IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 July 2025, accepted 19 July 2025, date of publication 23 July 2025, date of current version 31 July 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3592079

RESEARCH ARTICLE

Design and Evaluation of ADANet: A High-Fidelity Motion Acquisition Framework for Assistive Gesture-Based Interfaces

MD ETTASHAMUL HAQUE[1], ATIQUE TAJWAR[1], AKM AZAD[2,3], SALEM A. ALYAMI[2,3], (Member, IEEE), AND MD MEHEDI HASAN[1]

[1]Department of Robotics and Mechatronics Engineering, University of Dhaka, Dhaka 1000, Bangladesh
[2]Department of Mathematics and Statistics, Faculty of Science, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 13318, Saudi Arabia
[3]King Salman Center for Disability Research, Riyadh 11614, Saudi Arabia

Corresponding author: Md Mehedi Hasan (mmhasan@du.ac.bd)

The authors extend their appreciation to the King Salman center For Disability Research for funding this work through Research Group no KSRG-2024-113.
This work involved human subjects in its research. Approval of all ethical and experimental procedures and protocols was granted by the Ethical Committee of the Faculty of Engineering and Technology, University of Dhaka, under Application No. 274/2025.

ABSTRACT Accurate and low-latency gesture recognition is critical for real-time assistive technologies that enable individuals with motor impairments to interact more intuitively with their environment. However, current systems often suffer from poor signal fidelity, limited adaptability, and high computational overhead. To address these limitations, this article presents ADANet — Advanced Disability Assistive Neural Network-driven framework for accelerometer-based gesture recognition, tailored for wearable assistive applications. The proposed system combines a high-fidelity ADXL335 triaxial accelerometer with an ESP32 microcontroller, forming a lightweight and cost-efficient motion acquisition pipeline. A structured preprocessing architecture is developed, incorporating zero-lag Butterworth filtering, entropy-based temporal smoothing, and computation of 33 handcrafted statistical features, including axis-specific jerk, signal magnitude area (SMA), and range-normalized entropy. ADANet's compact yet expressive neural architecture is optimized through comprehensive ablation studies to ensure strong generalization with minimal latency. Our model was trained on data from 21 healthy participants(12 male & 9 female) performing eight functional gestures demonstrate a test accuracy of 84.47%, F1-score of 0.84, and AUC$\geq$0.96, outperforming traditional classifiers such as XGBoost, SVM, and KNN across multiple evaluation metrics. This work validates a robust end-to-end instrumentation-to-inference framework and establishes ADANet as a practical solution for embedded gesture recognition in assistive devices, mostly for disabilities, where signal integrity, adaptability, and inference efficiency are jointly critical.

INDEX TERMS Gesture recognition, accelerometer, signal acquisition system, neural network, machine learning, human–computer interface (HCI).

I. INTRODUCTION

Assistive technologies for individuals with physical disabilities serve as essential tools to restore autonomy, dignity, and independence in daily life. Wearable systems and prosthetic solutions have become increasingly integral in supporting those with limited manual dexterity, enabling interaction with their environment through intuitive interfaces. Gesture-controlled assistance, in particular, has emerged as a promising field due to its natural interaction paradigm, offering nonverbal control mechanisms for navigation, communication, and actuation. With the proliferation of embedded systems and edge computing capabilities, wearable assistive devices are transitioning from bulky and specialized setups to compact, real-time, and user-friendly systems. These advances are vital for people with disabilities to achieve frictionless participation in everyday tasks.

The associate editor coordinating the review of this manuscript and approving it for publication was Mauro Fadda.

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

At the core of these innovations lies the strategic use of motion-sensing instrumentation, especially triaxial accelerometers, which translate human limb movement into measurable signals. Gesture recognition, in this context, depends heavily on the accurate capture of time-series data reflecting dynamic hand motions and postures. Accelerometers provide a reliable and low-power means to detect gesture patterns through features such as velocity changes, orientation shifts, and motion intensity. However, capturing high-fidelity gestures requires addressing challenges such as signal drift, noise interference, sensor placement variation, and environmental disturbances. The quality of gesture-based interaction hinges on effective signal acquisition and preprocessing, filtering, normalization, segmentation, as well as careful calibration and sampling strategies. From an instrumentation point of view, designing robust gesture recognition systems means ensuring measurement consistency across varied users and contexts while maintaining real-time responsiveness on constrained platforms.

In terms of computational approaches, traditional algorithms such as Dynamic Time Warping (DTW) [1] and Hidden Markov Models (HMM) [2] laid early foundations for gesture classification but have notable limitations: sensitivity to gesture speed variations, computational inefficiency, and dependence on extensive training. With the emergence of neural networks (NNs), gesture recognition has entered a new phase, marked by robustness to signal noise and user variability, and adaptability to diverse movement patterns [3]. Gestures enable natural and intuitive robot control, while NN-based gesture recognition enables real-time interaction in manufacturing, healthcare, human-robot interaction, authentication, and so on. In addition, NN-based gesture recognition improves the ease and precision of controlling assistive technologies such as prosthetic devices for individuals with disabilities [4]. However, implementing these architectures on low-power platforms requires attention to model optimization, real-time inference, and instrument-aware feature extraction. Future directions in NN-based gesture recognition include improving user-independent recognition, optimizing NN models for resource-constrained devices, and recognizing complex and continuous gestures. Focusing on that, the contributions of this paper are as follows.

- **Low-Cost Gesture Recognition System for Edge Computing:** This study presents a cost-effective wearable device utilizing ESP32 for accelerometer-driven gesture recognition. The wearable is designed to enhance cost-effectiveness and efficiency for scalable edge computing.
- **Instrumentation-Driven Data Acquisition and Preprocessing Pipeline:** A robust acquisition framework is implemented, featuring precise temporal sampling, noise mitigation, and signal normalization. This ensures consistency and reliability in motion data across diverse users and conditions.
- **Optimized NN Architecture for Resource-Limited Platforms:** The development of a neural network that is both compact and efficient was accomplished by striking a balance between the computational efficiency and the reliability of real-time gesture recognition.
- **Advanced Feature Engineering and Discriminative Signal Processing:** The system integrates statistical and entropy-based features to enhance class separability. This instrumentation-aware feature extraction improves robustness against signal overlap and variability in gesture execution.
- **Demonstrated Scalability and Multidisciplinary Applicability:** The proposed system paves the way for integration into fields such as Human-Robot Interaction, assistive technologies, and gaming, ensuring adaptability for future multimodal and real-world applications.

II. LITERATURE REVIEW

The evolution of gesture recognition technologies has been driven by the pursuit of seamless interaction between humans and machines. The two primary system types in this field are sensor-based and vision-based. Each type is appropriate for a variety of applications and employs distinct techniques. Vision-based gesture recognition (VGR) [5], [6] systems utilize camera images to interpret hand movements. Researchers have explored various VGR techniques including motion-based detection [7], appearance detection [8], skin color detection [9], skeleton-based detection [10] and depth detection [11]. Different camera technologies, such as RGB, IR-based, and Microsoft Kinect cameras [12], have been employed for VGR. Despite promising results, VGR systems are susceptible to environmental factors such as lighting [13], line of sight, and detector proximity. Depth-based and skeleton-based methods, while more robust, haven't completely overcome these limitations. RGB-D [14] techniques, for instance, can struggle in environments with high contrast or an absence of reflective surfaces. On the contrary, due to their low cost, small size, and low power consumption, accelerometers are used in smartphones and wearable devices for signal acquisition and gesture recognition. Many studies utilize data from a single 3-axis accelerometer [15], [16], [17], while others incorporate Inertial Measurement Units (IMUs) that combine accelerometers with gyroscopes to capture angular velocity in addition to acceleration, providing richer motion information [18].

Different studies employ varying sampling frequencies, a critical parameter that influences both the quality of the captured signal and the computational resources, depending on the application and the gestures. For instance, gestures captured using smartphones with accelerometers and gyroscopes have been sampled at 50 Hz [18]. The UCI-HAR and UniMiB-SHAR datasets recorded accelerometer data at 50 Hz [19]. Studies using MMG signals with accelerometers have employed sampling rates of 200 Hz [20], noting that traditional MMG research sometimes used higher rates

TABLE 1. Recent studies on hand gesture recognition.

Related Papers	Dataset	Methods	Metrics
Kim *et al*. (2019) [6]	10 Digits (3D)	RCE NN, Dynamic Time Warping	Overall accuracy 98.6%
Xie *et al*. (2016) [13]	24 gestures (8 basic, 16 complex)	FNN, SVM	User-Dependent: 99.88% User-Independent: 98.88%
Akl *et al*. (2011) [16]	18 Gestures	DTW, AP, CS	User-Dependent: 99.81–98.71% User-Independent: 96.84–94.60%
Anderez *et al*. (2019) [17]	8 Gestures	Crossings-based segmentation, TCNN	User-Dependent: 99.1% User-Independent: 98.8%
Chu *et al*. (2020) [18]	11 Gestures	Residual PairNet with Inception	Overall accuracy <80%
Liu *et al*. (2020) [20]	8 Gestures	Wavelet Packet Decomposition, KNN, SVM, LDA	Overall accuracy 90.74%
Pan *et al*. (2022) [21]	21 gestures	sEMG, IMU, DBN, SVM	Precision: 92.22% Recall: 91.01%
Koch *et al*. (2019) [25]	3 Datasets from Ninapro	Single LSTM Cell (size 256)	DB2: 81.8% DB3: 70.1%
Wu *et al*. (2009) [26]	12 gestures	FDSVM	User-Dependent: 95.21% User-Independent: 89.29%
Liu *et al*. (2009) [27]	8 gestures	DTW	Same-Day: 98.4% All-Day: 93.5%
Moiz *et al*. (2011) [28]	4 directional gestures	TDNN	User-Dependent: 99.2% User-Independent: 98.4%

like 1000 Hz before filtering. For sports referee training, sEMG signals were captured at 200 Hz [21]. Often a trade-off is required: lower sampling rate might miss important high-frequency components of the movement, potentially reducing recognition accuracy while higher rate require more computational resources [19].

The signal processing pipeline interprets accelerometer data for gesture recognition. To ensure accurate interpretation, digital filters are commonly applied to eliminate noise and artifacts. Low-pass filters reduce noise and separate low-frequency components like gravitational acceleration [22]. High-pass filters help in removing DC offsets and low-frequency disturbances, including the gravity component from accelerometer data [23]. Band-pass filters are employed to isolate the frequency range of interest, such as the 5-50 Hz range for MMG signals [20] to eliminate motion artifacts. Other filtering techniques include rolling median filters for spike removal and moving average filters for smoothing the signal and reducing high-frequency noise [13]. Studies highlight its importance, showing that lowering the sampling rate or resolution can reduce accuracy [24]. Techniques like high-pass filtering, as used by Liu et al. [20], remove motion artifacts while preserving key information. Other methods, such as mean subtraction and standard deviation scaling, improve data consistency, as seen in RNN-based models by Koch et al. [25]. Moreover, Wu et al. [26] demonstrated that combining spectral and temporal features enhances recognition.

This advancement in sensor data acquisition has led to increased interest in wearable or handheld devices equipped with sensors like accelerometers and gyroscopes. DTW [27] effectively measures the similarity between temporal sequences, accounting for variations in speed and length, which is crucial for recognizing gestures with inconsistent execution times. Studies by Akl et al. [16] demonstrate the effectiveness of combining DTW and AP for gesture recognition using accelerometer data, achieving up to 98.7% accuracy in controlled conditions but dropping to 94.6% when generalized. However, one primary limitation is the difficulty in handling large gesture vocabularies [29]. The sensitivity of the system to sensor placement is another limitation. Sensor placement inconsistencies can affect gesture recognition accuracy by causing data variations [30], [31].

