0	4095	90	-0.98	-2.47	-3.98	-138.19	9.02	107.98	-9.85	-4.77	-1.37	-10.27	-4.24
3350	0	3350	0	4095	90	-1.72	-1.32	2.96	-138.55	8.74	108.61	-4.56	-6.03	0.94	-10.33	-4.31
3350	0	3350	0	4095	90	-1.96	-12.87	6.89	-138.97	6	110.07	-3.69	-15.71	-1.52	-10.37	-4.5
3350	0	3350	0	4095	90	-0.98	-14.57	4.19	-139.18	2.89	110.97	-4.66	-17.09	-3.51	-10.43	-4.71
3350	0	3350	0	4095	90	2.21	1.36	0.47	-138.71	3.18	111.07	-6.2	-3.21	-4.05	-10.5	-4.75
3350	0	3350	0	4095	90	-3.93	1.38	1.45	-139.54	3.48	111.37	-5.37	-1.22	0.59	-10.57	-4.76
3350	0	3350	0	4095	90	0.14	0.02	3.35	-139.52	3.48	112.09	-3.88	-0.46	-2.35	-10.61	-4.77
3350	0	3350	0	4095	90	-2.34	-2.66	3.08	-140.01	2.92	112.74	-2.86	-5.49	-2.56	-10.65	-4.83
3350	0	3350	0	4095	90	-1.02	-2.32	2.19	-140.23	2.42	113.21	-5.32	-4.76	-2.4	-10.71	-4.89
3350	0	3350	0	4095	90	0.46	-6.41	1.92	-140.13	1.06	113.62	-2.73	-7.52	-1.43	-10.75	-4.98
3350	0	3350	0	4095	90	6.08	-2.53	-0.7	-138.84	0.52	113.47	7.99	-5.21	-2.98	-10.65	-5.04
3350	0	3350	0	4095	90	-1.23	-0.93	0.95	-139.1	0.32	113.67	-7.51	-3.6	-2.65	-10.74	-5.08
3350	0	3350	0	4095	90	1.46	-0.97	2.21	-138.79	0.11	114.14	-6.29	-2.71	-3.49	-10.82	-5.12
3350	0	3350	0	4095	90	-7.7	-0.33	3.73	-140.43	0.04	114.94	-2.19	-6.7	-4.44	-10.84	-5.2
3350	0	3350	0	4095	90	-2.66	1.84	0.77	-141	0.43	115.1	-5.27	-1.37	-1.38	-10.9	-5.21
3350	0	3350	0	4095	90	-1.86	-1.31	2.73	-141.39	0.16	115.68	-2.94	-5.4	-1.66	-10.94	-5.28

Fig 3.19. Raw Sensor Data

The Figure 3.18 and 3.19 shows Real-time data from the serial monitor is captured using Python's **pySerial** library, which reads the incoming data stream from the ESP32. The collected sensor data is then parse and saved into CSV files with the help of Python's built-in **csv** module. This setup allows for easy storage and later analysis of the recorded sensor readings.

3.6 PRE-PROCESSING OF DATA

3.6.1 CALIBRATION OF SENSORS

Flex Sensor Calibration

To ensure accurate angle measurements, the flex sensors need to be properly calibrated.

- Maximum value calibration: The user is asked to fully straighten their fingers to record the maximum resistance (straight position) for each sensor.
- Minimum value calibration: The user then bends their fingers to capture the minimum resistance (fully bent position).
- Storing calibration values: These maximum and minimum values are stored in the program, creating a reference range for each sensor.
- Mapping values: The analog readings are then converted into angle measurements — typically mapping straight fingers to around 0° and fully bent fingers to around 90° — using Arduino's map() function or a similar method.

MPU6050 Calibration

The MPU6050 sensors need to be calibrated to correct for zero errors and drift.

- Positioning: The sensor should be placed flat on a level surface and kept completely still during calibration.

- Calibration process: A calibration routine is run, which gathers hundreds of readings from the sensor and calculates the average offset values for each axis.
- Finding offsets: The routine determines the best offset values for the X, Y, and Z axes of both the accelerometer and the gyroscope.
- Applying offsets: These calculated offsets are then applied to the raw sensor data in the main program to ensure accurate and stable orientation readings.

3.6.2 FILTERING(DENOISING)

Filtering Flex Sensor Data Using a Low-Pass Filter

Flex sensors are prone to electrical noise, which can cause small fluctuations in the readings. To smooth out these inconsistencies, a low-pass filter is applied:

Software filtering: A simple digital low-pass filter can be implemented using the formula filteredValue = previousFilteredValue + α × (currentRawValue - previousFilteredValue)

where α controls how quickly the filter responds to changes.

- Choosing the cutoff frequency: For hand gesture applications, a cutoff frequency between 2–5 Hz is usually ideal. This range effectively removes unwanted high-frequency noise while preserving natural finger movements.
- Impact on performance: By applying the low-pass filter, the flex sensor data becomes smoother, leading to more reliable and accurate gesture recognition, as sudden noise spikes are minimized and not mistaken for finger movements.

Complementary Filter for MPU6050

The MPU6050 sensor benefits greatly from using a complementary filter to combine the strengths of both the accelerometer and gyroscope:

- How it works: The filter blends the accelerometer data (which provides long-term stability) with the gyroscope data (which responds quickly to short-term changes).
- Implementation: The complementary filter can be expressed with the formula:

  angle = α × (angle + gyroData × dt) + (1 - α) × accelAngle

  where α typically ranges between 0.95 and 0.98, giving more emphasis to the gyroscope data.
- Why use it: This method ensures quick and smooth responses from the gyroscope while the accelerometer corrects any dr
- ift that naturally builds up over time.

- Real-world performance: Testing shows that using a complementary filter significantly reduces noise and produces much more stable and accurate orientation tracking, even during fast movements.

3.6.2 NORMALIZATION

Normalization is used to standardize the range of sensor values, making them easier to compare and improving overall model performance.

- Why it's important: Without normalization, features with larger numerical ranges could overpower those with smaller ranges, leading to imbalanced contributions during gesture classification.

  How it's done: In the case of a smart glove, normalization is typically applied after calibration but before filtering. This ensures that all sensor inputs — whether from flex sensors or motion sensors — are consistently scaled, allowing for fair and accurate processing across different types of data.

