BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING

Bengali Air Writing Using Wearable Wireless Motion Sensors

Md. Rafiur Rahman
200042167
Zubayer Tahmid
200042136
Abrar Mahabub Nowrid
200042103

Department of Computer Science and Engineering
Islamic University of Technology

September, 2025

Bengali Air Writing Using Wearable Wireless Motion Sensors

Md. Rafiur Rahman
200042167
Zubayer Tahmid
200042136
Abrar Mahabub Nowrid
200042103

Department of Computer Science and Engineering
Islamic University of Technology
September, 2025

Declaration of Candidate

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by **Md. Rafiur Rahman**, **Zubayer Tahmid**, and **Abrar Mahabub Nowrid** under the supervision of **Mohammad Ridwan Kabir**, Assistant Professor - On Study Leave, Department of Computer Science and Engineering and co-supervision of **Dr. Hasan Mahmud**, Professor, Department of Computer Science and Engineering, Islamic University of Technology, Dhaka, Bangladesh. It is also declared that neither this thesis nor any part of it has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others have been acknowledged in the text and a list of references is given.

Mohammad Ridwan Kabir
Assistant Professor - On Study Leave
Department of Computer Science and Engineering
Islamic University of Technology (IUT)
Date: September 30, 2025

Dr. Hasan Mahmud
Professor
Department of Computer Science and Engineering
Islamic University of Technology (IUT)
Date: September 30, 2025

Md. Rafiur Rahman
Student ID: 200042167
Date: September 30, 2025

Zubayer Tahmid
Student ID: 200042136
Date: September 30, 2025

Abrar Mahabub Nowrid
Student ID: 200042103
Date: September 30, 2025

Dedicated to our supervisor(s), without whom this study would not have been possible

Contents

List of Figures

List of Tables

List of Abbreviations

ACC	Accelerometer
Adam	Adaptive Moment Estimation (optimizer)
AUC	Area Under the Curve
AUROC	Area Under the Receiver Operating Characteristic Curve
Bn	Bengali (language/script)
BPF	Band-Pass Filter
CER	Character Error Rate
CPU	Central Processing Unit
CSV	Comma-Separated Values
CTC	Connectionist Temporal Classification
CV	Cross-Validation
DoF	Degrees of Freedom
DTW	Dynamic Time Warping
ECE	Expected Calibration Error
EMA	Exponential Moving Average
FFT	Fast Fourier Transform
FN	False Negative
FP	False Positive
GRU	Gated Recurrent Unit
GNB	Gaussian Naïve Bayes
GPU	Graphics Processing Unit
GYR	Gyroscope
HCI	Human–Computer Interaction
HMM	Hidden Markov Model
HPF	High-Pass Filter
HSMM	Hidden Semi-Markov Model
Hz	Hertz (samples per second)
IMU	Inertial Measurement Unit
IQR	Interquartile Range
JSON	JavaScript Object Notation
k-NN	k-Nearest Neighbors

LR	Logistic Regression
LOSO	Leave-One-Subject-Out (cross-validation)
LPF	Low-Pass Filter
LSTM	Long Short-Term Memory
mF1	Macro-*F1* Score
PSD	Power Spectral Density
PR	Precision–Recall
RF	Random Forest
RMS	Root Mean Square
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SE(3)	SE(3), Special Euclidean Group in 3D (rigid motions)
SMA	Signal Magnitude Area
SNR	Signal-to-Noise Ratio
STFT	Short-Time Fourier Transform
SVM	Support Vector Machine
TCN	Temporal Convolutional Network
TN	True Negative
TP	True Positive
TTA	Test-Time Adaptation
t-SNE	t-Distributed Stochastic Neighbor Embedding
UI	User Interface
UTC	Coordinated Universal Time
ZCR	Zero-Crossing Rate

Acknowledgement

We extend our deepest gratitude to our esteemed supervisor, Ridwan Kabir, Assistant Professor, Systems and Software Lab, Department of Computer Science and Engineering, Islamic University of Technology. His unwavering support, invaluable guidance and exceptional dedication have been the cornerstone of our research journey. We are especially thankful for the countless hours he devotedoften beyond regular working hours, on weekends and late nightsto patiently listen to our ideas, provide constructive feedback and offer insightful alternatives whenever we faced challenges.

More than just academic guidance, his constant encouragement and mental support gave us the strength to persevere, even during the most difficult phases. His belief in our potential uplifted our spirits and inspired us to push our limits. Learning under his mentorship has not only shaped our research but also deeply influenced our personal and academic growth. We are truly honored and grateful to have had him as our guide.

We are also grateful to Dr. Hasan Mahmud, Professor, Systems and Software Lab, Department of Computer Science and Engineering, Islamic University of Technology for his guidance.

Abstract

Air-writing with body-worn inertial sensors is a promising, camera-free pathway to text input on constrained or screenless devices. Yet Bengalione of the worlds most widely used scriptsremains sparsely represented in public corpora, limiting reproducibility and comparative progress. We introduce a motion-sensor Bengali air-writing dataset comprising *3,996 single-character trials* spanning *60 core symbol classes* (vowels, consonants, digits), captured with a wrist/hand-mounted 6-DoF IMU at *200 Hz* under a controlled $2 \times 3 \times 2$ protocol varying handedness (dominant/non-dominant), writing speed (slow/normal/fast), and environment (normal/noisy). A cue-based ready–go–write–hold procedure, deterministic file naming, and self-describing CSVs with user/session metadata support transparent filtering, canonical splits, and faithful re-analysis.

We benchmark recognition using *classical machine-learning pipelines* built on compact time/frequency and cross-axis features, alongside *sequence models* that consume minimally processed IMU streams. The baselines verify the feasibility of IMU-based Bengali character recognition while surfacing persistent *fine-grained confusions among kinematically similar graphemes*, *cross-condition generalization gaps* (hand/speed/environment), and the difficulty of *writer-independent* performance without personalization. We further outline evaluation tracks for *streaming, online decoding* (joint boundary detection and classification) to bridge from trial-bounded clips to real-time interfaces.

As of our study, this is the *second publicly described* IMU-captured Bengali air-writing corpus with a documented acquisition protocol and reproducible baselines. By releasing data, canonical splits, and challenge definitions, we provide a rigorous foundation for future work on rotation-/tempo-robust representations, few-shot personalization, and low-latency decodingadvancing inclusive, privacy-preserving text input for non-Latin scripts.

Chapter 1

Introduction

As technology is evolving day by day, the way humans interact with the digital world is shifting beyond traditional physical interfaces like keyboards and touchscreens. With the fast development of wearable technologies, future digital environments are expected to offer immersive and seamless user experiences. The systems are expected to replace traditional screens with wearable devices to interact with digital devices. However, while output modalities are advancing significantly, input mechanisms remain a major limitation in these next-generation platforms.
Existing input technologies like gesture recognition have been researched extensively. Though there are shortcomings to this. Gesture recognition is restricted by a limited set of controlled environments, which may not be enough to cover complex interaction needs. As a result, new input technologies that are flexible, accurate, and socially acceptable are needed. It is the demand of the time.

1.1 Motivations and Scope

Air-writing characters in the air using free-hand gestures has come up with a promising input modality for future human-computer interaction. It offers a surface-free means for text entry, allowing users to interact with digital systems without the need for physical contact. Air-writing is useful for small-screen ot screenless devices such as AR and VR, where the use of traditional text input is impractical.
Despite the growing interest in air-writing, most existing studies have focused on English and other Latin-based scripts. But the complex scripts like Bengali receive less attention. Bengali is the fifth most spoken native language in the world. Yet, it has remained significantly underrepresented in digital technologies. For a more inclusive digital future, it is a must to develop intelligent input systems for underrepresented

languages like Bengali. This study aims to contribute to that goal by developing a motion sensorbased Bengali air-writing dataset to facilitate the creation of accurate character recognition systems.

1.2 Problem Statement

Despite promising prototypes, Bengali air-writing remains a data-scarce domain in the motion-sensing setting: there is no publicly available, standardized IMU-captured corpus of Bengali characters with documented acquisition protocols, calibrated sensors, synchronized labels, and writer/session diversity. Consequently, model development is fragmented and not reproducible, and reported accuracies are unlikely to generalize across users, devices, or environments.

Existing studies typically rely on small, proprietary, or vision-based datasets collected under controlled conditions; as a result, learned models are brittle to variation in writing style, speed, stroke order, handedness, device orientation, and ambient contexta direct symptom of insufficient data diversity and sample size.

There is therefore a clear need for a robust, open benchmark dataset that: (i) Covers the full Bengali inventory (39 consonants, 11 vowels, 10 digits) with adequate repetitions; (ii) Captures natural inter- and intra-writer variability across multiple sessions and environments; and (iii) Provides standardized splits, metadata, and baseline models to enable fair comparison and accelerate progress toward reliable, real-world Bengali air-writing systems.

1.3 Research Challenges

Developing a comprehensive Bengali air-writing dataset poses several challenges. Because, the Bengali script includes complex curves and diverse stroke patterns, making character separation and recognition more difficult than in Latin-based scripts. Also, Capturing consistent and high-quality motion sensor data for dynamic hand movements in three-dimensional space requires precise hardware calibration and a controlled data collection protocol. Furthermore, Variations in writing speed, hand movement and individual writing styles introduce noise into the dataset, making it challenging to achieve high recognition accuracy. And identifying discriminative features from time-domain, frequency-domain and statistical representations is critical for training efficient machine learning models.

1.4 Contribution

We develop a 3D motionsensor air-writing dataset covering the full Bengali inventory - 39 consonants, 11 vowels, and 10 digits with 6 repetitions per character, collected under a standardized protocol with per-trial metadata.

We design a lightweight, battery-powered data-collection device paired with a USB receiver, engineered for low-latency streaming and stable, session-to-session calibration. A cross-platform acquisition suite provides deterministic, time-stamped streaming, live 3D stroke visualization, one-click calibration, and structured data management (customizable file naming, session/subject metadata), forming a reproducible backbone for current and future studies.

We implement and evaluate a comprehensive feature set spanning time, frequency, and statistics-based descriptors to identify compact, high-signal representations for Bengali character classification. We train and report a benchmark machine-learning model on the dataset, establishing reference performance and protocols that support future comparative research and applied deployments.

1.5 Organization

Chapter 2 reviews related work in air-writing and character recognition systems. Chapter 3 describes the hardware setup, data acquisition methodology and dataset specifications.Chapter 4 shows the results from the methodology. Finally, Chapter 5 wraps up the study with a conclusion.

Chapter 2

Related Works

In recent years, air-writingthe act of writing characters or words in midair and interpreting them via sensorshas evolved into a rich interdisciplinary field. Early efforts focused on adapting pen-based and gesture-recognition techniques to a contactless, three-dimensional setting. Since then, research has branched into four primary technological approachescomputer vision, radar sensing, WiFi-signal analysis and inertial motion sensingeach offering distinct advantages and challenges. In the following sections, we trace these thematic threads, critically compare methodologies and highlight remaining gaps that motivate our own contributions.

2.1 Computer Vision-Based Air-Writing Recognition

Computer vision was the first modality to tackle air-writing, leveraging the ubiquity of cameras and mature image-processing toolkits. Under consistent lighting and simple backgrounds, vision-based systems achieve high accuracy by tracking a colored pointer or fingertip and extracting its trajectory.

Building upon early marker-based tracking, Shahinur et al. [1] introduced a trajectory-based system in which users wave a colored fingertip before a static camera, converting the 2D motion path into character stroke sequences. Their approach achieved reliable recognition for English letters, yet it depends heavily on stable illumination and a plain backdrop. De et al. [5] extended real-time digit classification to live video streams, employing background subtraction and a shallow classifier to achieve strong accuracy under controlled conditions. Qu et al. [12] in contrast, utilized cosine-similarity measures and fast nearest-neighbor searches to recognize both letters and digits, demonstrating improved robustness to slight variations in writing speedbut still faltering when shadows or cluttered scenery intrude.

More recently, deep learning has been applied to air-writing [8]: researchers have trained convolutional neural networks on large corpora of characters, reporting near-human performance under fixed lighting. Likewise, another team proposed [19] the Temporal Convolutional Recurrent Network (TCRN)-a hybrid architecture tailored for online handwritten textand outperformed prior methods on benchmark datasets. However, despite these advances, all vision-based systems share a critical vulnerability: their reliance on controlled environments. Variations in ambient light, occlusions and dynamic backgrounds significantly degrade performance, limiting real-world applicability.

2.2 Radar Sensor-Based Air-Writing Recognition

To overcome lighting and occlusion issues, researchers have turned to millimeter-wave and ultra-wideband (UWB) radars, which sense object motion via reflected electromagnetic signals. Because radar is insensitive to visible-light conditions and penetrates certain materials, it offers a compelling alternative for in-air gesture capture.