Many traditional machine-learning (ML) algorithms recognise sensor-based gestures. SVMs effectively model non-linear relationships and handle high-dimensional data [32]. Wu et al. [26] also found that a multi-class SVM can recognise 12 gestures with a 98.93% accuracy using accelerometer data. A study used Mechanomyogram (MMG) signals to recognize eight gestures, with KNN (k=7) achieving 94.56% accuracy [20]. NN outperformed ML models in gesture classification of 5 classes, according to Tajwar et al. [33]. However, Andrez et al. [17] found that RF handles complex gestures and outperformed SVM, NN, NB, and KNN in hand gesture recognition with 95.8% accuracy using accelerometer data. Static neural networks and LDA performed similarly for user-independent classification in [28]. In one study, TDNNs outperformed LDA, but LDA may be suitable for gesture recognition. Many machine learning models use manually extracted features from raw sensor data [21]. This feature engineering process takes time, domain expertise, and may not capture optimal gesture recognition

results. NNs outperform traditional gesture recognition methods by automatically learning features and modelling temporal dependencies. 1D CNNs work well for time-series sensor data—PairNet, a 1D CNN, achieved a 92.36% hit rate [18]. 2D CNNs process time-frequency representations, as in Faisal et al. [5], who reported F1-scores up to 1. RNNs also show promise, achieving 81.8% and 70.1% accuracy on DB2 and DB3 datasets, outperforming prior sEMG-based results [13]. Deep Belief Networks (DBNs) have shown 92.22% accuracy for large motions and 90.19% for subtle gestures [21]. Despite their accuracy, CNNs [34], [35] and RNNs [15] are computationally intensive. To address this, Tang et al. [36] introduced a triplet attention mechanism that enhances feature learning by capturing interactions across sensor, time, and channel dimensions, improving performance.

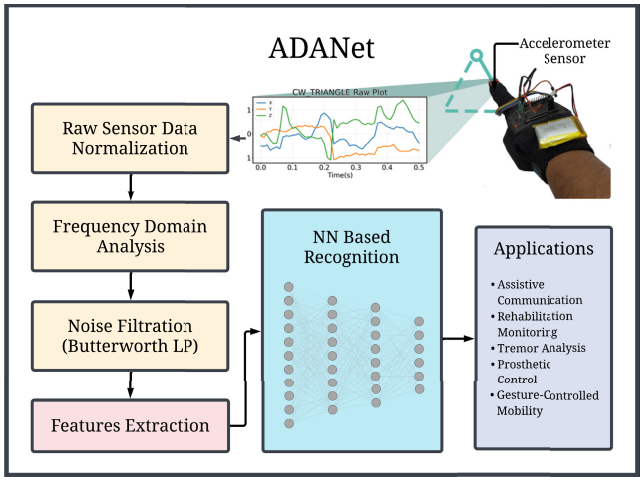


FIGURE 1. System pipeline for real-time gesture recognition using accelerometer data and neural network-based classification for assistive applications.

III. METHODOLOGY

In Figure 1, the gesture recognition system pipeline is shown along with the a) Sensor Data Acquisition System, (b) Signal Processing Pipeline, and (c) Neural Network Optimization.

A. HARDWARE CONFIGURATION AND SIGNAL ACQUISITION

The motion sensing architecture centers on the ADXL335, a compact tri-axial analog accelerometer (4 mm × 4 mm × 1.45 mm), capable of measuring both static and dynamic accelerations within a $\pm 3g$ range. Despite its analog nature and legacy design, the ADXL335 remains widely utilized for low-latency tilt and motion sensing due to its stable output and cost efficiency. Each axis produces a voltage centered around 1.65 V at $0g$, with sensitivity approximately 330 mV/g. However, the sensor's internal filter resistor ($R_{\text{FILT}} \approx 32\,\text{k}\Omega$), with a tolerance of $\pm 15\%$, introduces variability in bandwidth and effective signal response, making its behavior best approximated by a first-order system:

$$H(s) = \frac{V(s)}{A(s)} = \frac{K}{\tau s + 1} \tag{1}$$

where $A(s)$ is the true acceleration, $V(s)$ is the output voltage, τ is the sensor's time constant, and K is the axis sensitivity gain.

To digitize the analog signal, the ADXL335 was interfaced with an ESP32 microcontroller operating at 3.3 V. The ESP32 integrates a 12-bit ADC and dual-core Xtensa 32-bit LX6 processors with 520 KB SRAM, Wi-Fi, Bluetooth, and low-power support. The ADC samples analog signals at 100 Hz ($T_s = 10\,\text{ms}$), translating voltages V_{in} into digital values $v[k] \in [0, 4095]$ using:

$$v[k] = \mathcal{Q}\left(\frac{V_{\text{in}}(kT_s)}{V_{\text{ref}}} \cdot (2^n - 1)\right) \tag{2}$$

where $\mathcal{Q}(\cdot)$ denotes the quantization operator, V_{ref} is the reference voltage, and $n = 12$ corresponds to the ADC resolution.

In prior implementations, to compensate for ADC jitter and maintain buffer fill consistency, a *proportional-integral (PI) control loop* was introduced to dynamically adjust sampling intervals. This closed-loop correction system computes the timing error $e(k)$ as the difference between the expected buffer rate $r(k)$ and the observed fill level $b(k)$ at sample index k:

$$e(k) = r(k) - b(k) \tag{3}$$

The timing correction $\theta(k)$ applied to the sampling schedule is then updated as:

$$\theta(k) = K_P e(k) + K_I \sum_{i=0}^{k} e(i) \tag{4}$$

where K_P and K_I are proportional and integral gains, respectively. This control mechanism minimizes buffer underrun/overrun events by modulating sample timing based on real-time buffer behavior, thereby adapting to system latency and ensuring temporal fidelity of the captured signals.

However, such dynamic adjustment introduces complexity, particularly when deterministic and equidistant sampling is required, for instance, in real-time classification or time-frequency analysis tasks. To address this, the present system employs a *hardware timer-based sampling mechanism*, eliminating the need for feedback-based timing correction.

An internal hardware timer on the ESP32 is configured to generate precise interrupts at specific time itervals, thereby initiating analog-to-digital conversion at an exact sampling frequency. This fixed-rate, interrupt-driven architecture guarantees constant sampling intervals, independent of processor workload or software delays. As a result, the system naturally maintains buffer synchronization and timing integrity without requiring the compensatory PI loop.

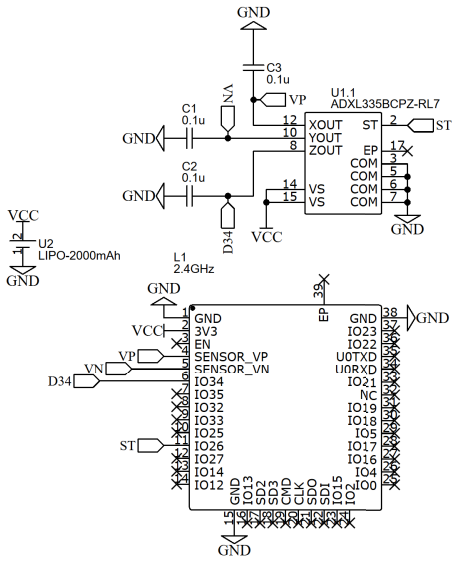


FIGURE 2. Schematic of interfacing ADXL335 accelerometer sensor with ESP32 for motion data acquisition.

The acquisition process is modeled as a discrete-time state-space system, now operating under *open-loop* sampling conditions:

$$x[k+1] = Ax[k] + Bu[k] + w[k] \tag{5}$$

$$y[k] = Cx[k] + v[k] \tag{6}$$

where $x[k]$ denotes the latent system state (e.g., acceleration, filtering effects), $u[k]$ denotes acquisition parameters (e.g., gain), and $w[k], v[k]$ represent process and measurement noise, respectively. With hardware-timed execution, the timing signal $\theta(k)$ is no longer required, as temporal alignment is intrinsically preserved.

This architectural simplification not only reduces computational overhead but also enhances determinism and reproducibility in downstream signal processing. The accelerometer was mounted on the fingertip of the index finger to exploit its superior motion range, thereby maximizing signal variability and richness for subsequent neural classification.

To accurately encode both the temporal dynamics and spectral intricacies inherent in gesture-induced inertial motion, tri-axial accelerometric data corresponding to the X (red), Y (green), and Z (blue) axes were acquired at a uniform sampling frequency of 400 Hz from 5 different participants during the execution of eight semantically distinct dynamic gestures. A time–frequency analysis of the Z-axis acceleration data, exemplified via Short-Time Fourier Transform (STFT)-based spectrogram visualization, reveals that the dominant energy concentration lies predominantly within the sub-50 Hz bandwidth across all gesture classes, with **a magnitude range of -15 to 0 dB**, shown in Figure 3. This spectral confinement underscores the quasi-stationary behavior of natural limb movements and confirms the absence of high-frequency transients in user-induced gestures.

According to the Nyquist–Shannon sampling criterion, the minimum admissible sampling frequency f_s must satisfy the inequality:

$$f_s \geq 2f_{\max}$$

where $f_{\max}$ denotes the highest significant frequency component embedded within the raw accelerometric signal. Given that $f_{\max} \approx 50\,\text{Hz}$ empirically bounds the spectral content of all observed gestures, the minimum theoretical sampling frequency required to preclude spectral aliasing is $f_s = 100Hz$.

B. DATASET PREPARATION

1) OVERVIEW

A dataset was collected from 21 healthy volunteers (12 men and 9 women), aged between 19 and 54 years (mean age: 31.2 ± 8.7 years) with no disabilities or any type of deformities. Participants represented a diverse group in terms of occupation and background, including university students, academic staff, and working professionals from both technical and non-technical fields. All participants were right-handed and reported no history of upper-limb neuromuscular disorders. Each volunteer contributed at least 20 examples per gesture, totaling 3,735 gesture instances. Written informed consent was obtained from all participants prior to data collection. Eight distinct motions—clockwise and counterclockwise circles, squares, and triangles, along with "S" and double tap—were selected for their clarity, memorability, and minimal accidental activation. Data was recorded via a USB-connected accelerometer at 100 Hz, with a baud rate of 115200. Participants, after giving consent and receiving instructions, performed gestures shown in Figure 4 within a 2-second window while a Python script captured and stored the data in CSV format. The procedure was repeated 20 times, focusing on eight specific gestures. Data was collected in eight separate sessions for each participant and stored in individual CSV files. Figure 5 illustrates the complete data collection process. The sensor was securely affixed to the fingertip, either by means of a glove or, when necessary, a Velcro tie to ensure stability. Each gesture was performed exclusively through wrist articulation, with the arm and body kept as still as possible to reduce motion-induced noise. To maintain consistency, gestures were limited to a duration of 2 seconds or 200 samples. As illustrated in Figure 4, each gesture followed a defined starting point and direction.

2) DATA CURATION

Despite the participants being instructed to complete their gestures within 200 samples, there were a few instances where the respondents needed extra time. Therefore, we reviewed all of that data to determine whether or not it complied with our methodology. Once more, some undesired data were present because of the laptop's Serial Communication Port's delay when using the recording device. We also pruned them.