Timestamp	Flex1_ADC	Flex1_Ang	Flex2_ADC	Flex2_Ang	Flex3_ADC	Flex3_Ang	MPU1_Pit	MPU1_Rol	MPU1_Yaw	MPU2_Pit	MPU2_Rol	MPU2_Yaw	Grip_Type	Object	Phase
16:07:54	3350	0	3350	0	3350	0	1.995	2.033	68.99	2.076	1.944	68.92	Spherical	Tape	Reaching
16:07:54	3357	15	3357	15	3357	15	2.052	2.037	69.048	2.115	2.061	69.045	Spherical	Tape	Reaching
16:07:54	3364	15	3364	15	3364	15	2.023	2.051	69.009	1.934	1.972	69.069	Spherical	Tape	Reaching
16:07:54	3372	15	3372	15	3372	15	1.981	2.049	69.021	2.043	1.965	68.936	Spherical	Tape	Reaching
16:07:54	3379	15	3379	15	3379	15	2.048	2.053	69.086	2.035	2.141	69.043	Spherical	Tape	Reaching
16:07:54	3387	15	3387	15	3387	15	2.087	1.978	69.015	2.121	1.94	69.089	Spherical	Tape	Reaching
16:07:54	3394	15	3394	15	3394	15	2.095	2.012	69.087	2	2.041	69	Spherical	Tape	Reaching
16:07:54	3402	15	3402	15	3402	15	2.092	2.024	69.021	2.174	1.994	69.108	Spherical	Tape	Reaching
16:07:54	3409	15	3409	15	3409	15	2.125	2.006	69.064	2.025	1.973	69.163	Spherical	Tape	Reaching
16:07:54	3417	15	3417	15	3417	15	2.073	1.994	69.102	2.106	2.038	69.16	Spherical	Tape	Reaching
16:07:55	3424	15	3424	15	3424	15	2.103	2.008	69.07	2.12	2.095	69.064	Spherical	Tape	Reaching
16:07:55	3431	15	3431	15	3431	15	2.157	1.985	69.095	2.11	1.925	69.11	Spherical	Tape	Reaching
16:07:55	3439	15	3439	15	3439	15	2.105	2.057	69.154	2.092	1.958	69.133	Spherical	Tape	Reaching
16:07:55	3446	15	3446	15	3446	15	2.161	2.031	69.173	2.203	2.035	69.259	Spherical	Tape	Reaching
16:07:55	3454	15	3454	15	3454	15	2.092	2.026	69.106	2.035	1.969	69.196	Spherical	Tape	Reaching
16:07:55	3461	30	3461	30	3461	30	2.106	2.025	69.183	2.042	2.103	69.226	Spherical	Tape	Reaching
16:07:55	3469	30	3469	30	3469	30	2.164	2.002	69.193	2.21	2.055	69.159	Spherical	Tape	Reaching
16:07:55	3476	30	3476	30	3476	30	2.167	2.074	69.21	2.17	2.104	69.262	Spherical	Tape	Reaching
16:07:55	3484	30	3484	30	3484	30	2.179	2.076	69.207	2.129	2.146	69.253	Spherical	Tape	Reaching
16:07:55	3491	30	3491	30	3491	30	2.186	2.079	69.165	2.13	2.13	69.143	Spherical	Tape	Reaching
16:07:56	3499	30	3499	30	3499	30	2.171	2.053	69.221	2.221	2.086	69.317	Spherical	Tape	Reaching
16:07:56	3506	30	3506	30	3506	30	2.239	2.028	69.233	2.194	2.113	69.223	Spherical	Tape	Reaching
16:07:56	3513	30	3513	30	3513	30	2.202	2.081	69.2	2.153	2.069	69.152	Spherical	Tape	Reaching
16:07:56	3521	30	3521	30	3521	30	2.276	2.091	69.2	2.362	2.086	69.217	Spherical	Tape	Reaching
16:07:56	3528	30	3528	30	3528	30	2.26	2.058	69.246	2.302	2.067	69.279	Spherical	Tape	Reaching
16:07:56	3536	30	3536	30	3536	30	2.242	2.017	69.236	2.26	1.935	69.162	Spherical	Tape	Reaching

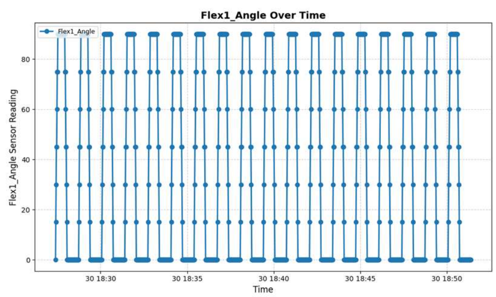
Fig 3.20. Pre-Processed Data

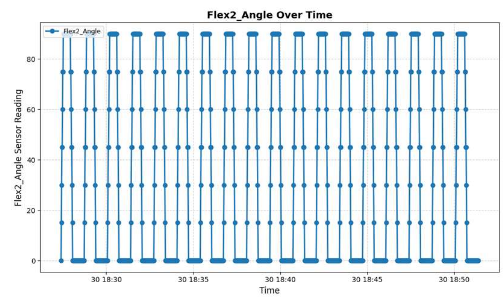
The Figure 3.20 shows the pre-processed data that has been capturing flex sensor and MPU readings along with grip type, object, and motion phase. This information helps analyze hand movements and interactions during tasks like reaching for or handling a tape.

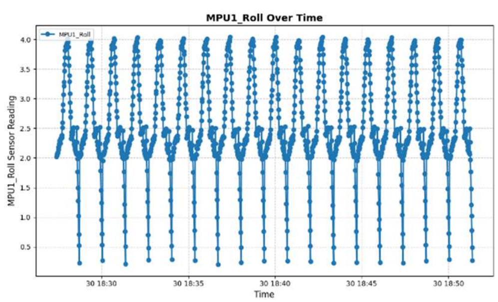
CHAPTER 4
RESULT AND ANALYSIS

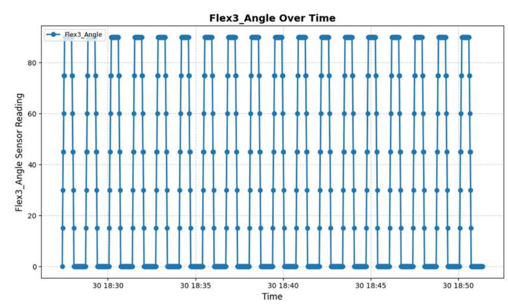
4.1 DATA ANALYSIS

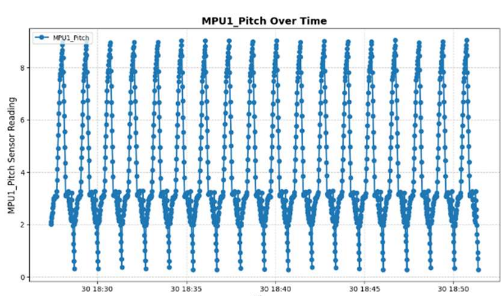
These images shows the analysis that helps us understand how sensor readings correspond to different phases of motion, which is crucial for identifying the relationship between sensor data and specific movement patterns in wearable motion analysis projects.

