International Journal of Technology

http://ijtech.eng.ui.ac.id

*Research Article*

# Enhancing Human Activity Recognition (HAR) via Semi-Adaptive Sliding Windows with Triplet Subwindow Similarity and Growth Capping

**Made Liandana[1*], Gede Angga Pradipta[2], Putu Desiana Wulaning Ayu[3], Dandy Pramana Hostiadi[2]**

[1]Department of Informatics and Computer, Institut Teknologi dan Bisnis STIKOM Bali, Denpasar,80232, Indonesia

[2]Department of Magister Information Systems, Institut Teknologi dan Bisnis STIKOM Bali, Denpasar,80232, Indonesia

[3]Information Technology Department, Politeknik Negeri Bali, Badung, 80225, Indonesia

[*]Corresponding author: liandana@stikom-bali.ac.id; Tel.: +6281-819-1670-0229; Fax: +6236-126-4773

**Abstract:** In recent years, sensor-based human activity recognition has become an active research topic. Sensor data are typically represented as time series, which require a segmentation process before feature extraction and machine learning–based classification. The sliding window is one of the most commonly used segmentation techniques; however, determining the optimal window length remains a major challenge for achieving accurate activity recognition performance. This study proposes a semi-adaptive sliding window method that integrates static and dynamic strategies using accelerometer sensor data. The proposed approach exploits temporal information by considering the current, past, and future windows, each of which is further divided into three sub-windows. The window size is adaptively updated based on the similarity among pairs of subwindows forming a triplet subwindow, and a growth-capping mechanism is incorporated to prevent excessive window expansion. Performance evaluation was conducted using the XGBoost and LightGBM classifiers on the FORTH-TRACE, SBHARPT, WISDM, and PAMA2 datasets. The experimental results show that using XGBoost and LightGBM, the proposed method achieves accuracies of 97.26% and 97.26% on the FORTH-TRACE dataset, 98.09% and 98.15% on the SBHARPT dataset, 98.97% and 99.06% on the WISDM dataset, and 92.21% and 92.43% on the PAMA2 dataset, respectively. These results demonstrate that the proposed semi-adaptive sliding window approach consistently improves human activity recognition performance.

**Keywords:** Accelerometer; Activity-recognition; Semi-adaptive; Similarity; Subwindow

## 1. Introduction

Wearable devices come in various forms, such as wristbands (Mekruksavanich and Jitpattanakul, 2022), bracelets (Machado et al., 2022), smart clothing (Petz et al., 2021), and antenna-based designs (Rahayu et al., 2021). These devices are widely used for various applications, one of the most prominent being Human Activity Recognition (HAR). HAR can be applied across various domains, including healthcare. Examples of applications in the healthcare sector include elderly monitoring (Dilshad Ansari et al., 2024; Jeon et al., 2023), patient rehabilitation monitoring (Chandramouli et al., 2024), sports activity monitoring (Host and Ivašić-Kos, 2022), and Parkinson's disease monitoring (Chen and Lin, 2024), as well as other physical activity monitoring. Recognition of human activity can significantly enhance the quality of life (Guerra et al., 2023; Wang et al., 2025).

Human activity recognition can be achieved using two main types of data: image and sensor data (Minh Dang et al., 2020; Rustam et al., 2020). Recognizing activities through sensor data

https://doi.org/10.14716/ijtech.v17i3.8295
Received December 2025; Revised March 2026; Accepted April 2026

offers benefits in terms of computational efficiency, as this type of data is typically lighter and does not necessitate complex hardware, such as image or video data (Liandana et al., 2024). Sensor data, which include inputs from accelerometers, gyroscopes, and magnetometers, are typically serial or time-series data. Therefore, specific techniques or methods must effectively process these data for accurate activity recognition (Liandana et al., 2025). Recognizing human activities involves data collection, preprocessing, and training of machine learning models (Vidya and Sasikumar, 2022). Segmentation is a crucial sub-step in the preprocessing phase. The sliding window method is a segmentation technique used to handle time-series data (Hirawat et al., 2022; Ma et al., 2020). Data from the sliding window are then converted into relevant features, either through handcrafted feature extraction techniques or deep learning–based approaches (Baraka and Mohd Noor, 2023; Wang et al., 2020). Subsequently, the extracted features were classified using machine learning models.