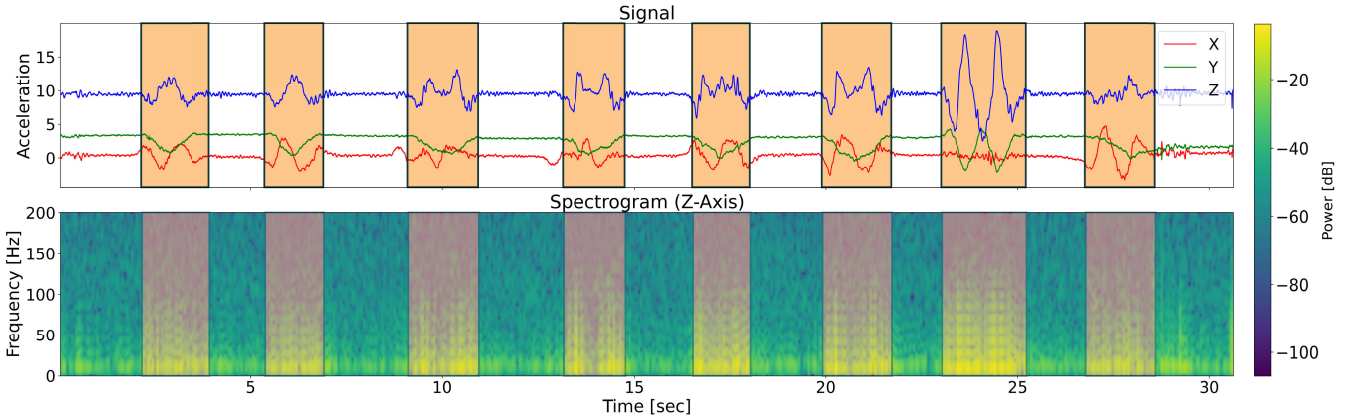


FIGURE 3. Accelerometer data (X: red, Y: green, Z: blue) for 8 distinct gestures (shaded) sampled at 400 Hz is shown above, with corresponding Z-axis spectrograms below. High-rate sampling enabled fine-grained spectral analysis to determine key frequency components. Based on the Nyquist criterion, a reduced sampling rate of ≥100 Hz was deemed sufficient to preserve gesture-relevant dynamics while optimizing system efficiency.

TABLE 2. Frequency content analysis of gesture signals from 5 participants for sampling rate justification.

Gesture Type	Participant	Dominant Frequency Range (Hz)	Frequency of Max Amplitude (Hz)	Remarks
CW Circle	P1–P5	0.23–34.45	9.4	Smooth cyclic motion with consistent tempo
CCW Circle	P1–P5	0.55–34.45	9.4	Similar to CW Circle, but in opposite direction
CW Square	P1–P5	3.01-31.33	10.2	Sharp corner turns cause increased spectral spread
CCW Square	P1–P5	3.96–43.86	11.6	Similar to CW Square, but with user-specific variation
CW Triangle	P1–P5	0.97-34.46	9.8	Less frequent than square; smoother transitions
CCW Triangle	P1–P5	1.29-40.73	9.4	Marginally lower spectral power than CW version
Double Tap	P1–P5	4.28–43.86	12.7	Highest frequency content due to impulsive nature
Letter 'S'	P1–P5	5.96–49.82	11.6	Moderate frequency content due to combined curves
Average across gestures	–	**2.53–34.45 Hz**	**10.51 Hz**	
Chosen Sampling Rate	–	–	**100 Hz**	Satisfies Nyquist criteria with high fidelity

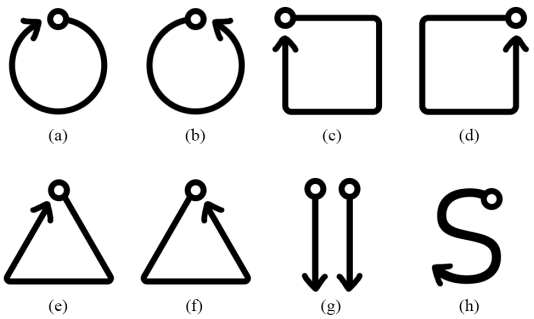


FIGURE 4. Gesture shapes: a) Clockwise circle, b) Counter-Clockwise circle, c) Clockwise square, d) Counter-Clockwise square, e) Clockwise triangle, f) Counter-Clockwise triangle, g) DoubleTap, h) S.

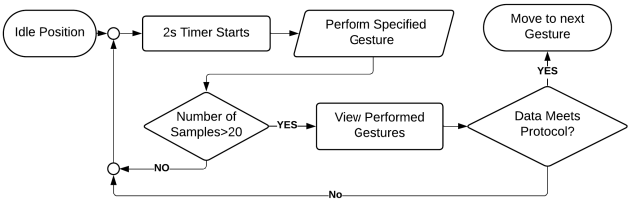


FIGURE 5. Gesture data acquisition protocol flowchart illustrating the looped process of timed gesture performance, validation against sampling and protocol standards, and progression through the gesture set.

3) NORMALIZATION

The raw accelerometer data acquired via the ESP32 microcontroller are 12 bit ADC values $r \in [0, 4095]$, representing analog voltages scaled to a reference voltage of $V_{\text{ref}} = 3.3\,\text{V}$. Under quiescent (0g) conditions, the accelerometer outputs a mid-scale voltage corresponding to $r = 2048$, i.e., $V_{\text{bias}} = \frac{V_{\text{ref}}}{2}$.

To convert these raw digital values into corresponding acceleration voltages, a linear mapping function was defined as:

$$V_r = \left(\frac{r[k] - 2048}{2048}\right) \cdot V_{\text{ref}} \tag{7}$$

This effectively normalizes the data into the voltage range $[-3.3, +3.3]\,\text{V}$, centered around 0V for 0g acceleration. The normalization accounts for the signed deviation from the bias and is critical for aligning multi-axis accelerometer data into a coherent time-series representation.

To further standardize the data across trials and subjects, a second-level statistical normalization was performed using a vectorized z-score transformation. Let $X = [x_{k,j}] \in \mathbb{R}^{N\times d}$ be the matrix of voltage-normalized samples, where $x_{k,j}$ is the value from the j^{th} axis at time k. Then:

$$z_{k,j} = \frac{x_{k,j} - \mu_j}{\sqrt{\sigma_j^2 + \varepsilon}}, \quad \forall j \in \{1, 2, \ldots, d\} \tag{8}$$

Here, $\varepsilon \approx 10^{-6}$ is a small numerical regularization factor added for stability.

$$\mu_j = \frac{1}{N}\sum_{k=1}^{N} x_{k,j} \tag{9}$$

$$\sigma_j^2 = \frac{1}{N-1}\sum_{k=1}^{N}(x_{k,j} - \mu_j)^2 \tag{10}$$

C. DATA FILTRATION

We initiated the signal filtering and processing pipeline, illustrated in Figure 6, by first visualizing the raw data to assess its time-domain characteristics and compare gesture patterns across different subjects shown in Figure 6a. Due to the accelerometer's high sensitivity and the sudden, jerky nature of gestures, high-frequency noise was introduced, distorting the signal. To mitigate this, we applied a simple moving average filter with window sizes of 5, 10, and 20.

Although a window size of 10 produced promising results, it still fell short of our expectations. To gain deeper insights, we converted the data from the time domain to the frequency domain using the Fast Fourier Transform (FFT), utilizing NumPy's FFT functionality in Figure 6b. FFT, an algorithm for computing the Discrete Fourier Transform (DFT) and its inverse, enabled us to analyze the signal's frequency components more effectively. The equation for DFT is given below:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j\frac{2\pi kn}{N}} \tag{11}$$

This allowed us to identify sinusoidal components within the signals. Additionally, we utilized PSD (Power Spectral Density) [37] in Figure 6c to assess the signal's power distribution across frequencies. The equation follows:

$$S_{xx}(f) = \frac{1}{N}|X(k)|^2 \tag{12}$$

Using this information, we aimed to approximate the cutoff frequency for filtering the signals in the frequency domain. However, this process proved challenging due to the large number of samples. To address this, we employed scatter plots to visualize FFT outputs for all samples of a specific gesture, enabling us to identify cluster ranges.

To suppress high-frequency noise while preserving essential gesture-related signals, a digital Butterworth low-pass filter was implemented due to its maximally flat magnitude response in the passband shown in Figure 6d. The analog prototype filter is governed by the squared magnitude transfer function:

$$|H(j\omega)|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}} \tag{13}$$

where n is the filter order and ω_c is the cutoff angular frequency. The design was transformed into a discrete-time filter using the bilinear transformation:

$$s = \frac{2}{T}\cdot\frac{1-z^{-1}}{1+z^{-1}} \tag{14}$$

Substituting into the analog transfer function yields the digital filter's Z-domain representation:

$$H(z) = \frac{\sum_{k=0}^{n} b_k z^{-k}}{1 + \sum_{k=1}^{n} a_k z^{-k}} \tag{15}$$

The filter coefficients $\{a_k, b_k\}$ were computed via the `scipy.signal.butter()` function. For implementation, we applied the recursive difference equation:

$$y[n] = \sum_{k=0}^{n} b_k x[n-k] - \sum_{k=1}^{n} a_k y[n-k] \tag{16}$$

where:

- $y[n]$ is the filtered signal at time step n,
- $x[n-k]$ is the input signal at time step $n-k$,
- a_k and b_k are the filter coefficients.

D. FEATURE EXTRACTION

1) MAGNITUDES

Jerk is defined as the rate at which acceleration changes over time. Our recorded data included acceleration measurements along the X, Y, and Z axes. From this, we calculated the total acceleration, the Jerk for each axis, and the overall total Jerk.

$$a_{tot}(t_i) = \|\mathbf{a}(t_i)\|_2 = \sqrt{a_x^2(t_i) + a_y^2(t_i) + a_z^2(t_i)} \tag{17}$$

$$j_x(t_i) = \left.\frac{da_x(t)}{dt}\right|_{t=t_i} \approx \frac{a_x(t_i) - a_x(t_{i-1})}{\Delta t} \tag{18}$$

$$j_y(t_i) = \left.\frac{da_y(t)}{dt}\right|_{t=t_i} \approx \frac{a_y(t_i) - a_y(t_{i-1})}{\Delta t} \tag{19}$$

$$j_z(t_i) = \left.\frac{da_z(t)}{dt}\right|_{t=t_i} \approx \frac{a_z(t_i) - a_z(t_{i-1})}{\Delta t} \tag{20}$$

$$j_{tot}(t_i) = \|\mathbf{j}(t_i)\|_2 = \sqrt{j_x^2(t_i) + j_y^2(t_i) + j_z^2(t_i)} \tag{21}$$

2) MIN & MAX

The minimum (Min), in Figure 7a represents the smallest value, and the maximum (Max), in Figure 7b is the largest value in a dataset. It serves as a vital feature, highlighting the lowest and peak values in data streams that may correspond to distinct characteristics of gestures.