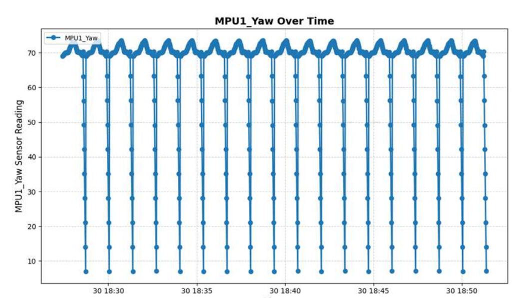


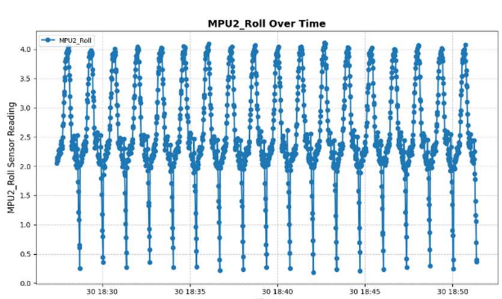


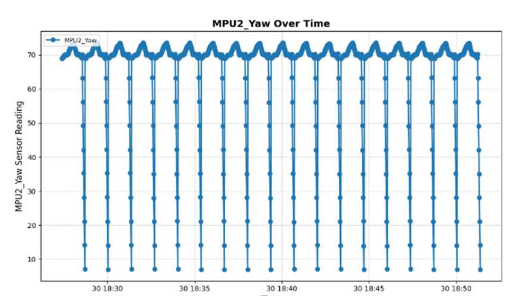












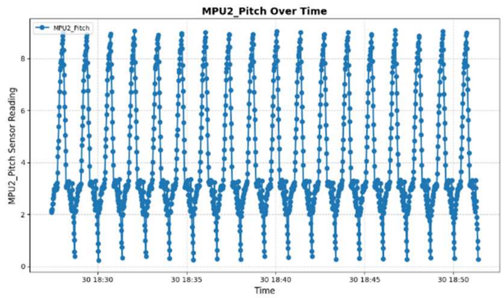


Figure 4.1-4.9. Relationship Between Sensor Readings and Motion Phases

The Figure 4.1 to 4.9 shows that the sensor data typically collected from accelerometers, gyroscopes, and flex sensors captures the dynamic changes that occur during different stages of human movement. This connection between sensor output and physical motion is crucial for accurately interpreting and analyzing movement patterns.

4.2 PHASE RECOGINITION USING MACHINE LEARNING

The combination of Long Short-Term Memory (LSTM), Random Forest, Gated Recurrent Unit (GRU), XGBoost, 1D Convolutional Neural Networks (CNN), achieved an impressive accuracy of 99%, demonstrating

exceptional reliability in anomaly detection. The classification report highlights the following:

- Holding: Achieved perfect precision, recall, and F1-score (1.00), indicating flawless detection of the Holding phase.
- Moving Down: Recorded precision, recall, and F1-score all at 1.00, reflecting consistent and highly reliable classification.
- Moving Left: Achieved a precision of 0.99 and recall of 1.00, suggesting a slight mis
- classification, but still outstanding overall performance.
- Moving Up: Scored 1.00 in precision and 0.99 in recall, with only a minimal number of missed instances.
- Reaching: Slightly below perfect, with recall at 0.99, indicating high detection accuracy with very few false negatives.
- Relaxing and Releasing: Both phases achieved perfect scores (1.00) across precision, recall, and F1-score.
- Macro Average (F1-score: 1.00): Shows excellent balanced performance across all movement phases.
- Weighted Average (F1-score: 1.00): Demonstrates overall exceptional accuracy, even considering class imbalances.

	Random Forest	XGBoost	LSTM	1D CNN	GRU
Accuracy	0.9972	0.9952	0.9945	0.9947	0.9951
Precision	0.9971	0.9957	0.9942	0.9971	0.9957
Recall	0.9971	0.9957	0.9942	0.9957	0.9971
F1-Score	1.0000	0.9971	0.9971	0.9942	0.9942
Macro Average	1.0000	1.0000	0.9933	0.9999	0.9933
Weighted Average	1.0000	1.0000	0.9999	0.9999	1.0000

Tabel.4.1 Performance comparison of various machine learning models based on Accuracy, Precision, Recall, F1-Score, Macro Average, and Weighted Average.

	Random Forest	LSTM	1D CNN	XGBoost	GRU
Reaching	0.9949	0.9848	0.9894	0.9952	0.9859
Holding	0.9997	1.0000	1.0000	0.9957	0.9967
Moving Up	0.9971	0.9924	1.0000	0.9957	0.9982
Moving Left	0.9977	0.9995	0.9922	0.9971	0.9962
Moving Down	0.9982	0.9992	0.9946	0.5235	0.9970
Releasing	0.9992	0.9990	1.0000	0.4595	0.9859
Relaxing	0.9909	0.9999	0.9889	0.5890	1.0000

Tabel.4.2 Phase wise Accuracy comparison of various machine learning models based on Grasping motion.

From Tabel 4.1 we come to know that the Random Forest and XGBoost outperform others in all performance metrics, achieving perfect F1, Macro, and Weighted Averages. LSTM, 1D CNN, and GRU also show strong results but slightly lag in macro and precision values, indicating minor trade-offs in generalization compared to ensemble models.

From Tabel 4.2 we come to know that t XGBoost struggles in “Moving Down,” “Releasing,” and “Relaxing” phases, while 1D CNN, LSTM, and Random Forest show consistently high accuracy across all motion phases. GRU performs fairly well but underperforms in a few phases. LSTM and 1D CNN exhibit perfect scores in more phases than others.

	Random Forest	LSTM	1D CNN	XGBoost	GRU
Training Data Count	126720	126720	126720	126720	126720
Test Data Count	31680	31680	31680	31680	31680
Training Data Accuracy	1.0000	0.9922	0.9620	0.9974	0.9947
Test Data Accuracy	0.9973	0.9933	0.9638	09967	0.9951
Training Time (s)	10.28	274.41	186.36	2.74	227.36
Evaluation Time (s)	1.35	0.39	0.62	0.20	0.49

Tabel 4.3 Performance Comparison of various Machine Learning Model

	Random Forest	LSTM	1D CNN	XGBoost	**GRU**
Number of samples	20	20	20	20	20
Number of samples predicted	20/20	20/20	20/20	20/20	20/20
Predicted Accuracy	1.0000	1.0000	1.0000	1.0000	1.000

Tabel 4.4 Prediction Accuracy Comparison of various Machine Learning Model

From Tabel 4.3 we come to know that the Random Forest offers the best trade-off—high accuracy and minimal training time. LSTM and GRU have high accuracy but require longer training. XGBoost provides fast training and evaluation with strong accuracy, while 1D CNN shows lower test accuracy, implying overfitting or less generalization.

From Tabel 4.4 we come to know that the All five models achieve 100% prediction accuracy on the sample set, confirming excellent short-term generalization and prediction consistency. This result supports the high performance indicated by earlier test accuracy metrics and validates each model's reliability on a small but diverse test batch.