In pioneering work, teams designed millimeter-wave radar applications for recognizing basic gestures, digits and alphabetic symbols in 2D space [1]. They combined range-Doppler signal processing with template matching to achieve strong recognition rates. Expanding this concept, other researchers proposed a 60 GHz FMCW radar network, where spatially arranged sensors trilaterate the users hand position, then feed the trajectory into a temporal convolutional networkyielding high accuracy on sizeable datasets.

Alternatively, another group exploited IR-UWB radar sensors in a rectangular layout to capture 3D character strokes without any handheld marker. Their system surpassed prior radar-based methods in classification accuracy but at the expense of portability: the requirement for multiple fixed sensors constrains mobility and increases system complexity. While radar solutions excel under varied lighting and background conditions, their hardware cost, spatial coverage requirements and power consumption pose obstacles for consumer-grade deployment.

2.3 WiFi Signal-Based Air-Writing Recognition

A nascent but promising direction leverages commodity WiFi devices to sense human motion through Channel State Information (CSI). By analyzing the subtle perturbations in WiFi signal reflections caused by hand gestures, these systems aim to infer writing trajectories without dedicated cameras or radars.

Though only a handful of studies have explored this modality, preliminary results indicate that fine-grained CSI features can distinguish between a limited set of characters with reasonable accuracy. However, WiFi-based air-writing remains in an exploratory phase: challenges include isolating hand-induced signal variations from other environmental changes and achieving sufficient spatial resolution to capture rapid stroke transitions. As CSI-based recognition matures, it may offer a highly pervasive and low-cost solution, but substantial algorithmic innovation is required to match the fidelity of vision or radar systems.

2.4 Motion Sensor-Based Air-Writing Recognition

Complementing external-sensor approaches, motion-sensor-based methods embed inertial unitsaccelerometers and gyroscopeseither in wearable armbands or handheld devices [21]. These systems translate 3D motion signals directly into stroke trajectories, free from lighting or background constraints.

The 2-DifViz algorithm converts raw Myo armband data into 2D pen-plotter commands, enabling real-time text entry from arm motions. Other researchers have employed contour-based 3D models, where hand gestures map onto parametric curves recognized as letters. Hidden Markov and continuous HMM frameworks appear frequently [15][10][2]: some work combines Leap Motion data with bidirectional LSTM networks, achieving strong word-level accuracy, while other systems use inertial readings and continuous HMMs to classify stroke sequences with high precision [7][9]. Additionally, digital pens with built-in accelerometers have realized over 90 percent accuracy in controlled trials.[20][16]

Motion-sensor techniques offer robustness to external factors and are easily portable-but often demand that users hold or wear a device, potentially impacting user comfort and limiting long-form writing. Moreover, drift and noise in inertial sensors necessitate sophisticated filtering and trajectory-reconstruction algorithms, which can introduce latency or reduce stroke fidelity[17].

2.5 Synthesis and Identified Gaps

Across these four modalities, several trends emerge. Vision-based systems lead in maturity and leverage deep learning for high-accuracy recognition, yet they falter in uncontrolled environments. Radar approaches overcome lighting issues but at the cost of bulky hardware. WiFi-based methods promise pervasive sensing but currently lack the spatial resolution for precise character capture. Motion sensors deliver portability and lighting invariance but require user-worn devices and contend with sensor drift.

A further thematic gap is linguistic diversity. Nearly all studies concentrate on English letters and digits; only a few address scripts like Bengali and Devanagari numeralsreporting high accuracy under constant illuminationyet they omit the full Bengali alphabet [13][11]. No existing system robustly recognizes non-Latin scripts in dynamic, real-world settings [4]. Additionally, few works integrate multi-modal sensing (e.g., combining vision with inertial data) to exploit complementary strengths and mitigate individual weaknesses.

2.6 Summary of the related works

In sum, air-writing recognition has progressed along distinct technological pathways, each with unique trade-offs in accuracy, robustness and portability. While computer vision and deep learning have driven recent advances, their environmental sensitivity underscores the need for more resilient designs. Radar and WiFi methods introduce powerful sensing alternatives yet face hardware and resolution constraints. Motion sensors deliver portability but wrestle with noise and user ergonomics. Critically, the narrow focus on English limits applicability in multilingual contexts.

By identifying these thematic strengths and weaknesses, as well as linguistic and multi-modal integration gaps, we establish the foundation for our proposed methodology: a hybrid system that combines vision and inertial sensing, enhanced by domain-adapted models for Bengali script, designed to operate reliably in real-world environments. This approach aims to bridge existing gaps and extend air-writing technology to new user populations and practical applications.

Chapter 3

Proposed Methodology

3.1 Overview of the Methodology

The proposed system aims to recognize air-written Bengali characters using motion data from sensors (accelerometer and gyroscope). The process begins with data acquisition via wearable devices or handheld sensors. Each gesture is captured, stored as raw motion data, and then pre-processed to reduce noise and standardize values. After that, a wide range of features (time-domain, frequency-domain) is extracted from the cleaned signal. Feature selection techniques are applied to retain the most informative attributes. Finally, a classification model is trained using the selected features to identify the corresponding Bengali character from the gesture.

3.2 Chronological Breakdown of Components

3.2.1 Data Collection Device Construction

We engineered a low-intrusion wearable to capture Bengali air-writing while preserving natural kinematics and user comfortcore goals in HumanComputer Interaction (HCI). The sensing element is a six-degree-of-freedom IMU (tri-axial accelerometer and gyroscope) positioned near the index finger to maximize sensitivity to micro-strokes and fine angular changes; this fingertip-proximal placement follows HCI practice of instrumenting the locus of control to reduce mediation between intention and sensed motion. Processing, power, and physical I/O are offloaded to a compact, 3D-printed wrist module secured by an adjustable Velcro strap. Distributing mass to the forearm while keeping the sensor light at the finger lowers distal inertia, minimizes fatigue, and maintains a stable sensorskin couplingcrucial for ecological validity. Flex-

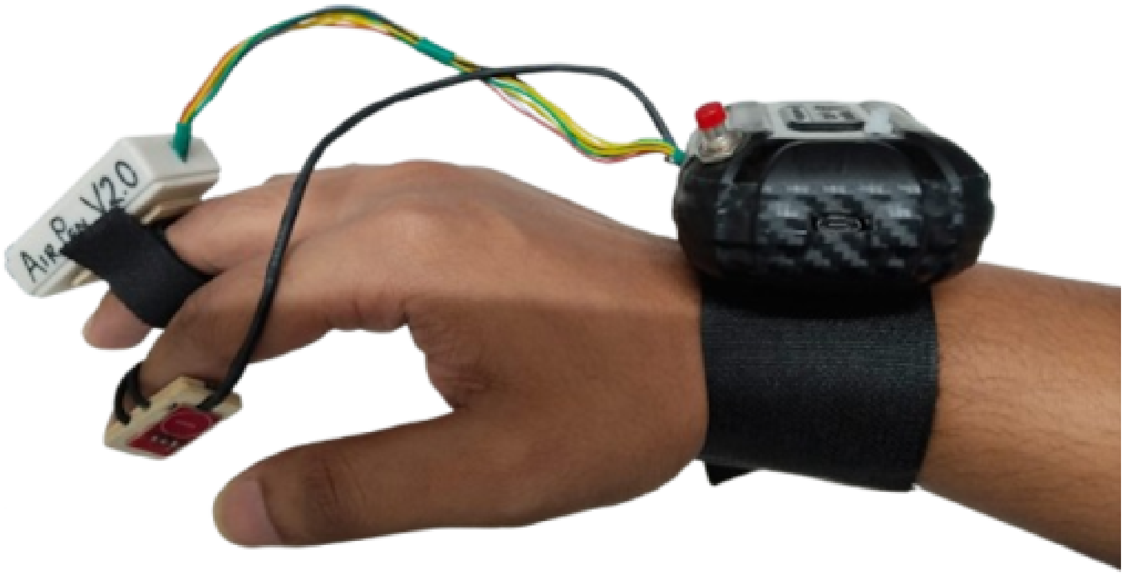


Figure 3.1: Data Collection Device

ible, strain-relieved wiring preserves wrist freedom and prevents cable torque from biasing trajectories. A single tactile button and status LED expose the system state (idle/armed/recording) and allow participants or the operator to drop explicit event markers that align with our *readygowritehold* protocol; this immediate, glanceable feedback reduces cognitive load and supports learnability and error recoverabilitykey HCI qualities in wearables. The microcontroller timestamps samples at 200 Hz and streams loss-bounded packets to the logger, enabling auditability and post-hoc synchronization with cues. Materials and enclosure geometry were iteratively refined for all-day comfort (rounded edges, skin-safe contact surfaces), ambidextrous use, and quick don/doff in lab settings. The electronics are modular to facilitate maintenance and replication across sites. Notably, we omit a magnetometer to avoid indoor ferromagnetic interference; instead, downstream preprocessing performs gravity alignment and drift control. Collectively, these design choicesergonomic form factor, transparent feedback, stable coupling, and modularityembody HCI principles of embodied interaction and low intrusiveness, ensuring that the captured signals reflect authentic mid-air handwriting rather than instrumentation artifacts.

3.2.2 Data Collection Tool Development

System Overview

The *Bengali Air Writing System* is a research-grade tool specifically designed to capture fine-grained motion sensor data during the in-air writing of Bengali characters. Unlike camera-based handwriting datasets, which often raise privacy concerns and demand high computational cost for image preprocessing, our system relies on a wearable inertial measurement unit (IMU) that records six degrees of freedom (6-DoF) motion data in real time. This approach allows for highly portable, privacy-preserving, and low-latency data collection, which is particularly valuable for under-represented non-Latin scripts such as Bengali.

The system comprises two tightly integrated components: (i) Hardware Layer: A wireless or USB-connected IMU sensor that continuously streams accelerometer and gyroscope data. (ii) Software Layer: A cross-platform desktop application responsible for session management, real-time signal visualization, user guidance, metadata handling, and persistent storage of data. Together, these components provide a controlled acquisition environment that guarantees reproducibility, supports large-scale dataset development, and enables downstream research in handwriting analysis, gesture recognition, and humancomputer interaction (HCI).

Technical Architecture

The architecture of the Bengali Air Writing System has been designed to meet three primary requirements: (i) Robustness: ability to handle noisy streams, malformed packets, and unexpected user interactions. (ii) Real-Time Performance: ensuring live feedback to the operator, both visually and aurally, with low latency. (iii) Cross-Platform Compatibility: seamless deployment on Windows, macOS, and Linux environments without requiring specialized drivers.

Software Framework

The application is implemented in Python 3.12, chosen for its extensive library ecosystem, rapid prototyping capabilities, and strong community support in data science. The software adopts an event-driven, multi-threaded model to separate computationally intensive tasks (e.g., serial I/O, data logging, visualization) from the main GUI thread. Key libraries and their roles are as follows:

Table 3.1: Used Python Libraries and Their Functions

Library	**Functionality**
PyQt5	Provides a cross-platform graphical user interface (GUI) with responsive layouts, theming, and operator interaction features.
PyQtGraph	Supports real-time plotting of accelerometer and gyroscope signals, with millisecond-scale refresh rates.
PySerial	Ensures stable serial communication with IMU devices at fixed baud rates.
NumPy	Enables fast, in-memory signal processing and buffer management.
pyttsx3 (TTS)	Delivers auditory instructions (e.g., announcing the current character or writing speed) to standardize experiment protocols.
PyGame	Provides supplementary auditory cues (such as beeps or warnings) for accessibility and immediate feedback.

This modular design ensures separation of concerns: GUI logic is independent from data acquisition, enabling easier maintenance and extensibility.

Hardware Integration

The system interfaces with a 6-DoF IMU comprising a 3-axis accelerometer and a 3-axis gyroscope. The IMU streams sensor readings over a serial communication protocol at 19,200 baud, ensuring a balance between throughput and packet reliability.

Each data packet follows a standardized seven-value format:

$$\text{Packet} = [\text{Timestamp}, A_x, A_y, A_z, G_x, G_y, G_z]$$

Where:

- A_x, A_y, A_z Linear accelerations along the x, y, and z axes.
- G_x, G_y, G_z Angular velocities around the x, y, and z axes.
- Timestamp A synchronized time marker, aligned with the host system clock.

Incoming packets undergo strict validation (length, delimiter checks, numeric ranges). Valid data is then time-aligned, buffered, visualized, and finally stored to disk for later analysis.

Core Functionalities

A. Acquisition Parameters

A distinguishing feature of this system is the rich metadata captured alongside raw sensor readings, ensuring end-to-end traceability and experiment reproducibility. Each recording session logs the following parameters:

Table 3.2: Metadata Categories Captured During Data Collection

Category	Description
Character Class	A set of 60 Bengali characters, including the full alphabet (consonants and vowels) and digits.
Handedness	Whether the participant used the dominant or non-dominant hand.
Data-Type Flag	Indicates whether the recording represents natural writing or a Parkinsons disease simulation.
Writing Speed	Labeled as slow, normal, or fast, to capture kinematic variations.
Participant Demographics	Includes age, gender, and optional study-specific annotations (e.g., clinical conditions).