3) RANGE

Range, in Figure 7c quantifies the spread or variability within a dataset, providing insight into its dispersion and potential outliers. It can also reflect the extent of motion or intensity variation in a gesture, aiding the model in distinguishing gestures based on their level of variability.

$$Range = Maximum - Minimum \tag{22}$$

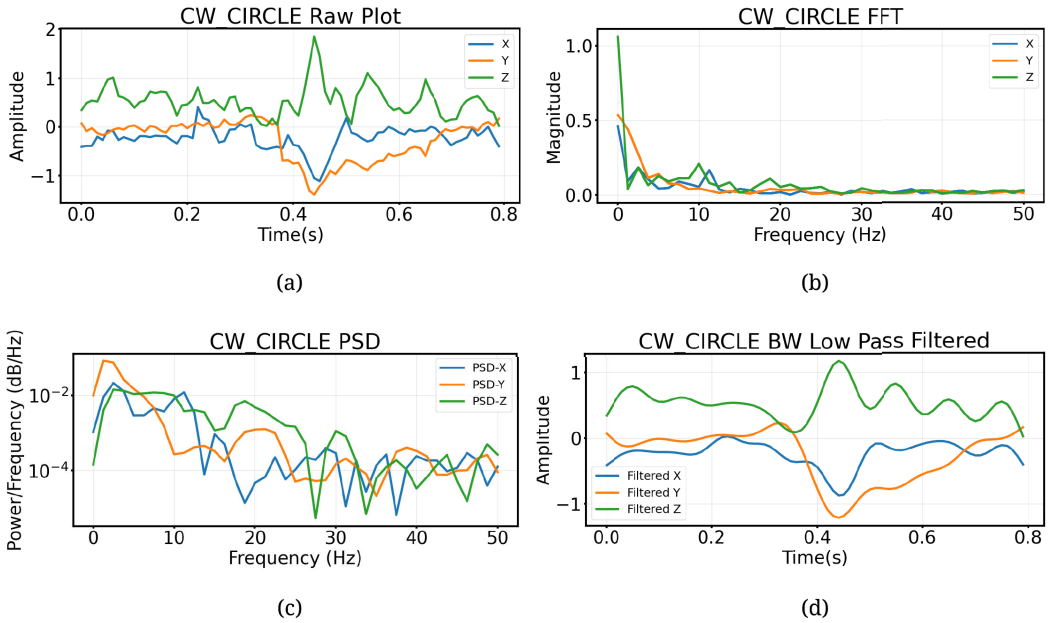


FIGURE 6. Signal processing steps: a) The gestures are plotted in the time domain; b) FFT is then applied on raw data to identify important contributing frequencies; c) Power content is measured using PSD across all frequencies; d) FFT & PSD determine the cutoff frequency, and the Butterworth LP filter generates the filtered signal.

4) MEAN & STANDARD DEVIATION (SD)

The mean, in Figure 7d offers a straightforward measure of a dataset's central tendency, facilitating data normalization and SD, in Figure 7e indicates how spread out a dataset is relative to its mean. A higher standard deviation reflects greater dispersion, as it measures the extent to which data points differ from the mean.

$$Mean = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{23}$$

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{24}$$

5) RMS

The root mean square (RMS), in Figure 7f is calculated and commonly used to evaluate the signal's energy and amplitude.

$$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} \tag{25}$$

6) ENTROPY

Entropy, in Figure 7g measures the level of uncertainty or randomness within a signal. It is calculated using the probability distribution of unique signal values. Based on Shannon's entropy, the formula quantifies the information content of the signal.

$$H(\mathbf{S}) = -\sum_{i=1}^{n} p_i \log(p_i) \tag{26}$$

Here, $H(\mathbf{S})$ is the entropy of the signal $\mathbf{S}$ and p_i is the probability of the i-th unique value in the signal.

7) SIMPLE MOVING AVERAGE (SMA)

The Simple Moving Average (SMA), in Figure 7h calculates the average of the absolute values of three input variables, x,y, and z. It is widely used in time series analysis to reduce short-term variations and emphasize longer-term trends.

$$\text{SMA}(t_i) = \frac{1}{N}\sum_{n=0}^{N-1} \|\mathbf{a}(t_{i-n})\|_1 \tag{27}$$

E. FEATURE ANALYSIS: CHI-SQUARED TEST

The Chi-Square (χ^2) test is a statistical test used to determine the relationship between categorical variables by comparing observed and expected frequencies calculated as:

$$\chi^2 = \sum \frac{(O-E)^2}{E} \tag{28}$$

The Chi-Square testing in Figure 8 showcased Y-jerk-rms and Y-jerk-std are the most significant features, showing the strongest correlation with the target variable. These features likely provide critical information for the model, emphasizing the importance of jerk-related metrics, particularly for the Y-axis. Following these, features like Y-jerk-range, Z-range, and Z-jerk-rms also score high, underscoring the relevance of variability and movement dynamics along the Z-axis. Conversely, features such as X-jerk-mean, total-jerk-mean, and Z-jerk-mean have much lower scores, indicating weaker or negligible contributions to the model.

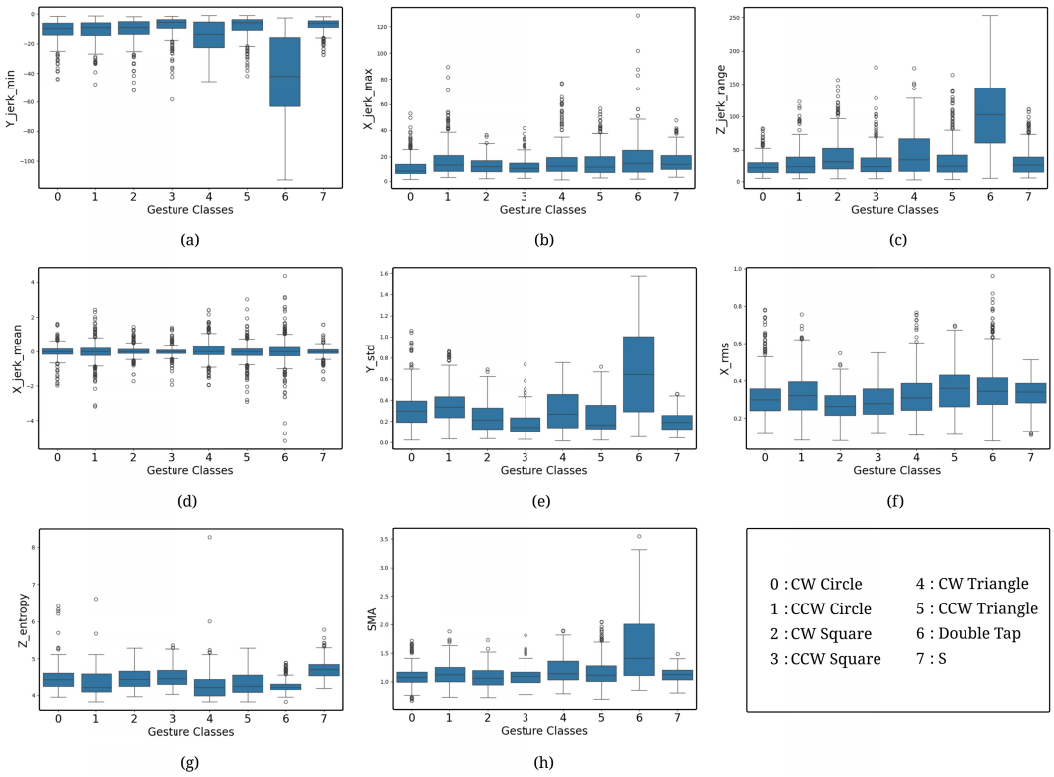


FIGURE 7. Distributions of statistical features extracted from accelerometer sensor data across eight gesture classes, labelled 0 to 7. Each subfigure represents a distinct feature: (a) Min, (b) Max, (c) Range, (d) Mean, (e) Standard deviation, (f) Root mean square, (g) Entropy, (h) Simple moving average.

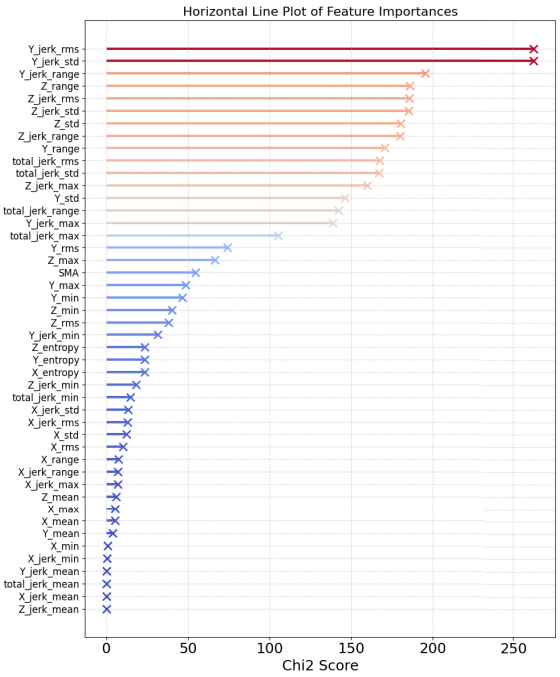


FIGURE 8. Chi-Squared feature ranking illustrating the discriminative power of each signal attributed in class-wise gestures.

IV. ADANet ARCHITECTURE AND TUNING

Following feature extraction, the dataset underwent meticulous preprocessing to prepare it for machine learning applications, as depicted in Figure 1. The data was initially partitioned into a 80:20 split for training and testing. To account for differences in feature scales and enhance training efficiency, each feature was normalized using Scikit-learn's PowerTransformer. PowerTransformer is particularly effective for transforming skewed or non-Gaussian distributed features into a more Gaussian-like distribution, which benefits many machine learning algorithms that assume normality. It stabilizes variance and minimizes the influence of outliers by applying a power-based transformation. Figure 9 presents the before and after power transformation distributions of "Y Jerk RMS".

The PowerTransformer supports both Yeo-Johnson and Box-Cox methods; in this study, the Yeo-Johnson transformation was used as it can handle both positive and negative values. The transformation is defined as:

$$x' = \begin{cases} \dfrac{(x+1)^{\lambda}-1}{\lambda} & \text{if } x \geq 0, \lambda \neq 0 \\ \log(x+1) & \text{if } x \geq 0, \lambda = 0 \\ \dfrac{-(-x+1)^{2-\lambda}+1}{2-\lambda} & \text{if } x < 0, \lambda \neq 2 \\ -\log(-x+1) & \text{if } x < 0, \lambda = 2 \end{cases} \tag{29}$$

Hyperparameter optimization was performed using GridSearchCV, enabling the identification of optimal model configurations for superior performance. We opt for fully connected (dense) architectures over convolutional neural networks (CNNs) for modeling temporal patterns in tri-axial

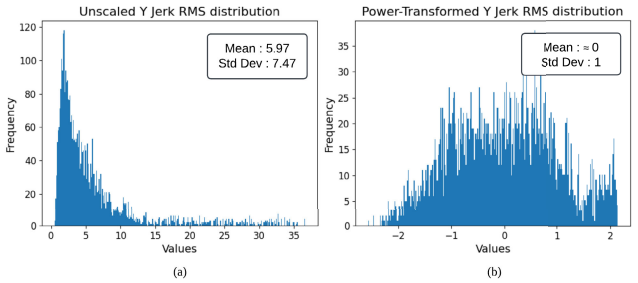


FIGURE 9. Two histograms display the distribution of Y Jerk RMS values. (a) The "Unscaled Y Jerk RMS distribution" shows raw, right-skewed data with a mean of 5.97 and a standard deviation of 7.47. (b) The "Power-Transformed Y Jerk RMS distribution" illustrates the data after a power transformation, yielding a more symmetrical, normalized distribution with a mean of 0 and a standard deviation of 1.