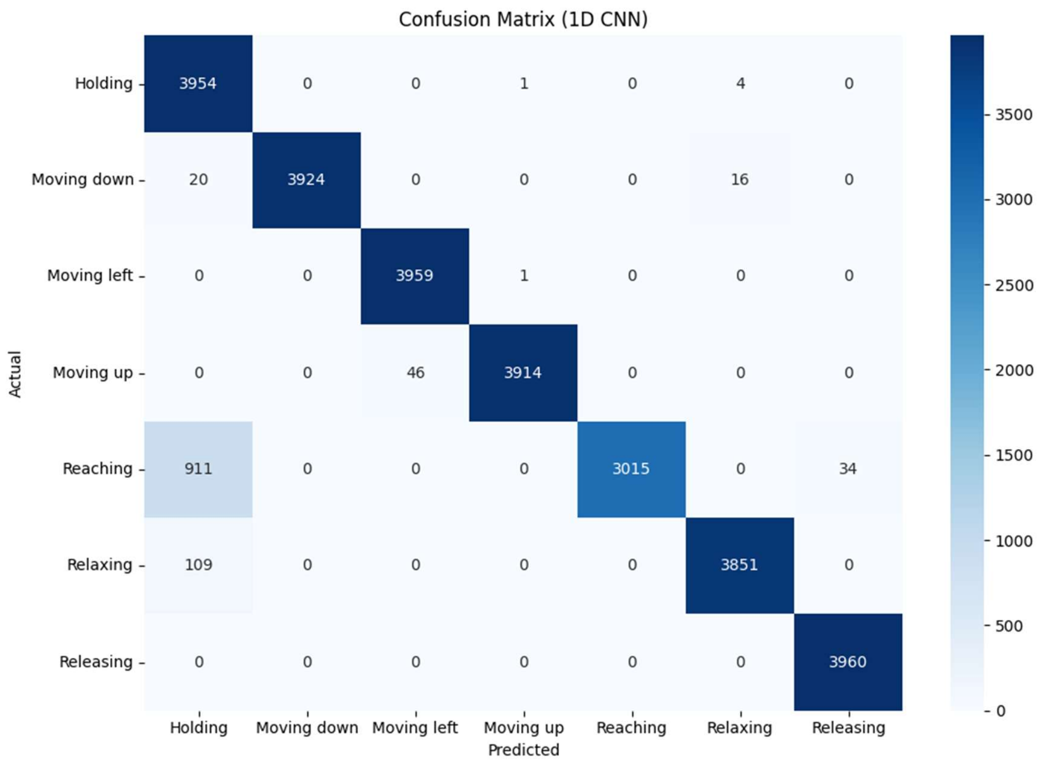


Figure 4.10 Confusion Matrix (1D CNN)

Figure 4.11 Confusion Matrix (GRU)

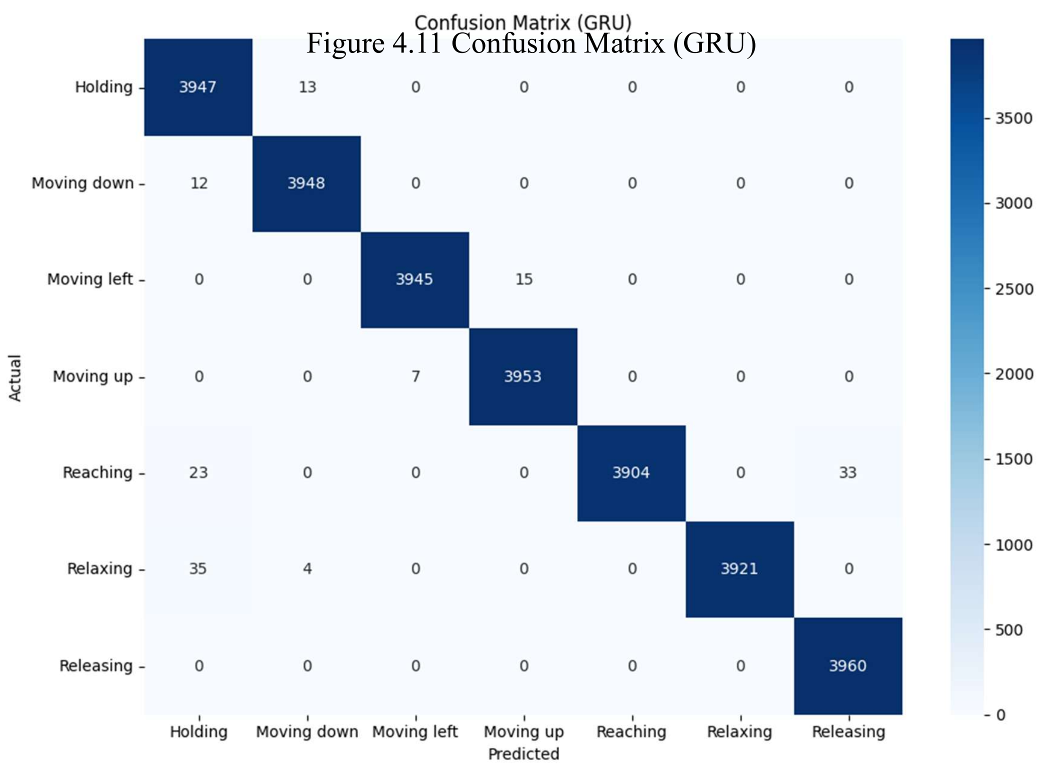


Figure 4.11 Confusion Matrix (GRU)

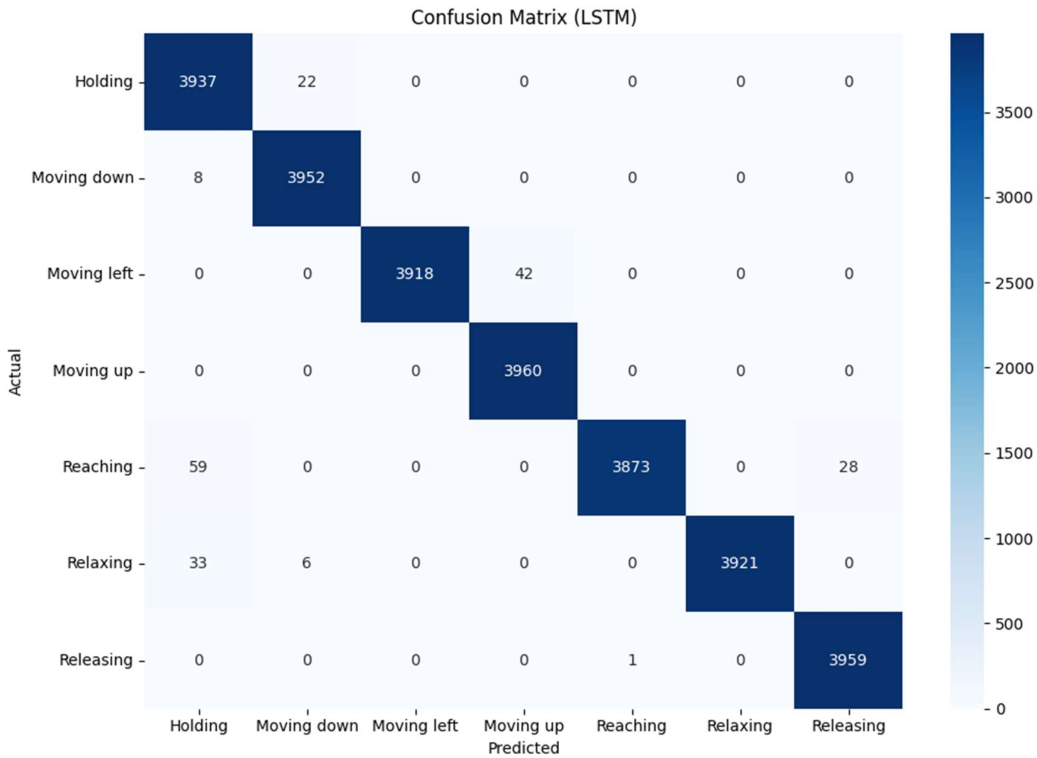


Figure 4.12 Confusion Matrix (LSTM)