This structured parameterization allows for systematic research on handwriting variability and pathological handwriting features.

B. Real-Time Processing

The system maintains a rolling buffer of 50 samples (approximately 1 second of data), ensuring smooth visualization and operator oversight. The GUI refreshes every $\sim$30 ms, rendering dual synchronized plots of accelerometer and gyroscope signals.

Additional real-time features include:

- **Signal Quality Indicators:** Alerts for packet drops, latency spikes, or missing fields.
- **Auditory Guidance:** Automatic TTS announcements of the current task (e.g., Character: , Hand: Right, Speed: Normal), reducing operator workload and preventing mislabeling.
- **Automated Error Prevention:** Guardrails such as collision-proof filename generation prevent accidental overwrites and missing coverage.

Data Management System

A. Directory Schema

To simplify data retrieval and ensure reproducibility, recordings are organized in a deterministic hierarchical structure:

```
recordings/
 data_training/
    [Character]/
        [Character]_[DataType]_[Speed]_[UserID]_[Hand]_[Timestamp].csv
 users.csv
```

This structure guarantees uniqueness of each file and enables efficient filtering based on metadata.

B. CSV Record Layout

Each row in the generated CSV file contains:

Table 3.3: Key Data Elements Captured in Each Recording

Data Element	Description
User and Session Metadata	Demographics, character class, writing speed, writing environment.
Relative Timestamp	Time elapsed since session start.
Sensor Values	Accelerometer and gyroscope readings: $A_x, A_y, A_z, G_x, G_y, G_z$.

This design makes the dataset self-describing and compatible with diverse analysis tools such as Python (NumPy/Pandas).

Participant Management

Each participant is assigned a UUID for anonymity and traceability. The Participant Manager provides features such as: (i) Search and filtering of participant records. (ii) Import/export of user profiles across studies. (iii) Progress tracking across all characters, speeds, and hands. (iv) By maintaining a completion matrix, the system helps identify missing data and ensures balanced dataset coverage. (v) Participant's list can be sorted by various parameters (such as ID, Name, Age, Gender, Completion Percentage)

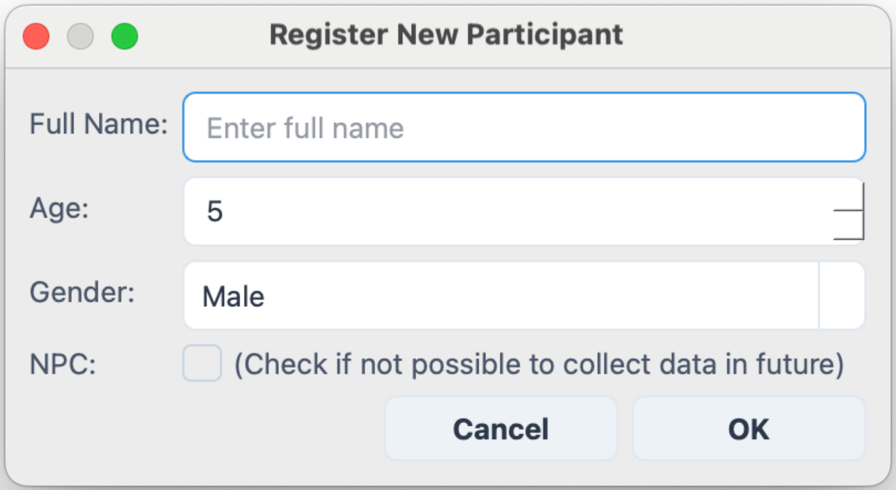


Figure 3.2: Participant Register

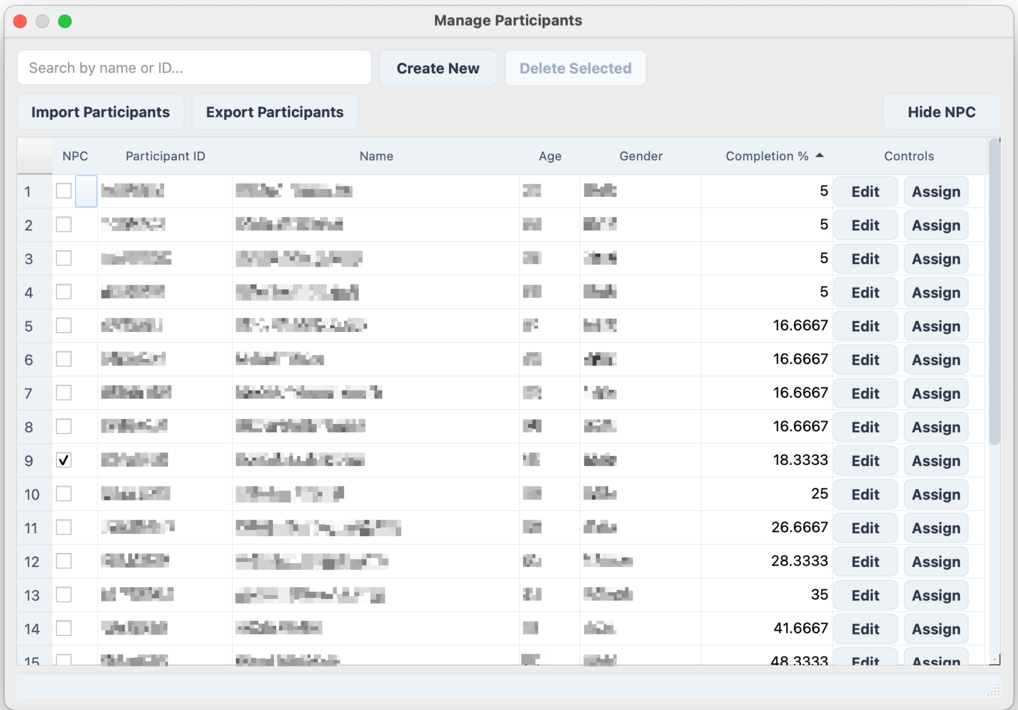


Figure 3.3: Participant Management

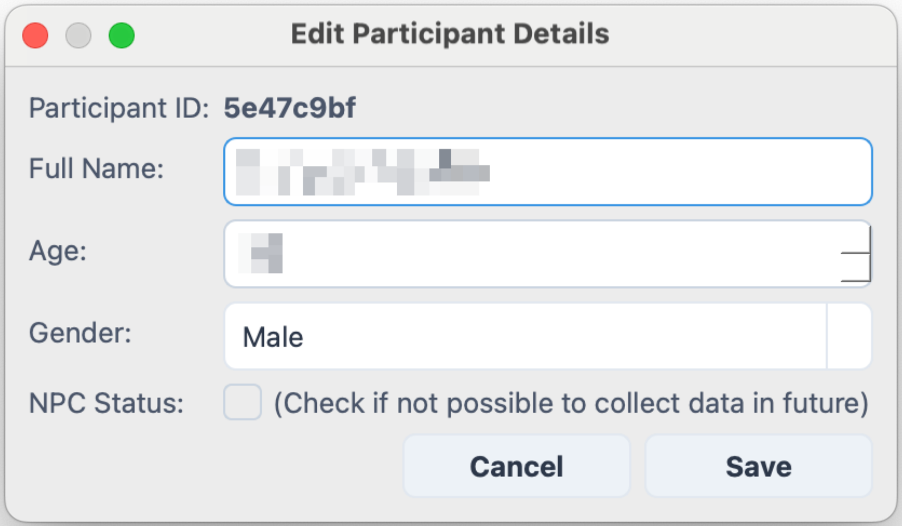


Figure 3.4: Edit Participant

User Interface and Accessibility

The graphical user interface (GUI) has been thoughtfully designed to minimize cognitive load on operators while enhancing accessibility for a broad range of users. Key design considerations include adaptability, clarity, and user customization, ensuring efficient and error-resistant operation during data collection sessions.

Table 3.4: Key Features of the User Interface for Accessibility and Usability

Feature	Description
Responsive Layout	Adapts seamlessly to various screen sizes and resolutions; supports both light and dark themes to reduce visual fatigue and accommodate user preferences.
Visual States	Distinct color-coded indicators communicate connection status, recording activity, and idle conditions, enabling operators to quickly interpret system state at a glance.
Customization	Supports keyboard shortcuts and saves user settings persistently, streamlining repetitive workflows and reducing manual configuration overhead.

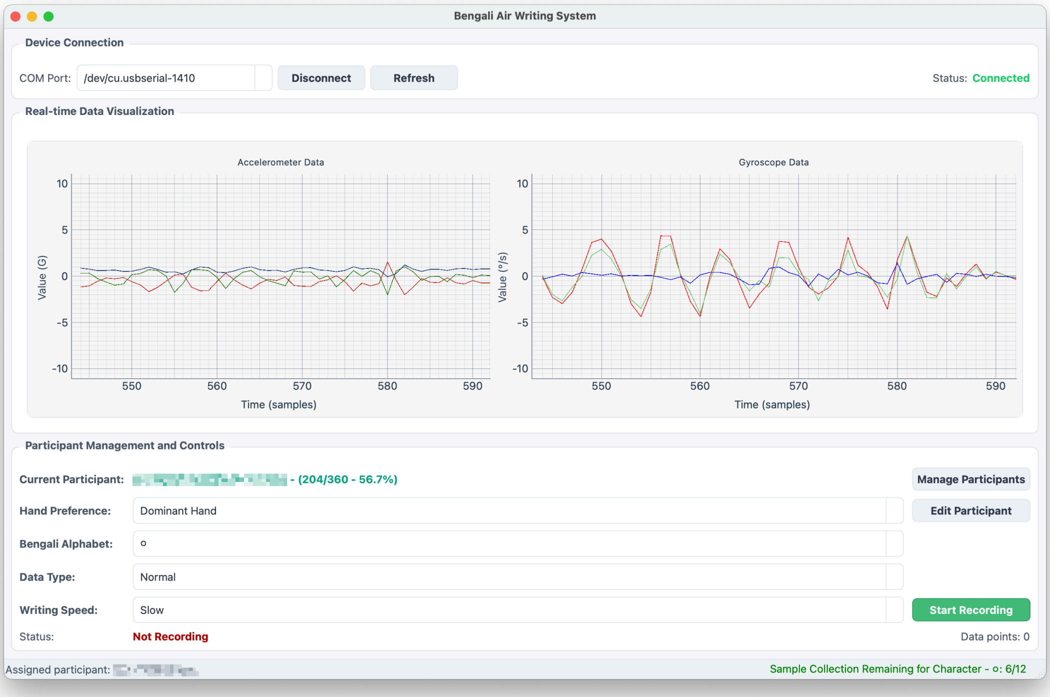


Figure 3.5: Data Collection Tool

Specialized Panels

The Sensor Viewer is a crucial component that displays dual real-time plots for accelerometer and gyroscope signals. This simultaneous visualization allows operators to closely monitor the motion data streaming from the wearable sensors, with updates at the millisecond level. Such immediate feedback helps in verifying the quality and consistency of the recorded data during each session.

The Participant Manager provides an interface for inline editing of participant demographics. It also tracks the progress of data collection in real time, giving researchers an up-to-date overview of participant status. This integration reduces administrative overhead and helps ensure that metadata is accurately maintained throughout the study.

The Session Controller simplifies the process of recording data by offering a one-click start and stop mechanism. It incorporates robust, collision-proof file handling to prevent accidental overwriting or loss of data files. This reliability is essential for maintaining data integrity across large-scale recording sessions and supporting reproducible research.

Technical Specifications

The systems performance is enhanced by utilizing a 50-sample buffer that effectively eliminates any lag during real-time visualization of sensor data. This buffering ensures smooth and responsive feedback for operators. Moreover, data is securely saved to CSV files using atomic file creation methods combined with exception-aware retries, which safeguard data integrity by preventing corruption or loss even in cases of unexpected interruptions.

In terms of compatibility, the software has undergone extensive testing across multiple operating systems, including Windows, macOS, and Linux. This cross-platform support ensures broad usability in diverse environments. Furthermore, the system is designed to work seamlessly with any inertial measurement unit (IMU) device that complies with the established packet format, making it flexible and adaptable to various hardware configurations.

To maintain rigorous quality assurance, several validation and error-handling mechanisms are incorporated. These include duplicate-detection routines, strict checks on metadata validity, and timestamp guards to ensure temporal consistency. Additionally, the software provides clear, descriptive error prompts to notify users of common issues such as busy serial ports, malformed data packets, or disk write failures, thereby improving reliability and aiding in troubleshooting during data collection.

3.2.3 Data Collection Protocol

This subsection specifies acquiring Bengali air-writing motion data from inertial sensors. It covers participant briefing and eligibility, apparatus and environment, task design and condition factors, calibration, trial flow, counterbalancing, quality control, post-session procedures, timing, risk mitigation, and data governance. The protocol is designed to generalize across labs while remaining tightly coupled to the acquisition software and hardware described in.