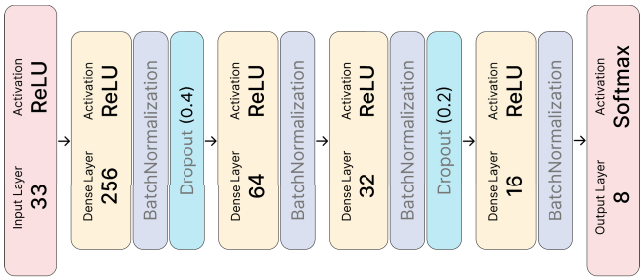


FIGURE 10. Architecture of ADANet comprising dense layers with ReLU activations, dropout, and batch normalization, designed to classify 33 input features into 8 gesture classes using an output layer.

inertial data. While convolutional layers are traditionally effective for grid-structured data such as images or spatially coherent signals, their efficacy diminishes when applied to low-dimensional, non-visual time-series inputs with limited spatial locality. Accelerometer data, being 1D and channel-wise segmented (X, Y, Z), lacks the hierarchical spatial dependencies that CNNs typically exploit. Applying convolutions on such signals can introduce unnecessary inductive biases (e.g., translation invariance), which are not inherently meaningful in gesture-induced sequences where signal shape and statistical transitions are more critical than localized spatial features.

In contrast, densely connected layers offer superior flexibility in learning cross-channel statistical correlations and abstract representations, particularly when paired with engineered features such as spectral entropy, RMS, and jerk derivatives. Dense layers directly map global feature interactions without the need for local receptive fields, making them well-suited for modeling the nonlinear temporal-spectral relationships present in hand gestures. Moreover, dense MLPs enable compact model footprints with tunable expressivity, which is crucial for deployment on edge devices with limited compute and memory. Our Ablation Study subsection further indicate that deeper convolutional models do not yield consistent gains and often underperform due to insufficient receptive field alignment with temporal segments and increased risk of overfitting in low-sample regimes.

The proposed neural network (NN) architecture, as depicted in Figure 10, is composed of multiple densely connected layers activated by the Rectified Linear Unit (ReLU) function. Batch Normalization and Dropout are integrated to regularize learning and reduce overfitting, thereby improving generalization to unseen data. The output layer employs a standard softmax activation to transform the logit vector $\mathbf{z}$ into class probabilities, expressed as:

$$P(y = i \mid \mathbf{z}) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{30}$$

where K denotes the number of gesture classes.

This ensures $P(y_i) \in (0, 1)$ and $\sum_i P(y_i) = 1$, making it probabilistically interpretable and well suited for use with cross-entropy loss. However, in cases involving noisy or ambiguous feature spaces common in real-time sensor data standard softmax can yield *overconfident* predictions that degrade model calibration. To mitigate this, we adopted a *temperature-scaled softmax* that introduces a positive scalar $\tau > 0$ to control the smoothness of the output distribution:

$$P(y_i; \tau) = \frac{e^{(z_i/\tau)}}{\sum_{j=1}^{C} e^{(z_j/\tau)}} \tag{31}$$

To optimize the proposed deep multilayer perceptron (MLP), we adopted the Adam optimizer, leveraging its first- and second-moment estimates for adaptive gradient scaling. Adam was particularly effective in handling the stochastic nature of accelerometer-derived gesture data. Its parameter update rule is defined as:

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{32}$$

where $\hat{m}_t$ and $\hat{v}_t$ are the bias-corrected first and second moment estimates of the gradient, respectively, and ϵ is a small constant for numerical stability.

To enhance convergence and prevent overfitting, an exponential learning rate decay schedule was applied, where the effective learning rate at epoch t is calculated as:

$$\eta_t = \eta_0 \cdot e^{-\lambda \cdot t} \tag{33}$$

Here, η_0 is the initial learning rate and λ is the decay rate to gradually reduce the step size and allow fine-tuning in later stages. Hyperparameters such as dropout rates, learning rate bounds, and hidden layer dimensionality were fine-tuned using a GridSearchCV approach over the validation set, thereby avoiding manual selection biases. This configuration yielded a robust model that balanced precision and efficiency while remaining lightweight and deployable on resource-constrained edge devices. To further evaluate the model's consistency and generalization ability, **K-fold cross-validation** was employed. This method partitions the training data into K disjoint subsets (folds), using $K-1$ folds for training and the remaining fold for validation, iteratively

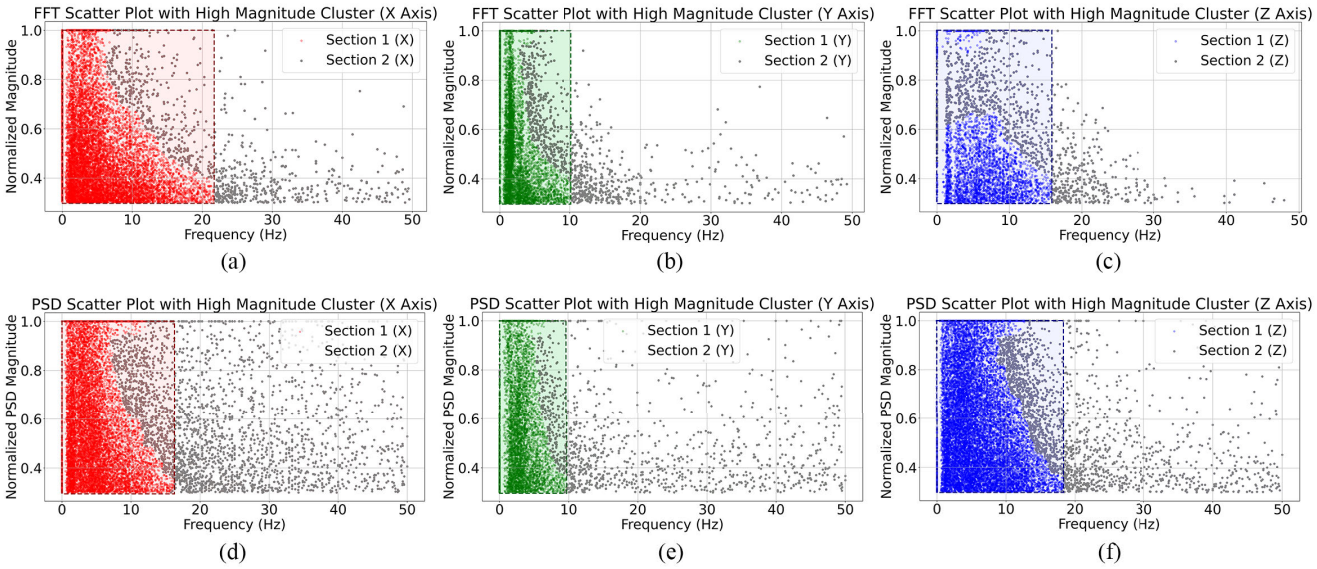


FIGURE 11. **Frequency domain scatter plots showing density-based clusters in accelerometer sensor data. (a, b, c): FFT scatter plots for X, Y, Z axes highlighting magnitude clusters across frequencies. (d, e, f): PSD scatter plots for X, Y, Z axes identifying key frequency ranges with significant power. Signal filtration frequency range is suggested by these clusters.**

across all combinations. The average performance metrics and their standard deviation over all folds provide a reliable estimate of model stability across different data splits. In this study, 10-fold cross-validation was used to measure how well the model generalizes to unseen data and to detect potential overfitting or variance issues.

To further verify its performance, the proposed NN was benchmarked against both classical machine learning classifiers and alternative deep learning architectures. These comparative results, elaborated in the subsequent sections, highlight the proposed model's superior precision, robustness, and real-time applicability in embedded gesture recognition scenarios. The grid search tested different combinations of settings, summarized in Table 3, provided insights into the model's optimal architecture and training parameters.

TABLE 3. **Optimized parameter for the ADANet model.**

Parameter	Best Choice
Dense Layers	$256 \rightarrow 64 \rightarrow 32 \rightarrow 16$
Activations	ReLU
Dropout Rates	$0.4 \rightarrow 0 \rightarrow 0.2$
Output Activation	Softmax with Temperature
Optimizer	Adam
Loss Function	Sparse Categorical Cross-entropy
Batch Size	64
Learning Rate	0.01
Decay Rate	0.96
β_1	0.9
β_2	0.999
ϵ	1×10^{-7}
Epochs	800
K-Fold	10

V. RESULTS & DISCUSSION

Figure 1 illustrates the gesture recognition system, which begins with data acquisition from an accelerometer-equipped wearable device and ESP32 microcontroller. We generate meaningful features from sensor signals using entropy, SMA, and RMS for precise classification. Finally, we will evaluate the compact model's gesture recognition accuracy, ROC curves, and confusion matrices.

A. SIGNAL PROCESSING

The FFT and PSD scatter plots for the X, Y, and Z axes were analyzed using density-based clustering to identify regions of high signal intensity. Section I, in Figure 11, spanning the frequency range of 0–20 Hz, consistently exhibited a higher density of normalized magnitudes across all axes, signifying that the dominant energy is concentrated within this range. Section II, covering 20–50 Hz, demonstrated sparse and irregular clusters, indicative of less significant high-frequency components or noise.

Axes-specific frequency behaviours were found after further analysis. Figures 11a,11d show a prominent amplitude up to 20 Hz for the X-axis, while Figures 11b,11e show a significant drop in signal intensity beyond 10 Hz for the Y-axis. However, the Z-axis in Figures 11c,11f showed significant amplitudes up to 15 Hz. These observations determined a cutoff frequency that balanced signal preservation across all axes. To determine the final cutoff frequency, we examined a Butterworth low-pass filter's frequency response at 10 Hz, 15 Hz, and 20 Hz. The findings are:

- A 20 Hz cutoff with a 5th-order filter left residual noise, especially in the Y-axis.

- Cutoff frequencies of 10 Hz and 15 Hz performed similarly in noise reduction while preserving key signals.
- A 10 Hz cutoff with a 4th-order Butterworth filter was chosen, effectively suppressing high-frequency noise while retaining essential Y-axis signal components. The transfer function for the filter can be written as

$$H(z) = \frac{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}}{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}} \quad (34)$$

This rigorous analysis, integrating spectral insights and filter design, ensured signal clarity by retaining key low-frequency features and reducing noise across all axes.

B. NN MODEL EVALUATION CRITERIA

We used important measurements like accuracy, precision, recall, and F1 score, along with data from the confusion matrix, ROC curves, and Gini coefficients, to evaluate how well the gesture recognition system is performing. These metrics provide a clear way to assess how well the proposed Neural Network (NN) and other models perform.

$$\text{Accuracy} = \frac{\sum_{i=1}^{n} \delta(y_i = \hat{y}_i)}{n} \quad (35)$$

$$\text{Precision} = \frac{\sum_{i=1}^{n} \delta(y_i = \hat{y}_i \wedge \hat{y}_i = 1)}{\sum_{i=1}^{n} \delta(\hat{y}_i = 1)} \quad (36)$$

$$\text{Recall/TPR} = \frac{\sum_{i=1}^{n} \delta(y_i = \hat{y}_i \wedge y_i = 1)}{\sum_{i=1}^{n} \delta(y_i = 1)} \quad (37)$$

$$\text{FPR} = \frac{\sum_{i=1}^{n} \delta(\hat{y}_i = 1 \wedge y_i = 0)}{\sum_{i=1}^{n} \delta(y_i = 0)} \quad (38)$$

$$F1\ \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (39)$$

$$\text{AUC} \approx \sum_{i=1}^{n-1} (\text{FPR}_{i+1} - \text{FPR}_i) \cdot \frac{\text{TPR}_{i+1} + \text{TPR}_i}{2} \quad (40)$$

where:

- y_i: The true label for the i^{th} sample.
- $\hat{y}_i$: The predicted label for the i^{th} sample.
- δ(condition): Indicator function that returns 1 if the condition is true, otherwise 0.
- n: Total number of samples.
- TPR: True Positive Rate.
- FPR: False Positive Rate.