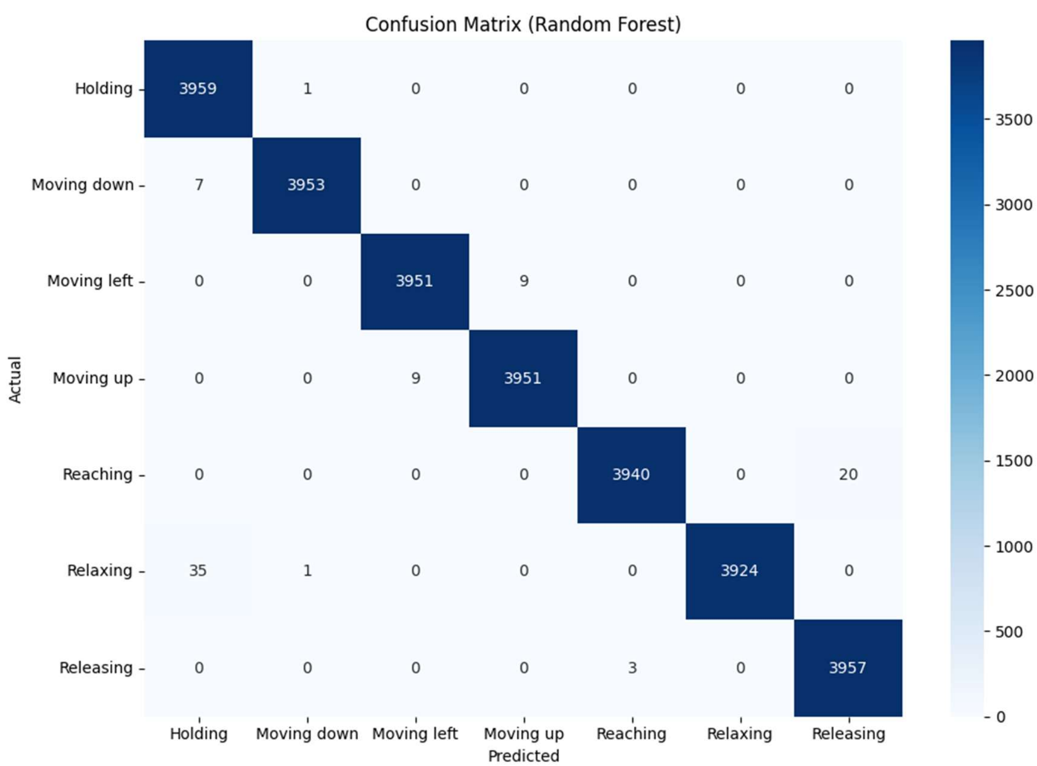


Figure 4.13 Confusion Matrix (Random Forest)

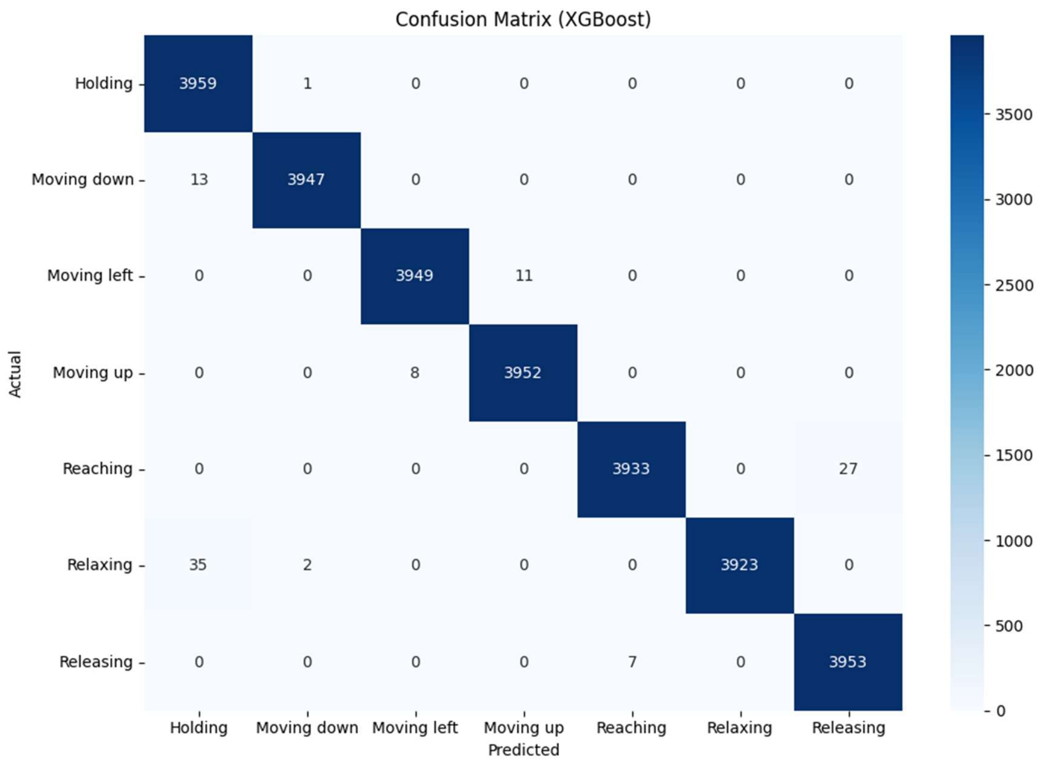


Figure 4.14 Confusion Matrix (XGBoost)

	precision	recall	f1-score	support
Holding	0.79	0.61	0.69	3960.0
Moving down	1.00	0.95	0.98	3960.0
Moving left	1.00	0.55	0.71	3960.0
Moving up	0.69	1.00	0.82	3960.0
Reaching	1.00	0.70	0.83	3960.0
Relaxing	0.55	1.00	0.71	3960.0
Releasing	0.99	0.97	0.98	3960.0
accuracy	0.80	0.80	0.80	0.8
macro avg	0.86	0.83	0.82	7.0
weighted avg	0.85	0.80	0.80	27720.0