Participant Briefing and Consent

Participants attend a structured briefing that includes:

- Study objectives (motion-sensor capture of Bengali air-writing for recognition research)
- Procedures and expected duration
- Data types recorded (IMU streams, timestamps, metadata)

- Risks/benefits (minimal risk; scientific contribution)
- Confidentiality and de-identification
- Voluntary nature of participation

The session concludes with an opportunity for questions and written informed consent. Participants are reminded they may pause or withdraw at any point without penalty.

Eligibility (Inclusion/Exclusion)

Table 3.5: Participant Inclusion and Exclusion Criteria

Criteria	Description
Inclusion	Native Bengali speakers or individuals with functional command of the Bengali script, including reading and writing abilities.
Exclusion	Individuals with acute upper-limb injuries or conditions that would make arm or hand movement unsafe or painful on the day of recording. Participants with stable neuromotor conditions may be enrolled under a separate protocol where applicable.

Demographic and Baseline Enrollment

At enrollment, the data are recorded: age, gender, and dominant hand (self-reported). The acquisition suite assigns a UUID and stores a baseline profile in `users.csv`.

Apparatus and Environment

Sensors: Wrist/hand-mounted IMU (3-axis accelerometer + 3-axis gyroscope), streamed via serial link. Each packet includes:

$$\text{Packet} = [\text{Timestamp}, A_x, A_y, A_z, G_x, G_y, G_z]$$

Operator console: Desktop/laptop running the cross-platform GUI (mentioned in table:3.1), with real-time plots.
Posture: Participants stand comfortably with the writing hand free; the non-writing hand may rest at the side. A clear writing volume (∼40 cm cube) is indicated at chest height to standardize strokes.

Task Design (Characters and Hands)

Participants air-write Bengali alphanumerics using both dominant and non-dominant hands. The character set includes 60 classes (digits, vowels and consonants). The software announces the target character via on-screen label and TTS. To reduce cognitive load, characters are grouped, with short rests between groups.

Condition Factors (Speed x Environment x Hand)

Each character is recorded under 12 variations in a $2 \times 3 \times 2$ factorial design:

Table 3.6: Experimental Variables

Variable	Conditions
Hands	Dominant, Non-dominant
Speeds	Slow, Normal, Fast
Environments	Normal, Parkinson (Noisy)

The acquisition UI encodes these factors in the filename and metadata to ensure full traceability.

Calibration and Warm-Up

To control sensor bias and alignment without magnetometers:

- **Static Bias Capture (3 seconds):** Participants are instructed to hold the device completely motionless for three seconds. During this period, the software estimates the gyroscope bias to calibrate sensor drift and verifies the accelerometers gravity magnitude, which should be approximately 9.81 $\mathrm{m/s^2}$ within an acceptable tolerance range. This step ensures accurate baseline sensor readings.
- **Axis Alignment Gesture (2 seconds):** Participants perform a slow, deliberate L-shaped movement within the sagittal and transverse planes. This gesture is used to validate the sensors sign conventions and confirm correct axis orientation, ensuring consistent interpretation of motion data across sessions.
- **Warm-up Trials (per hand):** Participants complete 3 to 5 practice characters for each hand, which are not logged in the dataset. These warm-up trials help familiarize participants with the designated writing volume and target speeds, reducing variability due to unfamiliarity during actual data collection.

Trial Flow and Timing

Each trial follows a consistent cue-based structure:

- **Ready Cue (TTS + Tone, 1 second):** This phase uses a combination of text-to-speech (TTS) announcements and an auditory tone lasting one second to prepare the participant for the upcoming trial. During this cue, the system clearly communicates essential parameters including the character class to be written, the hand to be used (dominant or non-dominant), the intended writing speed (slow, normal, or fast), and the environmental condition (normal or noisy). This ensures that the participant is fully informed and ready before the recording begins.
- **Go Cue (Tone, 0.5 seconds):** Following the ready cue, a brief half-second tone signals the participant to begin the air-writing task. This auditory prompt marks the start of the active writing phase, during which the participant is expected to initiate motion within the predefined 3D writing volume. The tone provides a clear, unambiguous indication to promote consistent timing across trials.
- **Write Window (10–35 seconds):** During this extended interval, the participant performs the in-air writing of the specified Bengali character. Multiple strokes are allowed within this window to accommodate natural writing behaviors, including corrections or complex character components. The duration range allows flexibility while ensuring sufficient time to complete the character without excessive delay.
- **Hold Still (0.8–1.2 seconds):** At the conclusion of the writing phase, participants are instructed to hold the hand steady for a brief period between 0.8 and 1.2 seconds. This quiescence interval serves to capture post-motion stabilization data and helps delineate the end of each trial clearly in the sensor recordings. It also reduces motion artifacts that might interfere with downstream analysis.

Per character, at least six repetitions are targeted across the 12 condition cells (balanced as scheduling permits). The software enforces a minimum inter-trial interval of 1.52.0 s to avoid carry-over motion.

Table 3.7: Summary of Trial Phases and Total Duration per Variation

Phase	Duration	Description	Duration (s)
Ready Cue	Fixed	TTS and tone prepare participant by announcing character class, hand, speed, and environment.	1
Go Cue	Fixed	Tone signals participant to start air-writing within the defined 3D volume.	0.5
Write Window	Variable	Participant performs in-air writing; multiple strokes allowed for complex characters.	10–35
Hold Still	Variable	Participant holds steady to capture post-motion stabilization and reduce artifacts.	0.8–1.2
Total Time per Trial Variation	—	Sum of all phases	**12.3 – 37.7**

Table 3.8: Approximate Time Requirements for Data Collection for Each Character from Each Participant

Description	Minimum Time (s)	Maximum Time (s)
Time per repetition (trial + inter-trial interval)	13.8	39.7
Total time for all 12 conditions	165.6	476.4

Randomization and Counterbalancing

To mitigate potential confounding effects such as order bias, participant fatigue, and learning throughout the data collection process, the Bengali Air Writing System employs a carefully designed experimental protocol. First, a Latin-square design is used to counterbalance the sequences of writing speed and environmental conditions for each participant. This approach ensures that no single speed or environment condition consistently precedes or follows another, reducing systematic order effects across the dataset.

Second, the experiment alternates hand usage in structured blocks, cycling between the dominant and non-dominant hands (e.g., Dominant Non-dominant Dominant, and so forth). Between these hand blocks, short rest periods are incorporated to help alleviate muscular fatigue and maintain participant performance consistency. This alternating scheme also allows clear separation of data by hand, supporting hand-specific analysis without confounding cross-hand carryover effects.

Finally, within each hand block, the order of characters presented to the participant is randomized without replacement. This ensures that participants do not repeatedly write the same characters in a predictable sequence, minimizing learning effects related to memorization or motor repetition. Importantly, the exact order of trialsincluding speed, environment, hand, and characteris logged in the session metadata. This detailed record facilitates sophisticated mixed-effects modeling in downstream analyses, allowing researchers to account for any residual variability due to trial order or participant-specific factors.

Quality Control and Re-Collection Criteria

The acquisition suite integrates rigorous online quality control procedures to ensure the validity and consistency of the collected data. During each trial, the system continuously monitors incoming sensor data for anomalies that could compromise the quality or interpretability of the recording. One key metric is packet integrity; if the proportion of malformed or dropped data packets exceeds 0.5% of the trial duration, the trial is considered unreliable due to potential data loss or corruption.

In addition, the system evaluates the motion envelope to detect issues related to participant performance or sensor limitations. Specifically, if the recorded peak acceleration is abnormally low, it may indicate under-articulation, where the participants movements are insufficiently expressive or incomplete. Conversely, if sensor signals saturate due to clipping, it suggests the motion exceeded the sensors measurable range, potentially distorting the data. Both scenarios are flagged as unacceptable.

Lastly, the system verifies protocol compliance by checking that the participant uses the correct hand specified for each trial. Any hand mismatch is automatically detected and marked, as this would invalidate the condition for that trial.

When any of these quality checks fail, the acquisition software promptly notifies the operator with an on-screen warning. The affected trial is immediately re-queued for re-collection to maintain data integrity. Furthermore, all quality flags, alongside any operator interventions or overrides, are meticulously logged in the session metadata. This ensures transparency and supports downstream data auditing and analysis.

Annotation, File Naming, and Metadata

Each trial is saved as:

`[Character]_[Env]_[Speed]_[UserID]_[Hand]_[Timestamp].csv`

Files contain time-aligned rows: relative timestamps, $A_x, A_y, A_z, G_x, G_y, G_z$. A session manifest (CSV) summarizes trial counts by condition and flags.

Post-Session Questionnaire and Debrief

Following the completion of the recording session, participants are asked to fill out a brief questionnaire designed to capture subjective feedback about their experience. The questionnaire combines Likert-scale items with open-ended free-text responses to provide both quantitative and qualitative insights. Participants report on several aspects including the perceived difficulty of the air-writing tasks, their comfort level and any fatigue experienced during the session, and the clarity and effectiveness of the auditory and visual cues provided throughout the experiment. Additionally, they are encouraged to note any anomalies or disruptions encountered, such as sensor slippage or environmental distractions, which might have impacted the data quality or their performance. This post-session feedback is valuable for assessing participant burden, refining experimental protocols, and interpreting the collected data within the context of user experience.

Open-ended feedback is encouraged, especially regarding non-dominant hand writing and noisy conditions. A short verbal debrief concludes the session.

Session Timing and Breaks

A typical session lasts 25–30 minutes including briefing, calibration, practice, trials, and questionnaire. Participants may take breaks as needed; the operator offers rest between blocks and resumes only with verbal confirmation. Multi-session participation is allowed with the unique UUID.

Risks, Comfort, and Safety

Minimal risk. Mild forearm fatigue is mitigated by short blocks, optional seated posture, and breaks. No personal images or audio are recorded. Participants may withdraw at any time without reason.

Data Privacy, Governance, and Release Readiness

The data management protocol prioritizes participant privacy and data integrity throughout the acquisition and distribution process. To ensure de-identification, only universally unique identifiers (UUIDs) are used in filenames and metadata, with any direct personal identifiers securely stored separately and excluded from distributed datasets. Each recording session logs the versions of the acquisition software and IMU firmware employed, enabling precise tracking of any potential software or hardware changes that might affect data consistency.

File integrity is rigorously maintained through atomic saving procedures accompanied by checksums, which help detect any corruption during storage or transfer. Additionally, the system implements duplicate-recording detection to prevent redundant data entries. Upon receiving ethics approval, a comprehensive public dataset release is planned. This release will include detailed documentation of all experimental protocols, precise definitions for each experimental factor, standardized train/validation/test splits, and baseline code to facilitate preprocessing and classification tasks. Together, these measures ensure transparency, reproducibility, and equitable benchmarking for future research leveraging this dataset.

3.2.4 Diagrams

Use Case Diagram

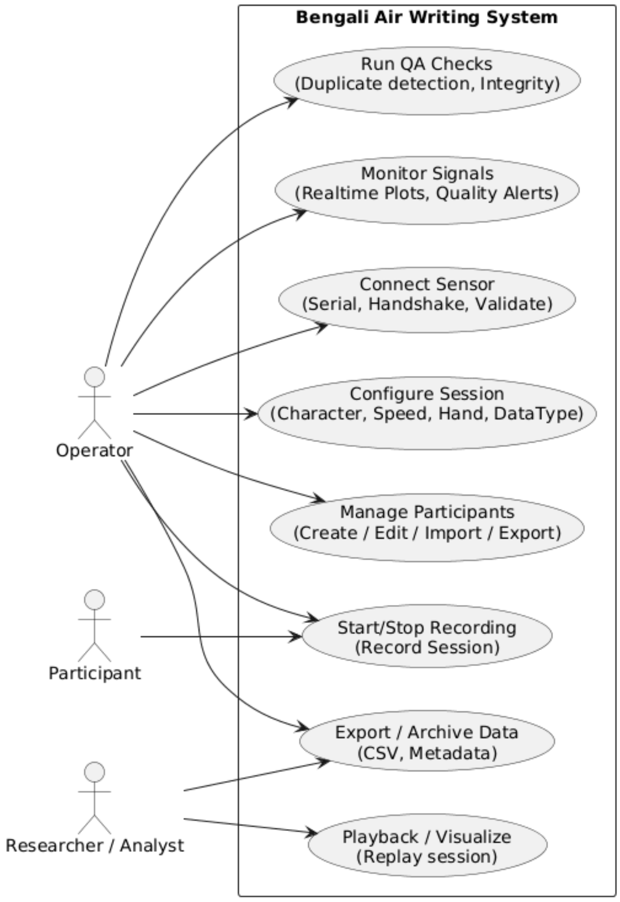


Figure 3.6: Use Case Diagram

The Use Case Diagram presented in Figure 3.6 models the functional requirements of the Bengali Air Writing System by illustrating the interactions between the system and its primary users, or "actors." It provides a clear visualization of the essential operations the system must perform to facilitate effective data collection and subsequent analysis for Bengali air-writing research.