C. PROPOSED MODEL ANALYSIS

The proposed neural network architecture consists of 29.3K trainable parameters, resulting in a compact model size of 114.72 KB. The model underwent training for 800 epochs with a batch size of 64. Optimization was performed using the Adam optimizer, with careful tuning of key hyperparameters, including the initial learning rate (α), first moment coefficient (β_1), and second moment coefficient (β_2). The final hyperparameter configuration was set as follows: $\alpha = 0.01$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$, ensuring a balance between convergence speed and stability throughout the training process. Furthermore, the softmax temperature parameter was introduced to calibrate the output confidence, specifically enhancing the separation between ambiguous gesture classes. After empirical optimization, the temperature was set to $\tau = 2.0$, which effectively smoothened the probability distribution, thereby improving generalization capabilities. The model achieved a mean cross validation accuracy of 81.19% and a test accuracy of 84.47%, demonstrating its viability for deployment on memory-constrained embedded and edge devices.

Table 4 presents the confusion matrix, offering a granular view of the system's performance across eight gesture classes. The model achieved macro-averaged precision, recall, and F1-scores of 0.84, 0.84, and 0.84, respectively. While gestures like "Double Tap", and "S" achieved near-perfect scores (F1: 0.97 and 0.80; Gini: 1.00 and 0.99), others such as "CW Triangle", "CCW Triangle" and "CW Triangle" had relatively lower precision, indicating potential feature overlap or class imbalance. These inconsistencies highlight the need for refined class distribution strategies, including augmentation, synthetic sampling, or class-weighted loss adjustments to boost underperforming classes.

The training dynamics are captured in Figures 12a and 12b, where the model exhibits a stable convergence in loss (∼0.25) and an accuracy plateau around 90%. The validation curve closely trails the training curve, indicating low overfitting due to effective regularization via dropout layers and batch normalization. This balance signifies robust generalization across unseen data, a crucial attribute for real-time and user-independent gesture recognition.

Figure 12c displays the class-wise ROC curves. High AUC values for classes like "Double Tap" (1.00) and "S" (0.99) reaffirm the model's discriminative power. The Gini coefficients (ranging from 0.93 to 1.00) further validate its class separation capability. Gestures with lower Gini (e.g., "CCW Square": 0.93) may benefit from improved preprocessing techniques or integrating attention-based modules to better capture temporal-spatial dependencies.

D. ABLATION STUDY

To systematically evaluate the effects of network architecture and regularization on generalization performance, we conducted extensive ablation experiments across eight multilayer perceptron (MLP) configurations (A1–A8), as summarized

TABLE 4. Confusion matrix showing precision, sensitivity (RECALL) and error rates for each recognized gesture.

		Predictions								%Sensitivity
		CW Circle	CCW Circle	CW Square	CCW Square	CW Triangle	CCW Triangle	Double Tap	S	%Error
Targets	CW Circle	**68** 9.10%	**8** 1.07%	**3** 0.40%	**0** 0.00%	**1** 0.13%	**3** 0.40%	**0** 0.00%	**4** 0.54%	**78.16%** 21.84%
	CCW Circle	**6** 0.80%	**67** 8.97%	**4** 0.54%	**0** 0.00%	**3** 0.40%	**2** 0.27%	**1** 0.13%	**0** 0.00%	**80.72%** 19.28%
	CW Square	**1** 0.13%	**5** 0.67%	**77** 10.31%	**8** 1.07%	**6** 0.80%	**0** 0.00%	**2** 0.27%	**1** 0.13%	**77.00%** 23.00%
	CCW Square	**1** 0.13%	**2** 0.27%	**6** 0.80%	**57** 7.63%	**2** 0.27%	**2** 0.27%	**1** 0.13%	**2** 0.27%	**78.08%** 21.92%
	CW Triangle	**1** 0.13%	**2** 0.27%	**3** 0.40%	**7** 0.94%	**98** 13.12%	**4** 0.54%	**0** 0.00%	**3** 0.40%	**83.05%** 16.95%
	CCW Triangle	**0** 0.00%	**2** 0.27%	**0** 0.00%	**3** 0.40%	**6** 0.80%	**66** 8.84%	**1** 0.13%	**2** 0.27%	**82.50%** 17.50%
	Double Tap	**0** 0.00%	**0** 0.00%	**1** 0.13%	**0** 0.00%	**0** 0.00%	**0** 0.00%	**114** 15.26%	**0** 0.00%	**99.13%** 0.87%
	S	**2** 0.27%	**0** 0.00%	**1** 0.13%	**0** 0.00%	**1** 0.13%	**3** 0.40%	**0** 0.00%	**84** 11.24%	**92.31%** 7.69%
	%Precision	**86.08%**	**77.91%**	**81.05%**	**76.00%**	**83.76%**	**82.50%**	**95.80%**	**87.50%**	
	%Error	13.92%	22.09%	18.95%	24.00%	16.24%	17.50%	4.20%	12.50%	

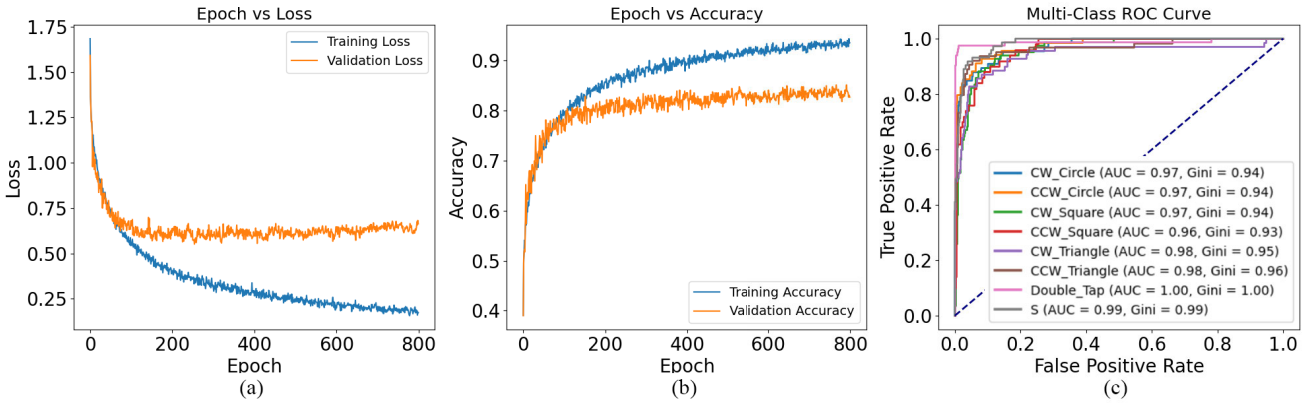


FIGURE 12. Model performance visualization: (a) The epoch vs. loss plot reflects effective convergence, with minimal overfitting observed. (b) Accuracy trends show consistent training and validation improvement, plateauing around epoch 800. (c) The Multi-Class ROC curve highlights strong class-wise separability, with all classes achieving AUC≥ 0.96 and "Double Tap" attaining highest discrimination (AUC = 1.00), underscoring robust generalization.

in Table V. All models were trained under identical data preprocessing pipelines, learning rate schedules, and random seeds to ensure fairness and reproducibility.

Our investigation began with a shallow 3-layer baseline (A1: [64-32-16]) incorporating both dropout and batch normalization (BN). While efficient in terms of parameters (15.7K), the model achieved limited test accuracy (75.04%), indicating insufficient representational capacity for the complexity of the input gesture sequences.

To address this, we progressively expanded the hidden dimensions. Configuration A2 increased the first layer to 256 neurons while keeping subsequent layers narrow. This yielded a 3.93% gain in test accuracy, highlighting the importance of high-dimensional early representations in capturing diverse gesture dynamics. However, removing dropout in A2 led to a slight increase in overfitting ($\Delta = 0.20$), suggesting that larger models benefit from regularization, even under stable training regimes. Overfitting was quantitatively estimated as the absolute difference between training and test accuracies after convergence. This metric reflects the generalization gap, with higher values indicating poorer robustness to unseen data. All reported values are computed post-800 epochs to ensure consistency across architectures. Further refinements explored deeper compositions. A3 and A4 introduced an additional 64-unit layer, while A5 extended the architecture to include a 32-unit bottleneck. A5, equipped with both dropout and BN, demonstrated the best overall performance, achieving 79.86% test accuracy and an F1-score of 0.7938, while maintaining a compact 85.56K parameter count. This

TABLE 5. Quantitative summary of ablation results.

Arch. ID	Architecture	Dropout	BN	#Params	Acc. (%)	Precision	Recall	F1	△ Overfit
A1	[-64–32–16-]	✓	✓	15.70K	75.04	0.7438	0.7450	0.7450	0.18
A2	[-256-32–16-]	✗	✓	55.20K	78.97	0.7825	0.7838	0.7938	0.20
A3	[-256-64-16-]	✓	✗	78.99K	77.01	0.7638	0.7625	0.7600	0.15
A4	[-256-64–32-]	✗	✗	82.49K	76.11	0.7550	0.7563	0.7538	0.22
A5	[-256-64–32-]	✓	✓	85.56K	79.86	0.8013	0.7900	0.7938	0.17
A6	[-256-64–32–16-]	✗	✗	83.70K	74.69	0.7463	0.7413	0.7413	0.21
A7	[-256-64–32–16-]	✓	✗	83.70K	78.07	0.7725	0.7738	0.7725	0.19
A8	[-256-64–32–16-]	✗	✓	86.63K	78.25	0.7738	0.7800	0.7763	0.19

indicates that the combination of wide front-end layers [256-64-32], regularization, and normalization jointly contributes to robust feature abstraction and improved generalization.

Adding a fourth hidden layer of 16 units in A6–A8 tested the effect of marginally increasing depth. Without regularization (A6), the model showed degraded performance (74.69%) and a notable rise in overfit ($\Delta = 0.21$). Introducing dropout (A7) or BN (A8) partially mitigated this, restoring accuracy to $\sim$ 78%. Nonetheless, the incremental performance gains came at the cost of additional parameters ($\sim$ 86%), offering diminishing returns compared to the leaner A5 configuration. Overall, the study reveals that architectures with front-loaded width (particularly in the 256–64–32 pattern), combined with dropout and batch normalization, achieve optimal performance. Deeper or overly wide models tend to overfit or underperform without proportionate benefit.

E. COMPARATIVE ANALYSIS

The performance of various models presented in Table 6 was comprehensively evaluated to benchmark their effectiveness against the proposed ADANet architecture. The comparative analysis underscores ADANet's superior generalization capability and balanced performance, which outclass traditional machine learning models and even several deep learning counterparts. ADANet achieved the highest testing accuracy of **84.5%**, along with consistent precision (**0.843**), recall (**0.847**), and F1-score (**0.847**), indicating robust classification performance. Its cross-validation accuracy of **81.2%** with a low standard deviation (**0.018**) further validates its reliability and stability across data folds.