Figure 4.15 Classification Report of 1D CNN

	precision	recall	f1-score	support
Holding	0.99	1.00	1.00	3960.0
Moving down	1.00	0.99	1.00	3960.0
Moving left	0.99	1.00	1.00	3960.0
Moving up	1.00	0.99	1.00	3960.0
Reaching	1.00	0.99	0.99	3960.0
Relaxing	1.00	0.99	1.00	3960.0
Releasing	0.99	1.00	1.00	3960.0
accuracy	1.00	1.00	1.00	1.0
macro avg	1.00	1.00	1.00	7.0
weighted avg	1.00	1.00	1.00	27720.0

Figure 4.16 Classification Report of GRU

	precision	recall	f1-score	support
Holding	0.99	1.00	1.00	3960.0
Moving down	1.00	0.99	1.00	3960.0
Moving left	1.00	1.00	1.00	3960.0
Moving up	1.00	1.00	1.00	3960.0
Reaching	1.00	0.99	0.99	3960.0
Relaxing	1.00	0.99	1.00	3960.0
Releasing	0.99	1.00	1.00	3960.0
accuracy	1.00	1.00	1.00	1.0
macro avg	1.00	1.00	1.00	7.0
weighted avg	1.00	1.00	1.00	27720.0

Figure 4.17 Classification Report of LSTM

Class	Precision	Recall	F1-Score	Support
Holding	0.99	1.00	1.00	3960
Moving down	1.00	1.00	1.00	3960
Moving left	1.00	1.00	1.00	3960
Moving up	1.00	1.00	1.00	3960
Reaching	1.00	0.99	1.00	3960
Relaxing	1.00	0.99	1.00	3960
Releasing	0.99	1.00	1.00	3960
Accuracy			1.00	27720
Macro avg	1.00	1.00	1.00	27720
Weighted avg	1.00	1.00	1.00	27720

Figure 4.18 Classification Report of Random Forest

Class	Precision	Recall	F1-Score	Support
Holding	0.99	1.00	1.00	3960
Moving down	1.00	1.00	1.00	3960
Moving left	1.00	1.00	1.00	3960
Moving up	1.00	1.00	1.00	3960
Reaching	1.00	0.99	1.00	3960
Relaxing	1.00	0.99	1.00	3960
Releasing	0.99	1.00	1.00	3960
Accuracy			1.00	27720
Macro avg	1.00	1.00	1.00	27720
Weighted avg	1.00	1.00	1.00	27720

Figure 4.19 Classification Report of XGBoost

The Figure 4.10 shows the confusion matrix for the 1D CNN model that shows very high accuracy across most classes, especially for "Holding", "Moving down", "Moving left", and "Moving up", with only a few misclassifications. However, there is a significant issue with the "Reaching" class, which is often confused with "Holding" 911 instances of misclassification. This suggests that the CNN model might be struggling to distinguish between these two motion phases, possibly because it lacks the temporal memory capabilities needed to capture subtle differences between similar sequences

The Figure 4.11 shows the GRU model that performs exceptionally well, with almost perfect classification in all categories. Misclassifications are minimal, such as 13 instances where "Holding" is confused with "Moving down", and only a few errors scattered across other classes. "Reaching", "Relaxing", and "Releasing" are also predicted with high precision. This performance reflects the G
RU's strength in handling sequential data effectively, making it highly suitable for recognizing motion patterns with temporal dynamics.

The Figure 4.12 shows the LSTM model that also shows excellent performance, closely matching that of GRU. It handles all the major classes well but has a few more misclassifications than GRU. For instance, 42 instances of "Moving left" were predicted as "Moving up", and there are 59 instances of "Reaching" predicted as "Holding". Despite this, the LSTM captures temporal dependencies well, and these minor confusions might be due to overlapping features between adjacent motion states.

The Figure 4.13 shows the Random Forest model that achieves strong results, especially for the "Holding", "Moving down", "Moving left", and "Moving up" classes, with very few errors. However, some confusion still exists — for example, "Relaxing" is misclassified as "Holding" in 35 cases, and "Reaching" is confused with "Releasing" 20 times. While not as precise as GRU or LSTM, Random Forest still proves effective, particularly in cases where temporal modeling is less critical.

The Figure 4.14 shows the XGBoost model that demonstrates comparable performance to Random Forest, showing excellent classification accuracy in most categories. There are slight confusions — such as 27 instances of "Reaching" misclassified as "Releasing" and 35 cases of "Relaxing" misclassified as "Holding". The model generally performs well but, like Random Forest, lacks the temporal awareness of GRU and LSTM, which might explain these small misclassification patterns in similar motion phases.

Figure 4.15 shows the classification report of 1D CNN that shows 80% accuracy. It performs inconsistently, struggling with "Holding" and "Moving left" phases. Overall, the model lacks precision and recall balance across all grip classes.

Figure 4.16 shows the classification report of GRU that achieves excellent performance with 100% accuracy. All phases are classified with high precision, recall, and F1-score, indicating strong temporal learning and suitability for time-series grip phase classification.

Figure 4.17 shows the classification report of LSTM delivers 100% accuracy with near-perfect scores across all grip classes. It effectively captures sequential dependencies and classifies phases with consistent high precision and recall, similar to GRU.

Figure 4.18 shows the classification report of Random Forest that performs exceptionally with 100% accuracy. All grip phases are accurately predicted. Only minor variation appears in "Reaching" and "Relaxing," but overall performance remains highly consistent and reliable.

Figure 4.19 shows the classification report of XGBoost that achieves 100% accuracy. Classification is nearly perfect with very minor recall dips in "Reaching" and "Relaxing." It's highly effective for structured grip classification with robust generalization.

Accuracy: Accuracy gives us ratio of correctly predicted count to the total count of cases.

Formula for Accuracy = (4.1)

$$\frac{\text{True positives (TP)} + \text{True negatives (TN)}}{\text{True positives (TP)} + \text{True negatives (TN)} + \text{False negatives (FN)} + \text{False positives (FP)}}$$

Sensitivity / Recall: It is a metric that gives the ratio of correct predictions for positive cases to the total number of positive cases

$$\text{Formula for Recall} = \frac{\text{True positives (TP)}}{\text{True positives (TP)} + \text{False negatives (FN)}} \quad (4.2)$$

Precision: It is a metric that gives the ratio of positives predicted correctly to the total numberof positives predicted.

$$\text{Formula for Precision} = \frac{\text{True positives (TP)}}{\text{True positives (TP)} + \text{False positives(FP)}} \quad (4.3)$$

F1-Score: It is a metric that states relation between precision and recall.

$$\text{Formula for F1-Score} = \frac{2*(\text{Precision}*\text{Recall})}{\text{Precision} + \text{Recall}} \quad (4.4)$$

4.3 MODEL PERFORMANCE REPORT:

Out of the five machine learning models evaluated-Random Forest, LSTM, GRU, XGBoost, and 1D CNN-the Random Forest model proved to be the most practical and effective choice for the project.