Actors

The use case diagram identifies three primary actors involved in the Bengali Air Writing System. The foremost actor is the Operator, who is responsible for conducting the data collection sessions. Typically, this role is fulfilled by a researcher or lab assistant who oversees the hardware setup, configures the software parameters, manages participant information, and controls the entire recording process during the experiments. The Operator ensures that all necessary configurations are in place for accurate data acquisition and oversees the smooth execution of each trial.

The second actor is the Researcher or Analyst, who interacts with the system mainly after the data collection phase is complete. Unlike the Operator, whose focus is on live data capture, the Researchers role centers on post-processing and analysis of the recorded data. They may use functionalities such as playback and visualization to study the motion trajectories and assess data quality for further research or validation purposes.

Finally, the Participant is the source of the air-writing data but is considered a passive entity in the system model, existing outside the software boundary. Participants do not directly interact with the systems software interface; instead, their hand movements during the air-writing tasks are captured by the wearable inertial measurement unit (IMU) sensor managed by the Operator. This separation highlights the systems focus on capturing data through sensor hardware rather than direct user input to the software.

Core Use Cases (System Functionalities)

The use cases are organized to represent the typical workflow of the system, divided into three main phases:

In the initial phase, the Operator is responsible for preparing the system for data acquisition. This begins with establishing a reliable serial communication link between the system and the wearable Inertial Measurement Unit (IMU) sensor. The connection process involves performing handshake procedures and validating the incoming data streams to ensure that the hardware integration is successful and stable, as elaborated in the systems technical architecture. Following this, the Operator manages participant profiles by creating, editing, and maintaining demographic information. This participant management functionality is designed to uphold privacy through an anonymization protocol, utilizing universally unique identifiers (UUIDs) and a users.csv file system for secure data handling. Lastly, the Operator configures the parameters for each recording session. These configurations include selecting the target character for air writing, specifying the writing speed, choosing the hand to be used, and determining the data type, such as distinguishing between normal and Parkinsons simulation conditions. This structured setup ensures adherence to the experimental design and facilitates systematic data collection.

During the active data collection phase, the system provides real-time monitoring of sensor outputs, specifically accelerometer and gyroscope data. This immediate visual feedback allows the Operator to verify the quality and integrity of the incoming signals, ensuring that the data is suitable for research purposes. Quality assurance procedures are incorporated to detect issues such as duplicate recordings or data corruption. These QA checks may be automated or manually performed, serving as critical steps to maintain the reliability of the dataset in line with the quality control protocols described in the methodology. The core functionality in this phase is the initiation and termination of recording sessions. When recording is active, the system logs time-stamped sensor data and relevant metadata into CSV files. This process is tightly controlled to follow the precise flow of each trial as defined in the experimental protocol.

After data acquisition, the system supports comprehensive post-processing and data management functionalities. Recorded sessions can be securely packaged and archived, including all raw sensor data and associated metadata, to facilitate data sharing and future public release. Furthermore, the system offers playback and visualization tools that allow both Operators and Researchers to review recorded sessions. By visualizing the motion trajectories, users can perform detailed analyses, debug data quality issues, and conduct kinematic studies. This functionality bridges the gap between data collection and meaningful interpretation, supporting the research objectives of the Bengali air-writing project.

Relationships and System Boundary

The system boundary is represented by a rectangular box encapsulating all the system-provided functionalities (use cases). Actors outside this boundary initiate interactions via connecting lines that indicate the roles responsible for each use case. The Participant is intentionally placed outside the system boundary to signify indirect interaction through sensor data capture.

This Use Case Diagram effectively encapsulates the comprehensive software suite detailed in 3. It demonstrates the integration of hardware management, participant administration, real-time data monitoring, and robust post-processing tools necessary to support a rigorous and reproducible Bengali air-writing data collection protocol.

Activity Diagram

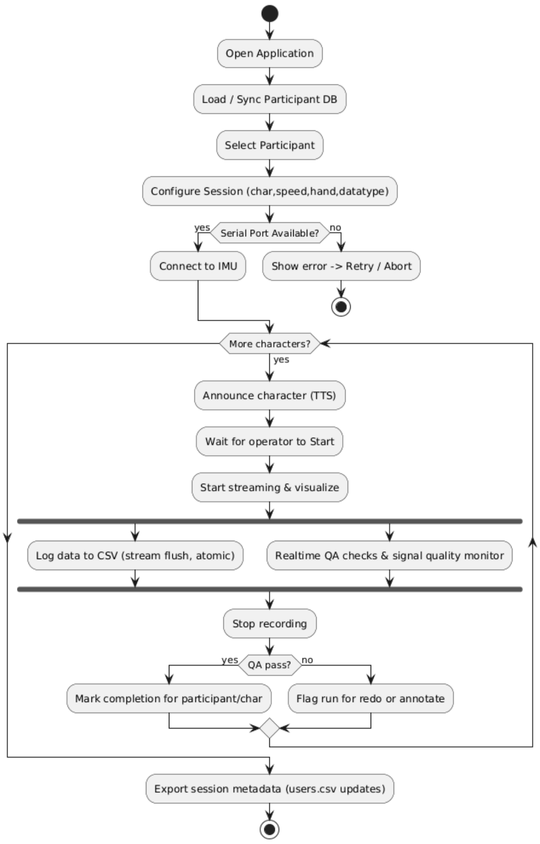


Figure 3.7: Activity Diagram

Figure 3.7 presents the Activity Diagram representing the control flow of a typical data collection session within the Bengali Air Writing System. It visualizes the sequential and parallel actions executed by both the Operator and the system, including key decision points and the repetitive nature of collecting multiple character samples per participant.

Initialization Phase

The session begins when the Operator launches the custom-built desktop application. Upon startup, the system automatically loads and synchronizes the participant database (`users.csv`), making all previously registered profiles and their progress accessible. The Operator selects a participant from this database, which assigns a unique UUID to the session and ensures that all subsequent data is correctly attributed.

The next step is session configuration, where the Operator defines the experimental parameters for the upcoming trial. This includes selecting the Bengali character to be written, choosing the writing speed (slow, normal, or fast), determining the hand to be used (dominant or non-dominant), and specifying the data type (normal or Parkinson's simulated condition).

Hardware Connection Phase

Before data collection can begin, the system checks the availability of the designated serial port used to communicate with the wearable IMU sensor. If the port is unavailabledue to being disconnected or already in usethe system displays an error message and prompts the Operator to either retry or abort the session. If the port is available, the system proceeds to establish communication with the IMU by performing a handshake and initiating the data stream.

Data Collection Loop

This is the core iterative segment of the workflow. For each character trial, the system announces the character and parameters using a built-in Text-to-Speech (TTS) engine, ensuring standardized instruction delivery to the participant. The system then pauses, waiting for the Operator to confirm readiness before initiating the recording.

Once started, the system concurrently logs sensor data to a uniquely named CSV file and visualizes the signal in real-time. The CSV captures timestamped accelerometer and gyroscope data across six axes ($A_x, A_y, A_z, G_x, G_y, G_z$). Real-time quality assurance checks run in parallel, detecting anomalies such as dropped packets, signal clipping, or unusually low motion. The Operator ends the trial manually when writing is complete.

Quality Assurance and Iteration

After each trial, the system performs post-recording quality checks to determine whether the data meets the criteria for inclusion. If the trial fails, it is flagged for redo or annotation, and the system returns to reattempt the same configuration. If the data passes QA, the system marks that character/speed/hand combination as complete. It then checks whether there are remaining characters to be collected for the session. If so, the loop continues with the next character trial.

Session Conclusion

Once all character trials are completed, the session ends. The system exports and updates metadata filesmost importantly, the `users.csv`to reflect the participants updated progress and preserve session integrity. This step ensures that all acquired data is properly recorded and integrated into the overarching dataset.

Sequence Diagram

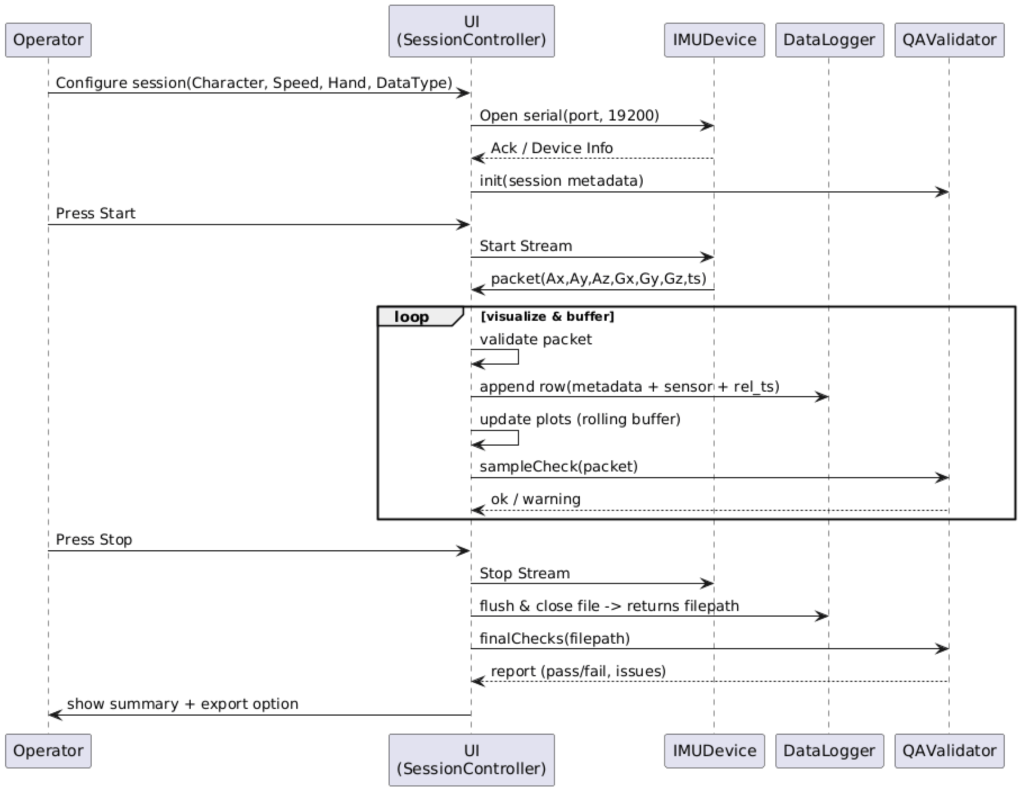


Figure 3.8: Sequence Diagram

Figure 3.8 presents the Sequence Diagram that illustrates the chronological flow of messages and interactions among the core components of the Bengali Air Writing System during the execution of the "Start/Stop Recording" use case. The diagram emphasizes the low-level communication and control flow between the systems software modules and the physical IMU sensor, detailing how data is collected, processed, and stored in real-time.

Actors and Objects

The sequence involves several key participants. The Operator is the human user who initiates and manages the recording session, typically a researcher or lab assistant. The UI, implemented as the `SessionController`, serves as the central graphical interface and orchestrates the entire recording workflow, from configuration to finalization. The `IMUDevice` module handles the physical connection and serial communication with the wearable IMU sensor. The `DataLogger` module is responsible for structured logging of sensor data and associated metadata into CSV files. The `QAValidator` component performs real-time quality assurance checks to maintain the integrity of the data stream during acquisition.

Sequence of Interactions

The use case is divided into three primary phases: session initialization, active data streaming, and session finalization.

In the first phase, the Operator begins by configuring the session parameters through the UI, specifying attributes such as the target character to be written, the writing speed, the dominant hand to be used, and the type of data collection (e.g., normal or simulated Parkinson's condition). Upon confirmation, the UI sends a command to the `IMUDevice` to establish a serial connection with the sensor at a baud rate of 19200. The `IMUDevice` interacts with the physical hardware, which responds with an acknowledgment or device information message, confirming that the connection has been successfully established. Following this, the UI initializes the `DataLogger` with the relevant session metadata, prompting the creation of a new CSV file that includes the necessary headers and identification information.

The second phase begins when the Operator presses the Start button in the UI. This action triggers a command to the `IMUDevice` to begin streaming data from the sensor. Once active, the device emits a continuous stream of data packets containing six degrees of freedom: three axes of accelerometer data (A_x, A_y, A_z), three axes of gyroscope data (G_x, G_y, G_z), and a timestamp (`ts`). Each incoming packet initiates a multi-threaded or parallelized processing loop. The packet is first sent to the `QAValidator`, which performs an initial validation check. This includes verifying that the data values fall within acceptable ranges, ensuring the structural integrity of the packet, and confirming the logical consistency of timestamps. In parallel, the raw packet is also forwarded to the UI for real-time visualization and buffering, allowing the Operator to monitor the sensors performance through live plots. Once validated, the data packet is sent to the `DataLogger`, which appends a new row to the CSV file, including the metadata, sensor values, and a relative timestamp. Additionally, the `QAValidator` may perform deeper inspection through a sample-level check on selected packets, returning a status flag (such as `ok` or `warning`) that can be displayed to the Operator for immediate feedback.