Both the sliding window method and feature extraction significantly impact the effectiveness of machine learning classification (Liandana et al., 2025; Qian et al., 2021). Consequently, the sliding window method is essential for improving machine learning outcomes. There are two main strategies for implementing sliding windows: adaptive (or dinamic) and static sliding windows (Jaén-Vargas et al., 2022; Sun et al., 2022). Adaptive sliding windows modify the window size in response to data changes, removing the need for manual window size specification; however, this can lead to excessively large window sizes (Sun et al., 2022). Large windows can complicate processing. However, static sliding windows require the window size to be manually set, which involves determining the optimal size (Jaén-Vargas et al., 2022). Additionally, too wide windows may capture several activities at once, whereas too small windows may lead to incomplete activity representation (Dilshad Ansari et al., 2024; Ma et al., 2020). Thus, it can be concluded that the adaptive and static approaches each have their own advantages and limitations. The adaptive approach does not require the manual specification of the window size, as it can automatically adjust the window length based on the data characteristics. However, this approach may result in excessively large window sizes that can capture multiple activities within a single window (Ma et al., 2020). In contrast, the static approach provides better control over the window size because it is predefined, thereby reducing the risk of overly large windows. Nevertheless, this approach requires the initial determination of an optimal window size, which is not always straightforward and highly dependent on the characteristics of the data used.

Therefore, a semi-adaptive sliding window approach is required because it combines the advantages of both methods by enabling window size adjustment while maintaining segmentation stability and producing a more accurate activity representation.

## 2. Related Works

In the field of sensor-based Human Activity Recognition (HAR), raw data obtained from accelerometer and gyroscope sensors are continuous time-series streams (Hirawat et al., 2022; Lu et al., 2025; Sun et al., 2022). Data streams must be segmented into smaller and more representative portions to enable classification models to process raw data (Jaén-Vargas et al., 2022; Sun et al., 2022). The sliding window method is the primary technique employed for segmentation, which operates by shifting a window of a specified size to create segments (Hirawat et al., 2022; Pan et al., 2025). Subsequently, each segment becomes a data unit from which features are extracted and utilized to train machine learning or deep learning models for activity recognition (Hirawat et al., 2022; Jaén-Vargas et al., 2022; Pan et al., 2025). As a critical preprocessing step, window size selection is of great importance because it directly affects accuracy, latency, and computational cost (Hirawat et al., 2022; Jaén-Vargas et al., 2022). Sliding window techniques are categorized into static and dynamic approaches. Static sliding windows maintain a fixed window and step sizes, representing the most fundamental and widely used method (Hirawat et al., 2022; Jaén-Vargas et al., 2022) . In contrast, dynamic or adaptive sliding windows employ variable window sizes that dynamically adjust based on specific criteria derived from the signal data (Hirawat et al., 2022). This approach aims to allow the window size to align with the

natural duration of each activity (Baraka and Mohd Noor, 2023).

The static sliding window proposed in earlier research uses similarity, referred to as the Similarity Segmentation Approach (SSA) (Baraka and Mohd Noor, 2023). This technique exploits the activity signal's temporal structure during segmentation. This approach divides each window into sub-windows and extracts internal features by measuring the window and its sub-windows' similarity. These internal features are also used to measure the dissimilarity between adjacent windows. Consequently, the proposed method can distinguish between transition and non-transition windows, thereby producing more effective classification outcomes. Experiments on two public datasets demonstrated that this approach achieved 97.65% accuracy in distinguishing Transition Activities (TAs) from Basic Activities (BAs), improving TA recognition accuracy by 33.41%. Overall, activity recognition accuracies of 92.71% and 86.65% were obtained for the two datasets (Baraka and Mohd Noor, 2023). Previous studies have also evaluated the sizes of static windows, ranging from very short windows (5, 10, 15, 20, and 25 frames) to longer windows (50, 75, 100, and 200 frames). This study compared the effects of different window sizes on two types of acceleration data sources: Inertial Measurement Unit (IMU) sensors and Motion Capture Systems (MOCAP). These data were classified using deep learning models, including deep neural networks (DNN), convolutional neural networks (CNN), long short-term memory networks (LSTM), and hybrid CNN-LSTM models. Larger windows (100 or 200 frames) produced almost no difference in accuracy, whereas smaller windows resulted in a decrease in F1-scores (Jaén-Vargas et al., 2022). A sliding window was also applied to segment continuous multi-channel sensor data into fixed-length sequences as part of the preprocessing stage before classification using the MTSDNet deep learning architecture. A key innovation of MTSDNet is the normalization applied within each sliding window layer. This process constrains the feature range and allows the model to learn generalized low-rank representations, thereby improving its generalizability across different participants (Pan et al., 2025). A deep multifeature extraction framework based on an attention mechanism (DMEFAM) model was introduced to address several common challenges in HAR, such as incomplete feature extraction and low feature utilization, which may lead to recognition errors. For preprocessing, the DMEFAM employs a static sliding window with 50% overlap. This overlap ensured the continuity of the time-series data across the windows and enhanced the recognition accuracy. The window size was adapted according to the dataset, with window periods of 4 s and 2.56 s applied in their experiments (Wang et al., 2023b). In addition, other studies have implemented static sliding windows with a 25% overlap ratio. The experiments were conducted with fixed window sizes of 60, 90, and 120 samples to determine the optimal configuration. Excessively large windows may encompass multiple activities, whereas negligible windows may fragment a single activity (Liandana et al., 2025).