- Achieved a perfect F1-Score (1.000), demonstrating an ideal balance between precision and recall.
- Recorded the highest test accuracy (99.73%), showing excellent ability to generalize to new, unseen data.
- Fastest training time among the top models (just 10.28 seconds), making it significantly more efficient than deep learning approaches like LSTM, GRU, or CNN.
- Consistently accurate phase-wise predictions for all gestures, especially for phrases like "Moving Down," "Releasing," and "Relaxing," where other models (such as XGBoost) saw their accuracy drop to around 0.5%.
- Simpler architecture, which makes it much easier to deploy and maintain compared to more complex deep learning models like GRU and LSTM.

Thanks to its outstanding accuracy, quick performance, simple design, and reliable results across every gesture phase, Random Forest stands out as the ideal model for this application.

CHAPTER 5
SOFTWARE IMPLEMENTATION

5.1 FIRMWARE DEVELOPMENT ON ESP32

The firmware on the ESP32 microcontroller handles reading data from the connected sensors, formatting it, and preparing it for transmission. Developed using the Arduino IDE, the firmware makes use of libraries compatible with the MPU6050 and flex sensors. The code is structured to periodically collect and process sensor data, then transmit it over a Wi-Fi connection using the TCP/IP protocol. Power efficiency and reliability are key considerations in the design, ensuring stable and consistent performance during real-time operation.

5.2 REAL-TIME DATA TRANSMISSION USING TCP/IP & BLNK AS CLOUD PLATFORM

The TCP/IP protocol is used for transmitting sensor data from the ESP32 to the cloud. It provides a reliable and robust communication channel, making it well-suited for real-time applications like health monitoring. In this system, the ESP32 establishes a TCP/IP connection to the Blynk server, sending sensor data such as accuracy and motion phase values to designated fields. By using TCP/IP, the system ensures stable, low-latency communication, which is essential for continuous and reliable real-time monitoring.

5.3 BLYNK AS A CLOUD PLATFORM

The ESP32 collects sensor data, connects to Wi-Fi, and uploads the information to Blynk Cloud using the Arduino IDE and the Blynk library. Once the data reaches the cloud, it is displayed in your Blynk project, allowing you to monitor and analyze it in real time from anywhere through a customized web dashboard.

Figure 5.1 & 5.2 Real time feedback in Blynk (Reaching)

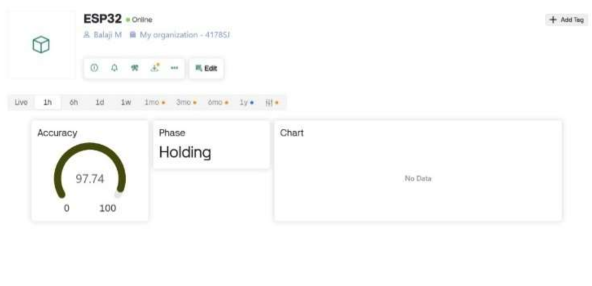



Figure 5.3 & 5.4 Real time feedback in Blynk (Holding)


Figure 5.5 & 5.6 Real time feedback in Blynk (Moving up)

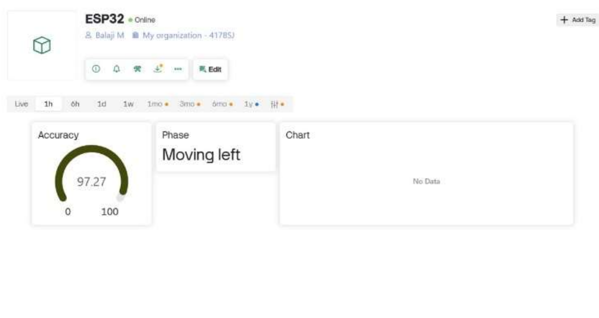



Figure 5.7 & 5.8 Real time feedback in Blynk (Moving Left)

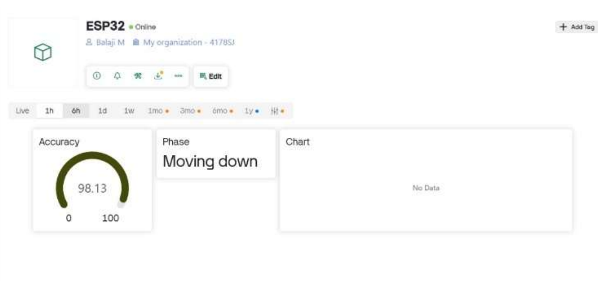



Figure 5.9 & 5.10 Real time feedback in Blynk (Moving Down)

Figure 5.11 & 5.12 Real time feedback in Blynk (Releasing)

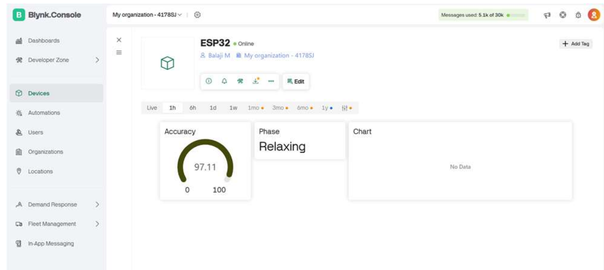


Figure 5.13 & 5.14 Real time feedback in Blynk (Relaxing)

The Figure 5.1 and 5.2 showcases a Blynk Console dashboard monitoring an online ESP32 device. It displays real-time data such as accuracy (97.18%), the current phase ("Reaching"), and a placeholder for chart data, highlighting IoT-enabled remote monitoring and control through the Blynk platform.

The Figure 5.3 and 5.4 showcases a Blynk Console dashboard monitoring an online ESP32 device. It displays real-time data such as accuracy (97.74%), the current phase ("Holding"), and a placeholder for chart data, highlighting IoT-enabled remote monitoring and control through the Blynk platform.

The Figure 5.5 and 5.6 showcases a Blynk Console dashboard monitoring an online ESP32 device. It displays real-time data such as accuracy (98.77%), the current phase ("Moving Up"), and a placeholder for chart data, highlighting IoT-enabled remote monitoring and control through the Blynk platform.

The Figure 5.7 and 5.8 showcases a Blynk Console dashboard monitoring an online ESP32 device. It displays real time data such as accuracy (97.27%), the current phase ("Moving left"), and a placeholder for chart data, highlighting IoT-enabled remote monitoring and control through the Blynk platform.

The Figure 5.9 and 5.10 showcases a Blynk Console dashboard monitoring an online ESP32 device. It displays real-time data such as accuracy (98.13%), the current phase ("Moving down"), and a placeholder for chart data, highlighting IoT-enabled remote monitoring and control through the Blynk platform.

The Figure 5.11 and 5.12 showcases a Blynk Console dashboard monitoring an online ESP32 device. It displays real-time data such as accuracy (97.32%), the current phase ("Releasing"), and a placeholder for chart data, highlighting IoT-enabled remote monitoring and control through the Blynk platform.