In the final phase, the Operator presses the Stop button to end the recording. The UI sends a command to the `IMUDevice` to stop the data stream and subsequently instructs the `DataLogger` to flush any buffered data and close the file. This step ensures that all data is safely written to disk and that the file is properly finalized. Upon completion, the `DataLogger` returns the file path of the saved session to the UI. To conclude the session, the UI invokes a post-hoc validation by calling the `finalChecks()` method in the `QAValidator`, providing the saved file path. This final check may include analyses such as signal saturation detection, minimum duration verification, and completeness assessment. The `QAValidator` then returns a summary report, including a pass/fail status and any issues identified during analysis. This report is presented to the Operator through the UI, along with options to export the results and data for subsequent review or sharing.

This structured and modular flow of interactions ensures that each recording session follows a reproducible protocol, with built-in safeguards for data integrity and quality assurance. The sequence diagram encapsulates the coordinated behavior of software modules and hardware components, forming the backbone of a reliable Bengali air-writing data collection system.

3.2.5 Research Applications

Table 3.9: Research Applications Enabled by the Bengali Air Writing System

Application Domain	**Description**
Handwriting Analysis	Enables fine-grained examination of stroke-level kinematics, including curvature and variability, facilitating detailed studies of handwriting dynamics.
Motor Skill and Kinematic Studies	Provides comprehensive profiling of movement metrics such as speed, jerk, and smoothness, supporting research in motor control and coordination.
Parkinsons Disease Research	Incorporates simulation flags to support controlled investigations of motor symptoms including tremor, bradykinesia, and micrographia.
Human-Computer Interaction (HCI) for Bengali	Offers a camera-free, privacy-preserving input modality tailored for Bengali text entry systems, enhancing accessibility and user experience.
Machine Learning Datasets	Produces a standardized, metadata-enriched corpus suitable for training and evaluating deep learning models in gesture and handwriting recognition.

By integrating precise hardware components with a research-focused software suite, the Bengali Air Writing System delivers a versatile and extensible platform uniquely suited to address the complexities and specific challenges inherent in Bengali handwriting research.

3.3 Integration and Interconnections

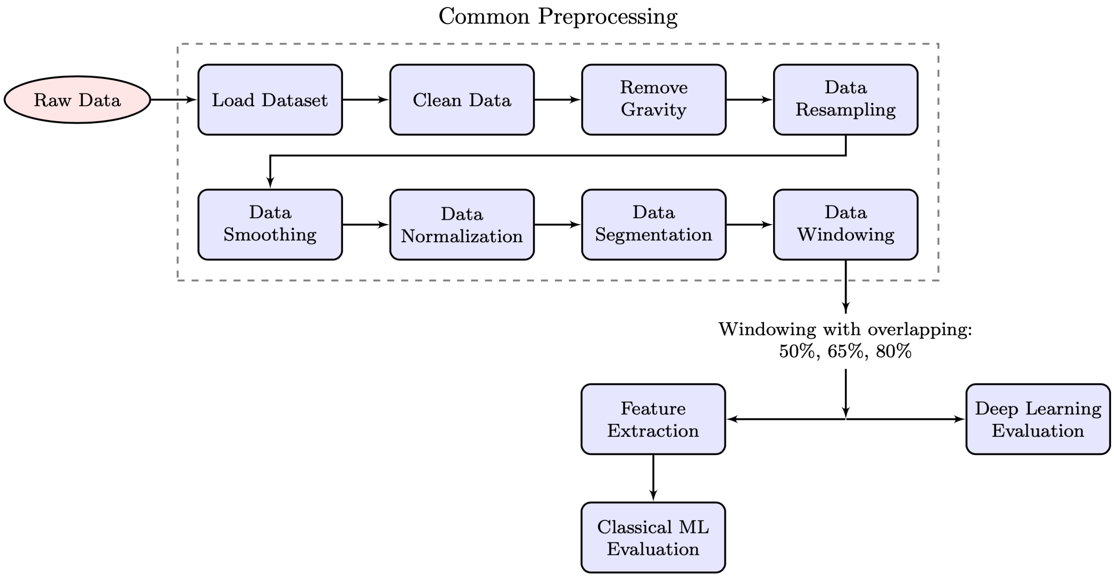
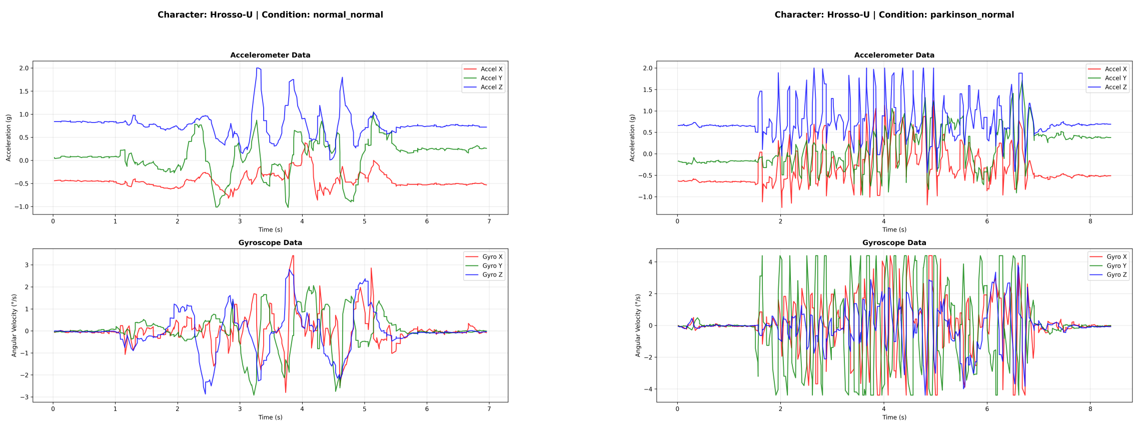


Figure 3.9: Overview of the data acquisition and processing pipeline for the Bengali Air Writing System.

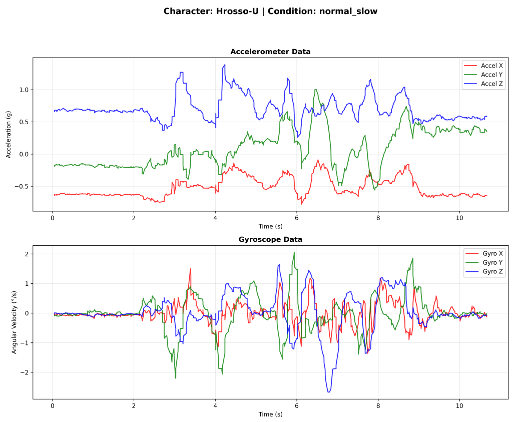
3.3.1 Raw Data → Load Dataset

The process begins with raw sensor streams (accelerometer, gyroscope) collected from motion-sensing devices or existing benchmark datasets. Singh et el. [14] combined UCI-HAR and PAMAP2 datasets to train HAR models, showing that using standardized datasets is the first step toward reproducible experiments.

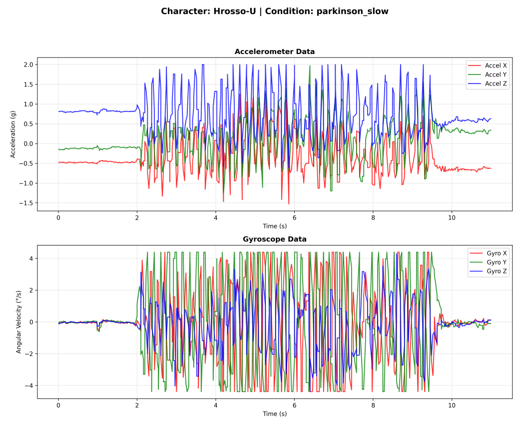


(a) Normal Condition **(b)** Parkinson Condition

Figure 3.10: Comparison of normal handwriting under different conditions.

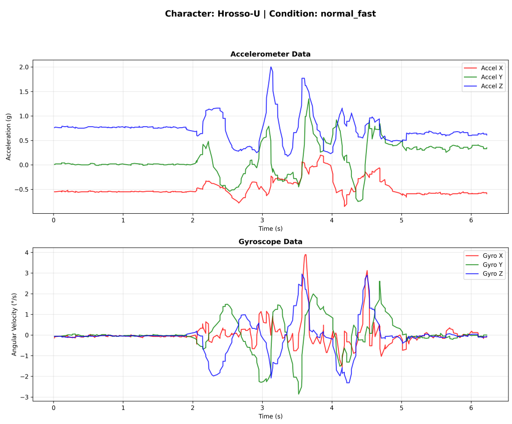


(a) Normal Condition

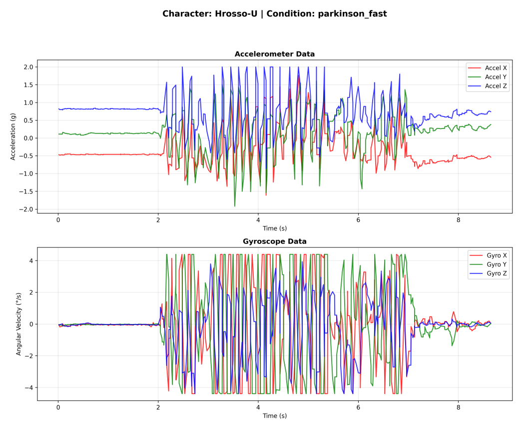


(b) Parkinson Condition

Figure 3.11: Comparison of slow handwriting under different conditions.



(a) Normal Condition



(b) Parkinson Condition

Figure 3.12: Comparison of fast handwriting under different conditions.

3.4 Preprocessing Pipeline

Preprocessing Operations and Equations

Notation

Let $\mathbf{a}_t = [a_{x,t}, a_{y,t}, a_{z,t}]^\top$ be linear acceleration, $\boldsymbol{\omega}_t = [\omega_{x,t}, \omega_{y,t}, \omega_{z,t}]^\top$ angular velocity at time t, sampled (after resampling, if needed) at f_s Hz. Magnitudes: $\|\mathbf{a}_t\| = \sqrt{a_{x,t}^2 + a_{y,t}^2 + a_{z,t}^2}$ and $\|\boldsymbol{\omega}_t\| = \sqrt{\omega_{x,t}^2 + \omega_{y,t}^2 + \omega_{z,t}^2}$.

1. Integrity & Cleaning

Packet validation: discard sample if any field is non-numeric or out of physical range; re-fill short gaps by linear interpolation:

$$x(t_i) = \frac{t_{i+1} - t}{t_{i+1} - t_i} x(t_i) + \frac{t - t_i}{t_{i+1} - t_i} x(t_{i+1}), \quad t \in [t_i, t_{i+1}].$$

2. Resampling to a Uniform Rate f_s

Given irregular timestamps $\{t_n\}$, resample onto $t'_k = \frac{k}{f_s}$ using bandlimited (e.g., polyphase) or linear interpolation:

$$x'(t'_k) \approx \alpha\, x(t_i) + (1-\alpha)\, x(t_{i+1}), \quad \alpha = \frac{t_{i+1} - t'_k}{t_{i+1} - t_i}.$$

3. Gravity Removal (Accelerometer)

(A) Butterworth High-Pass (3rd order): implement in SOS form with cutoff f_c; the discrete filter $H_{\mathrm{HP}}(z)$ yields body acceleration

$$\mathbf{a}_t^{\mathrm{body}} = \mathrm{HPF}_{\mathrm{Butter}, f_c}(\mathbf{a}_t)\,.$$

(B) Exponential Gravity Estimate (practical alternative):

$$\hat{\mathbf{g}}_t = (1-\alpha)\,\hat{\mathbf{g}}_{t-1} + \alpha\,\mathbf{a}_t, \quad \mathbf{a}_t^{\mathrm{body}} = \mathbf{a}_t - \hat{\mathbf{g}}_t, \;\; \alpha = \tfrac{2\pi f_c}{2\pi f_c + f_s}.$$

4. Smoothing / De-noising

(A) Moving-Average (length M):

$$y_t = \frac{1}{M} \sum_{k=0}^{M-1} x_{t-k}.$$

(B) Butterworth Low-Pass (order n, cutoff f_c):

$$y_t = \mathrm{LPF}_{\text{Butter}, f_c}(x_t).$$

5. Normalization (per axis)

z-score:

$$\tilde{x}_t = \frac{x_t - \mu_x}{\sigma_x}, \quad \mu_x = \frac{1}{N} \sum_t x_t, \;\; \sigma_x = \sqrt{\frac{1}{N-1} \sum_t (x_t - \mu_x)^2}.$$

min–max:

$$\tilde{x}_t = \frac{x_t - \min(x)}{\max(x) - \min(x)}.$$

6. Segmentation (Character Clips)

Activity-based gating on norms with hysteresis (thresholds τ_a, τ_ω, dwell H):

$$s_t = \mathbb{I}\big(\|\mathbf{a}_t\| > \tau_a \;\vee\; \|\boldsymbol{\omega}_t\| > \tau_\omega\big), \quad \text{start at first } H \text{ consecutive } s_t = 1, \text{ end at } H \text{ consecutive } s_t = 0.$$

(When trials are pre-cued and file-bounded, this step reduces to trimming leading/-trailing idle.)