In addition to static approaches, dynamic sliding window methods have also been explored. One example is the Dynamic-sized Non-overlapping sliding window spanning the entire activity duration (DyNOEA), which serves as an alternative to conventional static sliding windows with fixed sizes. The primary objective of DyNOEA is to detect the boundaries between activities by identifying when one activity ends and another begins. The algorithm scans the sensor data streams and monitors the activity labels in each record. When a label change is detected (e.g., from "IN" to "SITTING"), the algorithm marks the point as the end of the previous activity and the start of the new activity. However, this method is only effective when accurate labels are available at the segmentation stage, limiting its applicability for real-time use (Hirawat et al., 2022). The dynamic sliding window-based human activity recognition (DSWHAR) method defines window sizes based on the duration of individual activity occurrences rather than the overall activity span. This method emphasizes the activities' periodic characteristics. For instance, instead of defining a window for the entire walking activity, a window is defined using the duration of a single step. The objective is to prevent excessively large windows when an activity has a long duration. During the training phase, the model recognizes the activities and determines the optimal window size for each periodic activity. For each type of periodic

activity in the dataset (e.g., walking, jogging, cycling), the method analyzes accelerometer signals using Fast Fourier Transform (FFT) to identify the dominant frequency, which is then converted into a time period representing a single occurrence of the activity. During recognition, DSWHAR dynamically selects the most appropriate window size to classify incoming sensor data in real-time. Although effective, this method is relatively complex because it requires FFT-based period extraction, candidate window creation, multiple classifier training, and a dynamic selection mechanism during inference (Sun et al., 2022).

A substantial body of research has demonstrated that both static and dynamic sliding window methods play essential roles in segmenting sensor-based time-series data for HAR. Static methods are straightforward and widely adopted; however, their performance is highly dependent on the window size selection. In contrast, dynamic methods offer greater flexibility, particularly for periodic activities, but may lead to excessively large windows. Generally, attention-based approaches involve higher computational complexity. This study aims to bridge this gap by proposing a controlled window expansion mechanism that prevents excessive growth of window sizes to address these limitations. Windows representing the same activity tended to exhibit similar patterns, indicating that similar windows could be expanded while appropriate constraints were maintained. The relationships between the windows were evaluated using similarity metrics, including the Manhattan, Euclidean, and Minkowski distances. These metrics were used to reduce computational complexity while effectively capturing the relationships between similarity. From a temporal perspective, the similarity is evaluated across the past, present, and future windows. Windows with higher similarity levels are considered more suitable for expansion, whereas those with lower similarity levels do not require further enlargement. The similarity degree is determined using a voting mechanism. With this approach, window expansion is not performed statically; instead, the static configuration is applied only at the initial stage, referred to as the base window.

The main contributions of this study are as follows: (1) the development of a semi-adaptive sliding window framework as a segmentation formulation that integrates the stability of static approaches with the adaptability of dynamic methods within a controlled mechanism; (2) the incorporation of a three-directional temporal context (past–current–future) to address the limitation of dynamic windowing methods that typically rely on local signal changes; (3) the introduction of a triplet sub-window similarity mechanism that enables adaptive window size adjustment based on inter-window relationships; and (4) the implementation of a growth-capping mechanism as an explicit control over window expansion, which is not available in conventional dynamic methods, thereby preventing over-segmentation or the merging of multiple activities within a single segment.

## 3. Methods

The proposed model, illustrated in Figure 1, consists of two main stages: (1) the semi-adaptive sliding window process and (2) the training and testing process of the machine learning models. In this study, the sensor data were exclusively obtained from accelerometer sensors; therefore, only accelerometer data from the selected datasets were considered. This choice was made to reduce the data dimensionality and computational complexity. Subsections 3.1 and 3.2 provide a more detailed description of each stage of the proposed model.

### 3.1 Semi-Adaptive Sliding Window Process

The semi-adaptive sliding window stage consists of several sequential steps, including computing the signal vector magnitude from the accelerometer sensor, applying overlapping sliding windows, determining triplet windows, extracting basic features from each subwindow, computing the similarity between pairs of subwindows, calculating the average similarity value, performing similarity-based voting, and updating the window size accordingly.