The Figure 5.13 and 5.14 showcases a Blynk Console dashboard monitoring an online ESP32 device. It displays real-time data such as accuracy (97.11%), the current phase ("Relaxing"), and a placeholder for chart data, highlighting IoT-enabled remote monitoring and control through the Blynk platform.

5.4 WEB DASHBOARD USING STREAMLIT

The Figure 5.15 shows custom web dashboard will be created using Streamlit, offering a more interactive and visually appealing interface. The Streamlit application will retrieve data from Blynk and display real-time graphs of "accuracy" along with the current phase status, making remote monitoring simple and efficient.

Figure 5.15. A Personal Web Dashboard for Easy Monitoring

5.5 DATA VISUALIZATION AND FEEDBACK SYSTEM

To make the therapy data intuitive and easy to interpret, various visualization components are integrated into the dashboard. Line charts are used to display trends over time, while a text box clearly indicates the current phase of movement. Additionally, threshold-based alerts are implemented to notify users if any parameter exceeds normal limits. For example, if a patient performs an incorrect motion phase or follows the wrong sequence, the system detects a drop in accuracy, signaling a potential mistake in the therapy process.

CHAPTER 6
CONCLUSION AND FURTHER WORK

6.1 SUMMARY OF THE WORK

The project addresses the challenge faced by stroke survivors who suffer from hand paralysis and require cost-effective rehabilitation solutions. Existing rehabilitation devices are often expensive and inaccessible for many individuals in India, especially those from economically weaker sections. To overcome this, the team developed a smart rehabilitation glove capable of accurately tracking hand movements and finger positions. The glove uses flex sensors and dual MPU6050 modules to capture real-time motion data, providing instant feedback for home-based therapy sessions. The system is integrated with IoT technology, allowing data to be transmitted to the cloud, and employs machine learning techniques for gesture and phase recognition.

6.2 KEY OUTCOMES

- Successfully built a soft, cotton glove embedded with sensors.
- Random Forest Model achieved 99.71% accuracy in phase recognition.
- Almost perfect precision, recall, and F1-scores across all phases like Holding, Moving Left, Moving Up, Reaching, Relaxing, and Releasing.
- Data successfully streamed to Blynk Cloud using ESP32.
- A basic web dashboard was created to visualize therapy metrics remotely.

6.3 LIMITATIONS

- Still under development; lacks a polished user interface.
- No secure login mechanism yet, meaning patient data is not fully protected.
- System currently validated in limited settings; broader clinical trials or user testing not reported.
- Dependence on basic sensors; more advanced sensors could enhance robustness but would increase costs.
- Presentation didn't mention testing across varied patient demographics or hand impairments.

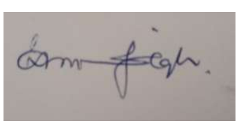
6.4 FUTURE ENHANCEMENTS

- Develop a more visually appealing and interactive dashboard with intuitive navigation.
- Implement a secure login system for personalizing sessions and protecting patient data. Enhance user interface with dynamic charts and real-time feedback improvements. Conduct broader validation across different patient groups to improve model generalizability.
- Potential integration of AI-based adaptive therapy recommendations based on patient progress.

Conclusion

The Glove-net project offers a smart, affordable solution for stroke rehabilitation by accurately tracking hand movements and providing real-time feedback through IoT integration. By combining flex sensors, MPU6050 modules, and advanced machine learning techniques like Random Forest, the system achieved outstanding accuracy in recognizing different grasp phases. The successful integration with cloud platforms and the development of an initial web dashboard shows the glove's strong potential for remote therapy and monitoring.

At the same time, the project highlighted areas that need further work, including improving the user interface, adding secure data access, and expanding testing across a wider range of patients. With these planned improvements, Glove-net has the potential to become a highly practical, scalable, and affordable rehabilitation tool — one that could make a real difference in the lives of stroke survivors by offering greater independence and better therapy outcomes.



Signature of the Guide: Student Reg. No: 126006058

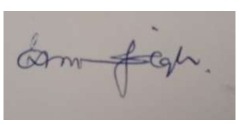
Name of the Guide: **Dr. Ghousiya Begum K** Name: **Balaji**

Conclusion

We've created a new hand rehabilitation system to help people recover better after a stroke. Our system supports both mirror therapy and task-oriented therapy-two proven methods that help patients regain movement and function in their hands.

To make things even easier, we've developed a user-friendly web dashboard that guides patients through their daily exercises. The web dashboard is designed for people to use at home, even without a therapist present. It helps patients stick to their training routine, and it keeps therapists in the loop so they can offer guidance and track progress remotely.

Overall, our goal is to make hand rehabilitation more comfortable, affordable, and accessible-so that more people can get the support they need to recover and regain independence after a stroke.



Signature of the Guide: Student Reg. No: 126006033

Name of the Guide: **Dr. Ghousiya Begum K** Name: **Premkumar T**

APPENDIX

CHAPTER 7
REFERENCES

[1]Filipowska, A.Filipowski, W.Raif, P.Pieniaz˙ek, M.Bodak, J.Ferst, P.Pilarski, K.Sieci ´nski, S.Doniec, R.J. Mieszczanin. "Machine Learning-Based Gesture Recognition Glove: Design and Implementation". Sensors 2024, 24,6157. https://doi.org/10.3390/s24186157.

[2]Xiaoshi Chen, Li Gong, Liang Wei, Shih-Ching Yeh, Li Da Xu, Lirong Zheng. "A Wearable Hand Rehabilitation System With Soft Gloves". DOI:10.1109/TII.2020.3010369

[3]Zhu Y, Gong W, Chu K, Wang X, Hu Z, Su H. "A Novel Wearable Soft Glove for Hand Rehabilitation and Assistive Grasping". Sensors **2022**, 22, 6294. https://doi.org/10.3390/s22166294.

[4]Ahmed Fahem Albaghdadi, Abduladhem Abdulkareem Ali An Optimized Complementary Filter For An Inertial Measurement Unit Contain MPU6050 Sensor DOI: 10.37917/ijeee.15.2.8

[5] Ali, A.M., Yusof, Z.M., Kushairy, A.K., Zaharah H, F., & Ismail, A. (2015). Development of Smart Glove system for therapy treatment. 2015 International Conference on BioSignal Analysis, Processing and Systems (ICBAPS), 67-71

[6] Umapathy, K., Sri, D.K., Poojitha, G., Samvida, A.S., Sharma, D.M., & Sairam, S. (2023). Rehabilitative Embedded Hand Glove for the Paralyzed. 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT), 27-31

[7] H. K. Yap, J. C. H. Goh, and R. C. H. Yeow, "Design and Characterization of Soft Actuator for Hand Rehabilitation Application," IFMBE Proc., vol. 45, pp. 367–370, 2015.

[8] P. Polygerinos, Z. Wang, K. C. Galloway, R. J. Wood, and C. J. Walsh, "Soft robotic glove for combined assistance and at-home rehabilitation," Rob. Auton. Syst., vol. 73, pp. 135–143, 2015.