7. Sliding Windowing

Windows of length W samples, stride S (overlap $= 1 - \frac{S}{W}$):

$$\mathcal{W}_i = \{\, t \mid t \in [iS,\, iS + W - 1] \,\}, \quad i = 0, 1, \ldots$$

Outputs:

$$\underbrace{\text{tensor} \in \mathbb{R}^{W \times 6}}_{\text{for deep models}} \quad \text{or} \quad \underbrace{\phi(\mathcal{W}_i) \in \mathbb{R}^d}_{\text{for classical ML features}}.$$

$$\mathrm{RMS}(x) = \sqrt{\frac{1}{W}\sum_{t\in\mathcal{W}} x_t^2}, \quad \mathrm{ZCR}(x) = \frac{1}{W-1}\sum_{t\in\mathcal{W}-\{t_{\max}\}} \mathbb{I}[x_t x_{t+1} < 0].$$

(Hjorth and spectral metrics can be computed per axis; correlations across $\{x, y, z\}$ for accel and gyro.)

Implementation notes

Unless stated otherwise, we (i) de-trend by subtracting the per-window mean; (ii) optionally remove residual gravity from $\mathbf{a}(t)$ via a moving-average high-pass (cutoff near 1 Hz) if not already filtered; (iii) apply robust outlier clipping (e.g., 0.1/99.9 percentiles) to curb spikes; and (iv) band-limit to a physiologically plausible range (e.g., 0.515 Hz for handwriting accelerometry, higher for gyroscopes), with an anti-aliasing low-pass at $f_s/2$.

3.5 Feature Extraction (for Machine Learning)

From each window, statistical, temporal, and frequency features are extracted (mean, variance, entropy, FFT coefficients, correlation, etc.) for classical ML models. [6] describe common hand-crafted features like mean, energy, entropy, correlation from IMU signals.

3.5.1 Time-Domain Features

For each axis within a window W:

1. **Root-Mean-Square (RMS):**

$$\mathrm{RMS}(x) = \sqrt{\frac{1}{L}\sum_{n=0}^{L-1} x[n]^2},$$

capturing instantaneous motion intensity (stroke vigor; angular-speed energy).

2. **Shannon Entropy (amplitude PDF):** Construct a B-bin histogram with probabilities p_b ($\sum_b p_b = 1$):

$$H_t(x) = -\sum_{b=1}^{B} p_b \log p_b, \qquad H_t^{\star}(x) = \frac{H_t(x)}{\log B} \in [0, 1],$$

measuring amplitude unpredictability in a window (normalized for cross-window comparability).

3. **Zero-Crossing Rate (ZCR):**

$$\mathrm{ZCR}(x) = \frac{1}{2(L-1)} \sum_{n=1}^{L-1} \big| \operatorname{sgn}(x[n]) - \operatorname{sgn}(x[n-1]) \big|,$$

a time-domain proxy for oscillation frequency, robust to amplitude scaling.

4. **Interquartile Range (IQR):**

$$\mathrm{IQR}(x) = Q_{0.75}(x) - Q_{0.25}(x),$$

a robust dispersion statistic distinguishing broad sweeps from tight strokes/holds.

3.5.2 Frequency-Domain Features (Welch PSD)

We estimate the single-axis PSD $S_x[k]$ via Welchs method with Hann subwindows and define the normalized spectrum

$$P_x[k] = \frac{S_x[k]}{\sum_j S_x[j]}, \qquad f_k = \frac{k}{L} f_s.$$

Feature set:

1. **Spectral Entropy:**

$$H_f(x) = -\sum_k P_x[k] \log P_x[k], \qquad H_f^\star(x) = \frac{H_f(x)}{\log K} \in [0,1],$$

indicating spectral flatness/complexity.

2. **Spectral Centroid:**

$$f_c(x) = \frac{\sum_k f_k S_x[k]}{\sum_k S_x[k]},$$

the intensity-weighted spectral “center of mass” (related to stroke cadence).

3. **Spectral Energy:**

$$E_f(x) = \sum_k S_x[k],$$

equivalent to time-domain energy (Parseval).

4. **Dominant Frequency & Peak Prominence:**

$$f_{\max}(x) = \arg\max_k S_x[k], \qquad \Pi(x) = \frac{S_x[k_{\max}]}{\mathrm{median}_k S_x[k]},$$

tracking the principal oscillation and its salience over background.

3.5.3 Magnitude-Based, Rotation-Agnostic Features

Define magnitudes

$$\|\mathbf{a}\|[n] = \sqrt{a_x[n]^2 + a_y[n]^2 + a_z[n]^2}, \qquad \|\boldsymbol{\omega}\|[n] = \sqrt{\omega_x[n]^2 + \omega_y[n]^2 + \omega_z[n]^2}.$$

We compute:

1. **Mean and Variance:**

$$\mu_m = \frac{1}{L}\sum_n \|\mathbf{u}\|[n], \qquad \sigma_m^2 = \frac{1}{L-1}\sum_n \left(\|\mathbf{u}\|[n] - \mu_m\right)^2, \quad \mathbf{u} \in \{\mathbf{a}, \boldsymbol{\omega}\}.$$

2. **Signal Magnitude Area (SMA):**

$$\mathrm{SMA} = \frac{1}{L}\sum_{n=0}^{L-1} \Big(|a_x[n]| + |a_y[n]| + |a_z[n]|\Big),$$

with an analogous definition for $\boldsymbol{\omega}$.

3.5.4 Cross-Axis Coupling (Shape and Coordination)

Let μ_x and σ_x denote per-axis mean and standard deviation within a window.

1. **Covariance Matrix:**

$$\Sigma_{ij} = \frac{1}{L-1}\sum_{n=0}^{L-1} \left(x_i[n] - \mu_{x_i}\right)\left(x_j[n] - \mu_{x_j}\right), \quad i, j \in \{x, y, z\}.$$

2. **Pearson Correlation:**

$$\rho_{ij} = \frac{\Sigma_{ij}}{\sigma_{x_i}\sigma_{x_j}}, \qquad \rho_{ij} \in [-1, 1].$$

We compute Σ and ρ for both $\mathbf{a}$ and $\boldsymbol{\omega}$ to capture planar/3D coordination; characters with loops/curls induce characteristic cross-axis patterns distinct from straight strokes.

3.5.5 Window-to-Trial Aggregation

For a multi-second trial with T windows and window-level features ϕ_w, we form trial-level descriptors via robust pooling:

$$\phi_{\text{trial}} = \Big[\, \text{median}_w(\phi_w),\ \text{IQR}_w(\phi_w),\ \max_w(\phi_w) \Big],$$

where median/IQR attenuate within-trial outliers (e.g., start/end effects), and maxima capture peak dynamics.

3.5.6 Feature Vector Construction

For each fixed-length analysis window (optionally pooled at the trial level), we construct a multivariate descriptor from tri-axial acceleration $\mathbf{a} = [a_x, a_y, a_z]^\top$ and angular velocity $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^\top$. Let

$$\mathcal{S} = \{a_x, a_y, a_z, \omega_x, \omega_y, \omega_z\}$$

denote the six sensor axes.

Axis-wise time-domain descriptors
For each $s \in \mathcal{S}$, we compute: root-mean-square RMS, normalized temporal entropy $H_t^\star$, zero-crossing rate ZCR, and interquartile range IQR (4×6 scalars).

Axis-wise frequency-domain descriptors
For each $s \in \mathcal{S}$, we compute: normalized spectral entropy $H_f^\star$, spectral centroid f_c, band energy E_f, peak frequency $f_{\max}$, and a spectral proportion statistic Π (5×6 scalars).

Magnitude-based descriptors
From the Euclidean norms $||\mathbf{a}||$ and $||\boldsymbol{\omega}||$, we compute the mean μ_m, variance σ_m^2, and signal magnitude area SMA ($3 \times 2 = 6$ scalars).

Cross-axis dependence descriptors
Within each modality ($\mathbf{a}$ and $\boldsymbol{\omega}$), we form the 3×3 sample covariance matrix Σ and correlation matrix ρ. We then vectorize their upper-triangular parts via $\text{vec}_\triangle(\cdot)$ to summarize inter-axis couplings (scale-dependent via Σ, scale-invariant via ρ). Each matrix contributes $u \in \{3, 6\}$ unique elements per modality (off-diagonal only or including diagonal, respectively).

Concatenation and dimensionality

$$\mathbf{z}_w = \Big[\underbrace{\{\mathrm{RMS},\, H_t^{\star},\, \mathrm{ZCR},\, \mathrm{IQR}\}_{s\in\mathcal{S}}}_{4\times 6},\ \underbrace{\{H_f^{\star},\, f_c,\, E_f,\, f_{\max},\, \Pi\}_{s\in\mathcal{S}}}_{5\times 6},$$
$$\underbrace{\mu_m,\, \sigma_m^2,\, \mathrm{SMA\ on}\ \|\mathbf{a}\|,\, \|\boldsymbol{\omega}\|}_{6},\ \underbrace{\mathrm{vec}_{\triangle}(\Sigma_{\mathbf{a}}),\, \mathrm{vec}_{\triangle}(\rho_{\mathbf{a}}),\, \mathrm{vec}_{\triangle}(\Sigma_{\omega}),\, \mathrm{vec}_{\triangle}(\rho_{\omega})}_{4u}\Big]^{\top}. \tag{3.1}$$

$$d = 4\cdot 6 + 5\cdot 6 + 6 + 4u = 60 + 4u, \qquad u \in \{3, 6\}. \tag{3.2}$$

3.6 Classical ML Evaluation

Extracted features are fed into ML models (e.g., SVM, Random Forest). Model performance is usually reported with accuracy, precision, recall, or F1. Wang et el. [18] compared multiple classifiers (SVM, Random Forest) and reported cross-validated accuracy.

3.7 Deep Learning Evaluation

In order to evaluate the efficacy of recurrent neural network architectures for motion-based character recognition, we implemented and compared bidirectional long short-term memory (Bi-LSTM) and gated recurrent unit (GRU) models. Both approaches were trained on the windowed inertial datasets described in Sec. **??**.

3.8 Data Preparation

For both architectures, the input consisted of windowed CSV files from the `Windowed_Data/` directory. Each sample retained six channels: accelerometer signals (a_x, a_y, a_z) and gyroscope signals ($\omega_x, \omega_y, \omega_z$). Class labels were derived directly from file names by tokenizing on underscores and extracting the class identifier. Labels were subsequently transformed into integer indices via `LabelEncoder` and converted to one-hot encodings for classification.

Prior to training, we verified dimensional consistency across all samples; any files with mismatched feature counts were excluded. For the GRU model, an additional preprocessing step was introduced: all sequences were aligned to a fixed length by trimming longer samples and zero-padding shorter ones.

3.9 Data Partitioning

We adopted a hold-out validation strategy. For the Bi-LSTM model, the dataset was randomly partitioned into 80% training and 20% test sets. From the training portion, 10% was further reserved for validation during optimization. The GRU model followed a similar 80/20 split, with the test set directly supplied as validation data during training. In all cases, the random seed was fixed to ensure reproducibility.

3.10 Model Architectures

3.10.1 Bi-LSTM Encoder

The Bi-LSTM architecture consisted of bidirectional recurrent layers with dropout regularization applied within the encoder and prior to the classification layer. The encoder output was fed into a fully connected dense layer with softmax activation for categorical prediction.

3.10.2 GRU Encoder

The GRU-based sequence model employed gated recurrent units as the temporal encoder. The final sequence output was collapsed into a fixed-length latent representation, which was mapped to class probabilities through a dense softmax layer. Unlike attention-based models, no explicit temporal weighting was used.

3.11 Training Configuration

Both models were trained using the Adam optimizer with categorical cross-entropy loss. A mini-batch size of 32 was used, and training proceeded for a maximum of 50 epochs. Early stopping on validation loss was enabled in both cases, with `restore_best_weights=True` to retain the parameters from the epoch yielding the lowest validation error. The Bi-LSTM employed light dropout regularization, while the GRU training employed early stopping with a patience of five epochs.

3.12 Evaluation

Model performance was quantified using classification accuracy on the held-out test set. The Bi-LSTM achieved a test accuracy of 94%, demonstrating strong generalization under the chosen configuration. In contrast, the GRU model obtained a test accuracy of only 23.48%, indicating underfitting and poor discriminative performance in the absence of more advanced temporal modeling mechanisms.

Overall, these results highlight the superiority of bidirectional LSTM encoders for capturing the sequential dependencies inherent in handwriting-related inertial signals.