In contrast, models such as Random Forest (RF), XGBoost, and Gradient Boosting (GB) showed lower testing accuracies (73.7%, 74.1%, and 68.9%, respectively), with higher parameter counts (up to **122K**) and significantly larger model sizes—for instance, **15.35 MB** for RF and **1.63 MB** for XGBoost—compared to ADANet's compact **114.72 KB** footprint. Though RF and LGBM had moderately high cross-validation scores (74.1% and 76.7%, respectively), their generalization performance remained inferior, reflected in F1-scores ranging between **0.736** and **0.752**. Similarly, Histogram GB and LGBM offered mid-tier performance, but lacked the deep representational power needed for fine-grained gesture distinctions. Among simpler classifiers, SVM, KNN, and LDA demonstrated limited effectiveness, with testing accuracies of **64.3%**, **65.7%**, and **50.7%**, respectively. Their F1-scores also remained low, particularly for LDA (0.505), which struggled to model complex data patterns. While KNN exhibited relatively fast inference (**0.005 ms**), it came at the cost of lower precision and higher model size (**1.02 MB**), failing to achieve the balance attained by ADANet.

Even more complex architectures, such as the Transformer model and the deep learning approach by Renqian et al., achieved testing accuracies of **77.8%** and **79.5%**, respectively, still falling short of ADANet's performance. Despite having massive parameter counts (e.g., **230K** for Transformer) and larger model sizes, these models incurred significantly higher inference times (e.g., **0.809 ms**) and exhibited weaker F1-scores (around **0.771–0.791**), limiting their practical deployment in real-time, embedded environments. The model proposed by Fahad et al. performed the pooerst second to LDA, with only **56.1%** testing accuracy and an F1-score of **0.546**, reaffirming the need for more advanced architectures tailored for gesture-based recognition in noisy, high-dimensional sensor data. **ADANet offers the most compelling trade-off** between accuracy, efficiency, and deployability. Its **modest parameter size (29.4K)**, **small storage footprint (114.72 KB)**, and **low average inference time (0.086 ms)** make it highly suitable for real-time, embedded applications such as wearable gesture recognition. Its consistent performance across metrics confirms ADANet as a state-of-the-art solution for robust, scalable, and resource-efficient sensor-based classification tasks. One of the most widely used approaches in gesture recognition is vision-based modeling. However, state-of-the-art vision-based systems are often computationally intensive and exhibit degraded performance under occlusions and varying lighting conditions. In contrast, the data collection strategy employed in ADANet, based on inertial sensing offers a more lightweight, robust, and occlusion invariant alternative, making it better suited for real-time, embedded applications.

To complement the performance metrics and ensure the robustness of our comparative claims, we conducted statistical significance testing using the Friedman test followed by the Nemenyi post-hoc test. These tests assess whether

TABLE 6. Model comparison.

Model	Testing Accuracy	Precision	Recall	F1-Score	Cross-Val Accuracy	Standard Deviation	Parameters	Model Size(KB)	Avg. Inference Time(ms)
Fahad *et al.* [28]	0.561	0.541	0.543	0.546	0.539	0.021	475	1.85	0.142
Renqian *et al.* [13]	0.795	0.792	0.796	0.791	0.775	0.024	50184	196.03	0.136
RF [38]	0.737	0.738	0.737	0.736	0.741	0.019	122510	15351.21	0.013
GB [39]	0.689	0.689	0.689	0.687	0.704	0.013	11756	1008.54	0.01
LGBM[40]	0.753	0.754	0.753	0.752	0.767	0.021	48800	2719.43	0.019
XGBoost [41]	0.741	0.742	0.741	0.740	0.751	0.011	473	1630.96	0.003
Histogram GB[42]	0.717	0.714	0.717	0.715	0.771	0.021	475	1564.87	0.027
SVM [32]	0.643	0.641	0.643	0.639	0.676	0.021	2541	945.22	0.027
KNN [43]	0.657	0.668	0.657	0.659	0.715	0.019	3174	1017.38	0.005
LDA [20]	0.507	0.533	0.507	0.505	0.592	0.025	640	9.04	0.003
Transformer [44]	0.778	0.773	0.774	0.771	0.751	0.021	230664	901.03	0.809
Proposed ADANet	**0.845**	**0.843**	**0.847**	**0.847**	**0.812**	**0.018**	**29,368**	**114.72**	**0.086**

TABLE 7. Nemenyi post-hoc test p-values comparing each model against ADANet.

Model	p-Value
Fahad *et al.*	0.0000000102
Renqian *et al.*	0.937104
Random Forest	0.071057
Gradient Boosting	0.0000624
LightGBM	0.88696
XGBoost	0.1984079
HistGradientBoosting	0.9298103
SVM (RBF)	0.000000798
KNN	0.00087544
LDA	0.00000000233
Transformer	0.204571
ADANet (Reference)	**1.00000**

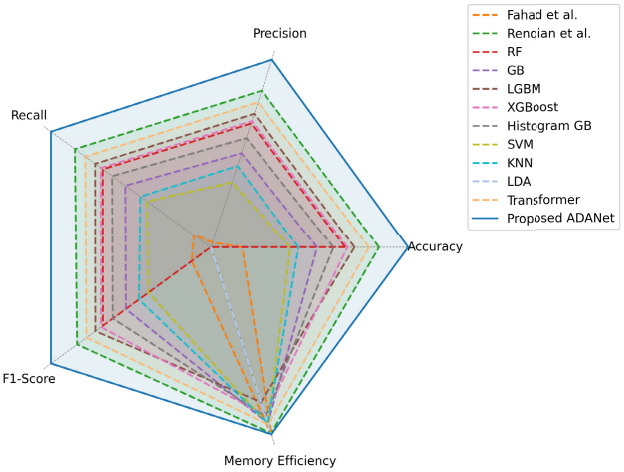


FIGURE 13. Radar chart comparing the proposed ADANet with benchmark models across precision, recall, accuracy, F1-Score, and memory efficiency. ADANet (solid blue) outperforms all others in every metric, highlighting its robustness, accuracy, and suitability for resource-constrained edge-based gesture recognition.

the observed differences among models are statistically meaningful across the 10-fold cross-validation results. The null hypothesis of the Friedman test states that *"all models perform equally"*, while the alternative hypothesis posits that *"at least one model performs significantly differently"*. Our analysis yielded a Friedman chi-square statistic of 75.5689 with a p-value < 0.00000, leading us to reject the null hypothesis at the 5% significance level.

To identify where these differences lie, we employed the Nemenyi post-hoc test using ADANet as the control method. The null hypothesis here assumes *"no significant performance difference between model pairs"*, while the alternative hypothesis indicates *"a significant difference"*. The Nemenyi test revealed that ADANet significantly outperformed several models like Fahad et al., Gradient Boosting, SVM, KNN, and LDA, with p-values shown in Table 7.

Other models, including Renqian et al., Random Forest, LightGBM, XGBoost, Histogram GB, and Transformer, did not show statistically significant differences when compared with ADANet. These findings statistically reinforce the empirical results shown in Table 6, validating ADANet's superiority in both predictive performance and practical deployability across a wide range of traditional and deep learning models.

The radar chart shown in Figure 13 underscores the comprehensive superiority of the proposed ADANet across all evaluated dimensions—precision, recall, accuracy, F1-score, and memory efficiency surpassing both conventional classifiers and recent state-of-the-art models. Where traditional methods often depend on shallow architectures and standard statistical features that fall short in handling gesture subtleties or overlapping motion profiles, ADANet distinguishes itself through a tightly integrated deep learning framework and instrumentation-aware signal processing pipeline. The feature set, enriched with entropy-based descriptors and jerk-informed temporal dynamics, enables nuanced discrimination of complex and transitional gestures. These features are extracted via a carefully calibrated acquisition process that ensures temporal consistency, noise resilience, and

cross-user generalizability. Importantly, ADANet is engineered with embedded deployment in mind, targeting platforms like the ESP32-S3, which provides up to 16 MB PSRAM and 512 KB of SRAM. Given the model's compact footprint, it fits comfortably within these constraints while preserving performance. The use of batch normalization and dropout across dense layers ensures both training stability and robustness against overfitting capabilities rarely attainable in resource-constrained deployments. This deliberate balance of computational efficiency, architectural depth, and signal discriminative power positions ADANet as a highly effective and scalable solution for real-time, on-device gesture recognition.

VI. CONCLUSION

This study demonstrates the efficacy of neural network–based models in accurately recognizing human gestures from accelerometer data, significantly outperforming conventional machine learning approaches in both precision and robustness. The proposed architecture leverages layered dense connectivity, batch normalization, dropout regularization, and optimized learning schedules to achieve strong generalization across varying user profiles and gesture complexities. From a measurement perspective, the system offers reliable real-time classification of nuanced motion signals, establishing its potential for integration into embedded platforms and wearable systems. The research contributes a practical framework for low-latency, high-accuracy gesture recognition, enabling deployment in applications where precision and responsiveness are critical. One of the most impactful implications of this work lies in the domain of disability assistance. When embedded within bionic limbs, the proposed gesture recognition mechanism could enable intuitive control of prosthetic devices, bridging the gap between user intention and mechanical actuation. This capability supports more natural and dexterous movements, promoting improved autonomy and quality of life for individuals with motor impairments. Additionally, the findings provide a foundational basis for future instrumentation-driven research, particularly in human–robot interaction and biomedical monitoring. Future extensions may incorporate multimodal sensor fusion, such as electromyography (EMG) or electroencephalography (EEG), to further elevate accuracy and contextual understanding. By advancing the reliability of sensor-based measurement systems, this work sets the stage for smarter, adaptive assistive technologies with meaningful real-world impact. The core gesture recognition framework holds strong promise for expansion into AR/VR and mixed reality ecosystems. By enabling precise and low-latency interpretation of hand and body movements, it could enhance natural user interaction, immersive control schemes, and contextual responsiveness in virtual environments. Such integration would facilitate more intuitive human–computer interfaces (HCI), redefine interactive experiences, and pave the way for richer applications in training, simulation, remote collaboration, and immersive accessibility tools.

ACKNOWLEDGMENT

The authors thankfully acknowledge the IoT & Robotics Research Laboratory at CARS, University of Dhaka, for providing support on advanced computational resources.

DATA AVAILABILITY STATEMENT

The custom dataset, source code for model training, and evaluation are available in IEEE Dataport. The repository DOI is 10.21227/2cs0-j680, and the access link is provided below: ADANet GestureMotion

REFERENCES

[1] H. Li, J. Liu, Z. Yang, R. W. Liu, K. Wu, and Y. Wan, "Adaptively constrained dynamic time warping for time series classification and clustering," *Inf. Sci.*, vol. 534, pp. 97–116, Sep. 2020.

[2] B. Mor, S. Garhwal, and A. Kumar, "A systematic review of hidden Markov models and their applications," *Arch. Comput. Methods Eng.*, vol. 28, no. 3, pp. 1429–1448, May 2020.

[3] J. S. Dramsch, M. Lüthje, and A. N. Christensen, "Complex-valued neural networks for machine learning on non-stationary physical data," *Comput. Geosci.*, vol. 146, Jan. 2021, Art. no. 104643.

[4] K. Dermitzakis, A. H. Arieta, and R. Pfeifer, "Gesture recognition in upper-limb prosthetics: A viability study using dynamic time warping and gyroscopes," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2011, pp. 4530–4533.

[5] M. A. A. Faisal, F. F. Abir, M. U. Ahmed, and M. A. R. Ahad, "Exploiting domain transformation and deep learning for hand gesture recognition using a low-cost dataglove," *Sci. Rep.*, vol. 12, no. 1, p. 21446, Dec. 2022.

[6] M. Kim, J. Cho, S. Lee, and Y. Jung, "IMU sensor-based hand gesture recognition for human–machine interfaces," *Sensors*, vol. 19, no. 18, p. 3827, Sep. 2019.

[7] J. Molina, J. A. Pajuelo, and J. M. Martínez, "Real-time motion-based hand gestures recognition from time-of-flight video," *J. Signal Process. Syst.*, vol. 86, no. 1, pp. 17–25, Jan. 2017.