3.13 Conclude with a Summary of the Methodology

The system begins by capturing raw accelerometer and gyroscope data generated from air-written gestures. This data is then processed through a series of steps to convert it into structured feature vectors. The first stage of processing involves smoothing, normalization and statistical analysis to clean and standardize the data, ensuring its suitability for further use.

Once the data is prepared, feature selection techniques are employed to identify the most relevant information, which helps optimize the performance of the subsequent machine learning model. These refined feature vectors are then fed into a model designed for character recognition, ensuring accurate mapping of the gestures to their corresponding Bengali characters.

Each step of this methodologyfrom data collection to feature extraction and model trainingworks synergistically to build a robust system capable of reliably interpreting air-written Bengali gestures. This end-to-end pipeline not only highlights the coherence and effectiveness of the approach but also sets the stage for the forthcoming chapters on Results and Discussion, where we will analyze the performance and implications of this method.

Chapter 4

Results and Discussion

4.1 Datasets and Experimental Setup

The Bengali air-writing dataset introduced in this study consists of 3,996 trials across 60 symbol classes (vowels, consonants, and digits). Each trial is captured at 200 Hz using a six-degree-of-freedom inertial measurement unit (3-axis accelerometer and 3-axis gyroscope). The collection protocol systematically varied three factors: handedness (dominant vs. non-dominant), writing speed (slow, normal, fast), and environment (normal vs. noisy). This factorial design ensured diversity in kinematic patterns and robustness to real-world variability.

For evaluation, Writer-dependent split data partitioning strategy were applied. A writer-dependent split partitions data so that the same writers appear in both training and testing sets, meaning the model is evaluated on handwriting styles it has already seen. [3]

All classical models were trained on extracted feature vectors, while deep learning baselines operated directly on segmented time-series. Feature extraction was implemented using NumPy, SciPy, and scikit-learn. Deep learning experiments were conducted in PyTorch. Training was performed on a CPU-only environment (Intel i5, 16 GB RAM), with early stopping to prevent overfitting.

4.2 Presenting the Results

We report results under both classical machine learning and deep learning paradigms. Metrics include classification accuracy and macro-averaged F1-score, which balances the impact of class imbalance across Bengali graphemes.

4.2.1 Classical Machine Learning Models

Two baseline classifiers were implemented using standardized feature vectors:

1. Linear Support Vector Machine (SVM)
2. Random Forest

The results are summarized in Table 4.1, where each models accuracy and macro-F1 are presented. These values provide a lightweight and interpretable baseline for future benchmarking.

Table 4.1: Performance of Classical Machine Learning Models on the Bengali Air-Writing Dataset.

Model	Accuracy (%)
Linear SVM	70.32%
Random Forest	96.49%

The Random Forest achieved the highest accuracy among classical models, while SVM showed comparatively weaker performance, suggesting the need for further hyperparameter tuning.

4.2.2 Deep Learning Baselines

To validate the feasibility of sequential modeling, three architectures were tested:

- **Bi-LSTM:** Achieved strong performance, leveraging bidirectional dependencies.
- **Bi-GRU:** Achieved the highest accuracy when trained and tested on the same dataset split.
- **GRU (without attention):** Performed poorly, suggesting the need for architectural refinements.

4.3 Interpreting the Results

Two main observations emerge from the experiments:

1. **Classical feasibility:** Even with compact feature vectors, models such as Random Forest achieved very high accuracy (96.49%), validating that the dataset contains strong discriminative patterns.

2. **Deep learning potential:** Bi-LSTM demonstrated strong sequence modeling capability

4.4 Challenges and Limitations

Despite establishing strong baselines, several limitations remain:

- **Computational resources:** Deep learning experiments were restricted to small-scale runs due to hardware constraints.
- **Cross-condition robustness:** While designed with hand, speed, and environment variation, models struggled to generalize across these axes.
- **Granularity of classes:** The dataset includes many visually similar characters, which significantly raises error rates in confusable subsets.

Chapter 5

Conclusion

5.1 Restating Research Objectives and Questions

This study set out to explore the feasibility of recognizing Bengali characters written in mid-air using motion sensor data. The core objectives were twofold: first, to construct a comprehensive 3D motion-sensor dataset encompassing Bengali vowels, consonants and digits; and second, to design and evaluate machine learning models capable of accurately interpreting this gesture data. These objectives were driven by the overarching research questions: Can Bengali characters be reliably recognized from accelerometer and gyroscope data collected during air-writing? And which features and classification algorithms contribute most effectively to recognition accuracy? These guiding questions shaped the design, methodology and analysis presented throughout the thesis and the results discussed in the following sections offer a data-driven response to them, affirming the viability of mid-air character recognition using motion sensors.

5.2 Summary of Key Findings

The key findings are:

- A sizable dataset of 60 out of 60 Bengali symbol classes (character strokes), totaling of 3,996 samples has been collected with a wearable device.
- Feature extraction from accelerometer/gyroscope signals enabled character classification by standard Machine Learning and Deep Learning models.
- Confusion mainly occurred in visually similar shapes, suggesting which sym-

bols need more data.

- In general, motion-sensor input proves to be a viable modality for Bengali text entry, aligning with findings in other languages.

5.3 Contributions of the Research

This study contributes:

- **A novel dataset:** We assembled the first publicly described Bengali air-writing dataset captured via motion sensors, covering all basic script symbols with multiple user samples.
- **Data acquisition system:** A wearable wrist watch type device and companion app were implemented to record hand-motion trajectories and transfer them via Bluetooth, advancing prior hardware setups.
- **Features:** We evaluated a range of time-domain, statistical and spectral features.

5.4 Acknowledging Limitations

We acknowledge several limitations. First, the dataset size and variety are still modestmore participants and samples are needed to cover handwriting diversity fully. Second, the current system is evaluated offline on individual characters; continuous writing and real-time deployment remain to be tested. Third, the preliminary nature of this work means results are subject to change as data grows. Lastly, we have not yet fully explored machine-learning architectures due to data constraints. These limitations mean that reported accuracies should be viewed as initial rather than definitive.

5.5 Suggestions for Future Research

Future work should expand the dataset (more users, writing styles and possibly entire words) to improve robustness. Research could also investigate adaptive learning to handle user-dependent versus user-independent cases. Integration into a real-time prototype with user feedback would be valuable. Cross-modal fusion (e.g., combining our device with camera vision) may further enhance performance. Finally, open-sourcing the dataset and models would enable the community to build upon this foundation.

5.6 Final Reflections

Overall, this repost highlights the interdisciplinary blend of Bengali linguistics, signal processing and machine learning. It underscores how even complex non-Latin scripts can be integrated into novel input interfaces. The collaboration of hardware engineering (wearables), data science and language knowledge is a hallmark of modern HCI research. As with many ongoing projects, early results are encouraging but invite iteration. By sharing methods and findings, we anticipate that further innovation in Bengali air-writing and input technologies for other languages will follow.

In conclusion, this pre-defense study has shown that Bengali air-writing with motion sensors is both feasible and promising. We have outlined the steps toward a rigorous recognition system: creating the dataset, extracting meaningful features. Our contributions address the critical need for Bengali language support in next-generation interfaces. As the research progresses, we expect to refine and extend these results, ultimately enabling seamless, contactless Bengali text entry and enriching the field of human-computer interaction.

References

[1] M. S. Alam, K. C. Kwon, M. A. Alam, M. Y. Abbass, S. M. Imtiaz, and N. Kim, "Trajectory-based air-writing recognition using deep neural network and depth sensor," *Sensors*, vol. 20, no. 2, p. 376, 2020. DOI: 10.3390/s20020376

[2] C. Amma, M. Georgi, and T. Schultz, "Airwriting: A wearable handwriting recognition system," *Personal and Ubiquitous Computing*, vol. 18, no. 1, pp. 191–203, 2014. DOI: 10.1007/s00779-013-0637-3

[3] A. Brink, M. Tahir, J. Fierrez, and R. Guest, "Knowledge transfer for writer identification," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications (CIARP 2017)*, ser. Lecture Notes in Computer Science, vol. 10657, Springer, 2017, pp. 163–170. DOI: 10.1007/978-3-319-75193-1_13

[4] R. R. Chowdhury, M. S. Hossain, R. U. Islam, K. Andersson, and S. Hossain, "Bangla handwritten character recognition using convolutional neural network with data augmentation," in *2019 Joint 8th International Conference on Informatics, Electronics and Vision (ICIEV) and 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, IEEE, 2019, pp. 91–95. DOI: 10.1109/ICIEV.2019.8858545

[5] O. De, P. Deb, S. Mukherjee, S. Nandy, T. Chakraborty, and S. Saha, "Computer vision based framework for digit recognition by hand gesture analysis," *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 1–5, Oct. 2016. DOI: 10.1109/iemcon.2016.7746361 [Online]. Available: https://doi.org/10.1109/iemcon.2016.7746361

[6] W. H. Gomaa and A. Khamis, "Feature engineering and preprocessing techniques for inertial data in har," *Sensors*, vol. 23, no. 5, pp. 1125–1142, 2023.

[7] Y. Gu, H. Oku, and M. Todoh, "American sign language recognition and translation using perception neuron wearable inertial motion capture system," *Sensors*, vol. 24, no. 2, p. 453, 2024. DOI: 10.3390/s24020453

[8] C.-H. Hsieh, Y.-S. Lo, J.-Y. Chen, and S.-K. Tang, "Air-writing recognition based on deep convolutional neural networks," *IEEE Access*, vol. 9, pp. 142 827–142 836, 2021. DOI: `10.1109/ACCESS.2021.3121093`

[9] H. Jeon, H. Choi, D. Noh, T. Kim, and D. Lee, "Wearable inertial sensor-based hand-guiding gestures recognition method robust to significant changes in the body-alignment of subject," *Mathematics*, vol. 10, no. 24, p. 4753, 2022. DOI: `10.3390/math10244753`

[10] F. Ott and M. Wehbi, "The onhw dataset: Online handwriting recognition from imu-enhanced ballpoint pens with machine learning," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 2, pp. 1–23, 2020. DOI: `10.1145/3411842`

[11] A. Qedear, A. AlMatrafy, A. Al-Sowat, A. Saigh, and A. Alayed, "Real-time air-writing recognition for arabic letters using deep learning," *Sensors*, vol. 24, no. 18, p. 6098, 2024. DOI: `10.3390/s24186098`

[12] C. Qu, "Online kinect handwritten digit recognition based on dynamic time warping and support vector machine," *Journal of Information and Computational Science*, vol. 12, no. 1, pp. 413–422, Jan. 2015. DOI: `10.12733/jics20105578` [Online]. Available: `https://doi.org/10.12733/jics20105578`

[13] M. Raquib, M. Hossain, K. Islam, and M. S. Miah, "Vashanet: An automated system for recognizing handwritten bangla basic characters using deep convolutional neural network," *Machine Learning with Applications*, vol. 17, p. 100 568, 2024. DOI: `10.1016/j.mlwa.2024.100568`

[14] R. Singh, A. Kumar, and Y. Zhao, "Benchmarking human activity recognition using uci-har and pamap2: Toward reproducible experiments," *IEEE Transactions on Human-Machine Systems*, 2025.

[15] S. K. Singh and A. Chaturvedi, "Machine learning for sensor-based handwritten character recognition: A brief survey," in *Distributed Computing and Intelligent Technology*, Springer, 2024, pp. 288–305. DOI: `10.1007/978-3-031-81404-4_21`

[16] Y. Song, M. Liu, F. Wang, J. Zhu, A. Hu, and N. Sun, "Gesture recognition based on a convolutional neural network–bidirectional lstm network for a wearable wrist sensor with multi-walled carbon nanotube/cotton fabric material," *Micromachines*, vol. 15, no. 2, p. 185, 2024. DOI: `10.3390/mi15020185`

[17] H. Wang et al., "Mems devices-based hand gesture recognition via wearable computing," *Micromachines*, vol. 14, no. 5, p. 947, 2023. DOI: `10.3390/mi14050947`

[18] L. Wang, T. Zhang, and X. Liu, "Improving gesture recognition with signal filtering and normalization," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 3, pp. 1129–1144, 2023.

[19] T. Watanabe, M. Maniruzzaman, M. A. Hasan, H.-S. Lee, S.-W. Jang, and J. Shin, "2d camera-based air-writing recognition using hand pose estimation and hybrid deep learning model," *Electronics*, vol. 12, no. 4, p. 995, 2023. DOI: 10.3390/electronics12040995

[20] M. Wehbi et al., "Towards an imu-based pen online handwriting recognizer," in *Document Analysis and Recognition ICDAR 2021*, Springer, 2021, pp. 289–303. DOI: 10.1007/978-3-030-86334-0_19

[21] H. Zhang et al., "A wearable real-time character recognition system based on edge computing-enabled deep learning for air-writing," *Sensors*, vol. 2022, 2022. DOI: 10.1155/2022/8507706