[8] A. Wadhawan and P. Kumar, "Deep learning-based sign language recognition system for static signs," *Neural Comput. Appl.*, vol. 32, no. 12, pp. 7957–7968, Jun. 2020.

[9] E. Stergiopoulou, K. Sgouropoulos, N. Nikolaou, N. Papamarkos, and N. Mitianoudis, "Real time hand detection in a complex background," *Eng. Appl. Artif. Intell.*, vol. 35, pp. 54–70, Oct. 2014.

[10] G. Devineau, F. Moutarde, W. Xi, and J. Yang, "Deep learning for hand gesture recognition on skeletal data," in *Proc. 13th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May 2018, pp. 106–113.

[11] W. Aly, S. Aly, and S. Almotairi, "User-independent American sign language alphabet recognition based on depth image and PCANet features," *IEEE Access*, vol. 7, pp. 123138–123150, 2019.

[12] Z. Zhang, "Microsoft Kinect sensor and its effect," *IEEE Multimedia Mag.*, vol. 19, no. 2, pp. 4–10, Feb. 2012.

[13] R. Xie and J. Cao, "Accelerometer-based hand gesture recognition by neural network and similarity matching," *IEEE Sensors J.*, vol. 16, no. 11, pp. 4537–4545, Jun. 2016.

[14] L. Roda-Sanchez, C. Garrido-Hidalgo, A. S. García, T. Olivares, and A. Fernández-Caballero, "Comparison of RGB-D and IMU-based gesture recognition for human-robot interaction in remanufacturing," *Int. J. Adv. Manuf. Technol.*, vol. 124, no. 9, pp. 3099–3111, Feb. 2023.

[15] A. Akl and S. Valaee, "Accelerometer-based gesture recognition via dynamic-time warping, affinity propagation, & compressive sensing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2010, pp. 2270–2273.

[16] A. Akl, C. Feng, and S. Valaee, "A novel accelerometer-based gesture recognition system," *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 6197–6205, Dec. 2011.

[17] D. O. Anderez, L. P. Dos Santos, A. Lotfi, and S. W. Yahaya, "Accelerometer-based hand gesture recognition for human–robot interaction," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2019, pp. 1402–1406.

[18] Y.-C. Chu, Y.-J. Jhang, T.-M. Tai, and W.-J. Hwang, "Recognition of hand gesture sequences by accelerometers and gyroscopes," *Appl. Sci.*, vol. 10, no. 18, p. 6507, Sep. 2020.

[19] L. Qin, L. Zhang, C. Li, C. Song, D. Cheng, S. Wang, H. Wu, and A. Song, "Towards better accuracy-efficiency trade-offs: Dynamic activity inference via collaborative learning from various width-resolution configurations," *IEEE Trans. Artif. Intell.*, vol. 5, no. 12, pp. 6723–6738, Dec. 2024.

[20] M.-K. Liu, Y.-T. Lin, Z.-W. Qiu, C.-K. Kuo, and C.-K. Wu, "Hand gesture recognition by a MMG-based wearable device," *IEEE Sensors J.*, vol. 20, no. 24, pp. 14703–14712, Dec. 2020.

[21] T.-Y. Pan, W.-L. Tsai, C.-Y. Chang, C.-W. Yeh, and M.-C. Hu, "A hierarchical hand gesture recognition framework for sports referee training-based EMG and accelerometer sensors," *IEEE Trans. Cybern.*, vol. 52, no. 5, pp. 3172–3183, May 2022.

[22] P. Casale, O. Pujol, and P. Radeva, "Human activity recognition from accelerometer data using a wearable device," in *Pattern Recognition and Image Analysis*, J. Vitriá, J. M. Sanches, and M. Hernández, Eds., Berlin, Germany: Springer, 2011, pp. 289–296.

[23] C. Bu, L. Zhang, H. Cui, G. Yang, and H. Wu, "Dynamic inference via localizing semantic intervals in sensor data for budget-tunable activity recognition," *IEEE Trans. Ind. Informat.*, vol. 20, no. 3, pp. 3801–3813, Mar. 2024.

[24] T. Pylvänäinen, "Accelerometer based gesture recognition using continuous HMMs," in *Pattern Recognition and Image Analysis*, J. S. Marques, N. Pérez de la Blanca, and P. Pina, Eds., Berlin, Germany: Springer, 2005, pp. 639–646.

[25] P. Koch, M. Dreier, M. Maass, M. Böhme, H. Phan, and A. Mertins, "A recurrent neural network for hand gesture recognition based on accelerometer data," in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2019, pp. 5088–5091.

[26] J. Wu, G. Pan, D. Zhang, G. Qi, and S. Li, "Gesture recognition with a 3-D accelerometer," in *Ubiquitous Intelligence and Computing*, D. Zhang, M. Portmann, A.-H. Tan, and J. Indulska, Eds., Berlin, Germany: Springer, 2009, pp. 25–38.

[27] J. Liu, Z. Wang, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "UWave: Accelerometer-based personalized gesture recognition and its applications," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, Mar. 2009, pp. 1–9.

[28] F. Moiz, P. Natoo, R. Derakhshani, and W. D. Leon-Salas, "A comparative study of classification methods for gesture recognition using a 3-axis accelerometer," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2011, pp. 2479–2486.

[29] J. Kela, P. Korpipää, J. Mäntyjärvi, S. Kallio, G. Savino, L. Jozzo, and S. D. Marca, "Accelerometer-based gesture control for a design environment," *Pers. Ubiquitous Comput.*, vol. 10, no. 5, pp. 285–299, Aug. 2006.

[30] M. H. M. Noor, Z. Salcic, and K. I.-K. Wang, "Adaptive sliding window segmentation for physical activity recognition using a single tri-axial accelerometer," *Pervas. Mobile Comput.*, vol. 38, pp. 41–59, Jul. 2017.

[31] I. Hussain, R. Jany, R. Boyer, A. Azad, S. A. Alyami, S. J. Park, M. M. Hasan, and M. A. Hossain, "An explainable EEG-based human activity recognition model using machine-learning approach and LIME," *Sensors*, vol. 23, no. 17, p. 7452, Aug. 2023.

[32] M. A. Hearst, S. Dumais, E. Osuna, J. Platt, and B. Schölkopf, "Support vector machines," *IEEE Intell. Syst. Their Appl.*, vol. 13, no. 4, pp. 18–28, Jul. 1998.

[33] A. Tajwar, M. E. Haque, and M. M. Hasan, "Development of an accelerometer-based data acquisition system for hand gesture recognition," in *Proc. IEEE 3rd Int. Conf. Robot., Autom., Artif.-Intell. Internet Things (RAAICON)*, Nov. 2024, pp. 13–18.

[34] M. M. Hasan, J. P. Chopin, H. Laga, and S. J. Miklavcic, "Correction to: Detection and analysis of wheat spikes using convolutional neural networks," *Plant Methods*, vol. 15, no. 1, p. 27, Mar. 2019.

[35] M. A. Batin, M. Islam, M. M. Hasan, A. Azad, S. A. Alyami, M. A. Hossain, and S. J. Miklavcic, "WheatSpikeNet: An improved wheat spike segmentation model for accurate estimation from field imaging," *Frontiers Plant Sci.*, vol. 14, Aug. 2023, Art. no. 1226190.

[36] Y. Tang, L. Zhang, Q. Teng, F. Min, and A. Song, "Triple cross-domain attention on human activity recognition using wearable sensors," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 6, no. 5, pp. 1167–1176, Oct. 2022.

[37] R. N. Youngworth, B. Gallagher, and B. L. Stamper, "An overview of power spectral density (PSD) calculations," *Proc. SPIE*, vol. 5869, Aug. 2005, Art. no. 58690U.

[38] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[39] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[40] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, Dec. 2017, pp. 3149–3157.

[41] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," 2016, *arXiv:1603.02754*.

[42] J. Cui, H. Hang, Y. Wang, and Z. Lin, "GBHT: Gradient boosting histogram transform for density estimation," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 2233–2243.

[43] B. W. Silverman and M. C. Jones, "E. Fix and J.L. Hodges (1951): An important contribution to nonparametric discriminant analysis and density estimation: Commentary on fix and Hodges (1951)," *Int. Stat. Rev./Revue Internationale de Statistique*, vol. 57, no. 3, p. 233, Dec. 1989.

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*.



MD ETTASHAMUL HAQUE is currently a Research Assistant at the AVIS Research Laboratory, University of Dhaka, Bangladesh. He is working on human–robot interaction and bio-inspired prosthetic systems, focusing on developing adaptive control strategies for enhanced functionality and user experience. His research interests include signal processing, robot vision, embedded systems, and autonomous robotics systems, with an emphasis on optimizing deep learning models.



ATIQUE TAJWAR is currently a Research Assistant at the AVIS Research Laboratory, University of Dhaka, Bangladesh, contributing to a project on prosthetic hand development for disability support. His expertise lies in signal processing, robotics, and autonomous system design. With a strong background in electronics and the IoT, he leverages machine learning, deep learning, and embedded systems to enhance control strategies.

AKM AZAD was a Cancer Data Scientist and Bioinformatician in multiple research institutes in Australia and South Korea, i.e., the Children's Medical Research Institute, Australia, the Kolling Institute of Medical Research, Australia, the University of New South Wales, Australia, and Gwangju Institute of Science and Technology, South Korea. He is currently a Cancer Data Scientist and an AI Scientist by research. He is appointed as a Faculty Member at the Faculty of Science, Imam Mohammad Ibn Saud Islamic University. He has an excellent research career with strong and highly focused track record on the development of advanced methodologies and computational resources for multi-omics (e.g., Transcriptomics, Proteomics, Genomics, etc.) data analysis and their relevant drug positioning. During his research career, he has gained experiences in multi-omics data for identifying network-level biomarkers and analyzing their drug-targets in various cancers, e.g., Glioblastoma, Breast, Pancreas, etc. His current research interests include AI-based biomedicine, data science, Bayesian methods, Bayesian networks, and drug repositioning.



SALEM A. ALYAMI (Member, IEEE) received the Ph.D. degree in bio-statistics from Monash University, Australia, in 2017. He has been contributing and leading several grants in Statistics and as the Dean of Scientific Research with IMSIU, since 2020. He is currently an Associate Professor with the School of Mathematics and Statistics, Faculty of Science, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia. His research interests include statistical inference, Bayesian statistics, sampling algorithms, statistical simulation, and the applications of statistics in biology and medicine.



MD MEHEDI HASAN received the Ph.D. degree in computer science from the University of New South Wales, Australia, in 2017. He joined as a Postdoctoral Researcher at UNSW. Afterward, he was at the Phenomics and Bioinformatics Research Centre, University of South Australia, as a Postdoctoral Researcher to conduct applied research on aerial and land-based imaging platforms with various geospatial sensors. Later on, he leads as a Machine Learning Engineer (Research and Development) at RME Australia to provide AI-driven industry 4.0 automation systems for mill relining machines. He is presently employed as a Faculty Member with the Department of Robotics and Mechatronics Engineering, University of Dhaka, Bangladesh. Earlier, he was an Associate Professor and the Research Director of SURC at Stamford University Bangladesh. His area of research interests include plant phenomics, robotics visual servoing, the IoT, 3D video coding, machine vision, and deep learning. He is a reviewer of several distinguished journals, including *Briefings in Bioinformatics*, *Plant Methods*, *Frontiers in Plant Science*, *Functional Plant Biology*, IEEE Access, etc.