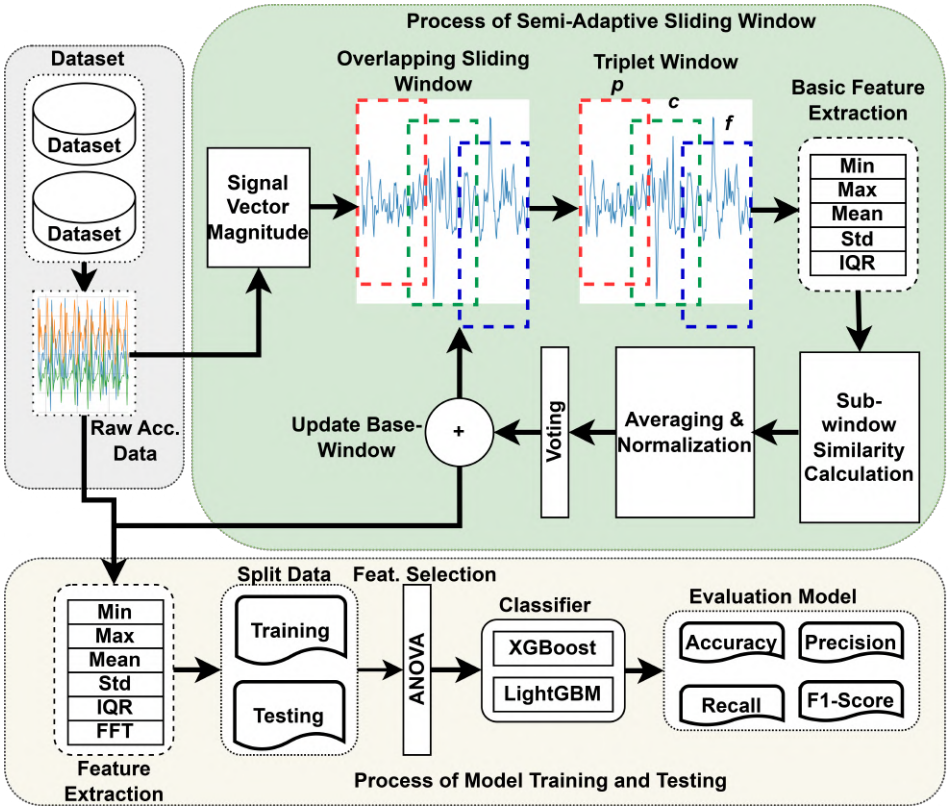



**Figure 1** Semi-Adaptive Sliding Window Process

### 3.1.1 Signal Vector Magnitude (SVA)

An accelerometer sensor consists of three axes: x, y, and z, each representing different directions and acceleration values. To represent these three axes with a single value, they were combined using the Signal Vector Magnitude (SVA). The window size determination process does not require the analysis of each axis separately by aggregating the axis values through SVA, thereby simplifying the analysis. The Signal Vector Magnitude (SVM) is computed using Equation (1). This formulation has been widely adopted to reduce the amount of triaxial sensor data, particularly for accelerometer sensors (Liandana et al., 2024).

$$SVA_k = \sqrt{A_x(k)^2 + A_y(k)^2 + A_z(k)^2} \tag{1}$$

where SVA denotes the signal vector magnitude. $A_x$, $A_y$, and $A_z$ represent the acceleration values along the x-, y-, and z-axes, respectively, of the accelerometer sensor. The index $k=1,2,\ldots,n$ indicates the sample number, where $n$ denotes the total number of samples.

### 3.1.2 Overlapping sliding window and triplet window

In previous studies, the window length was evaluated using widths of 60, 90, and 120 samples, with optimal performance achieved using a window length of 120 samples (Baraka and Mohd Noor, 2023; Liandana et al., 2025). Therefore, this study adopted the same window length. The sequence of the signal vector magnitude values computed using Equation (1) was initially segmented using a window length of 120 samples, and subsequent segments were generated with a 50% overlap. This overlap ratio was determined based on previous studies (Wang et al., 2023b).

Figure 2 illustrates the proposed overlapping and triplet window schemes as the basis for determining the semi-adaptive sliding window. Three windows are considered: the past window ($W_(t-1)$), the current window ($W_t$), and the future window ($W_(t+1)$), which are collectively referred to as the triplet window. Each window is further divided into three sub-windows, which

are denoted as follows:

- $W_{t-1}^{(j)}$ represents the *j-th* sub-window of the previous window,
- $W_t^{(j)}$ represents the j-th subwindow of the current window, and
- $W_{t+1}^{(j)}$ represents the j-th sub-window of the future window, where $j \in (1, 2, 3)$.

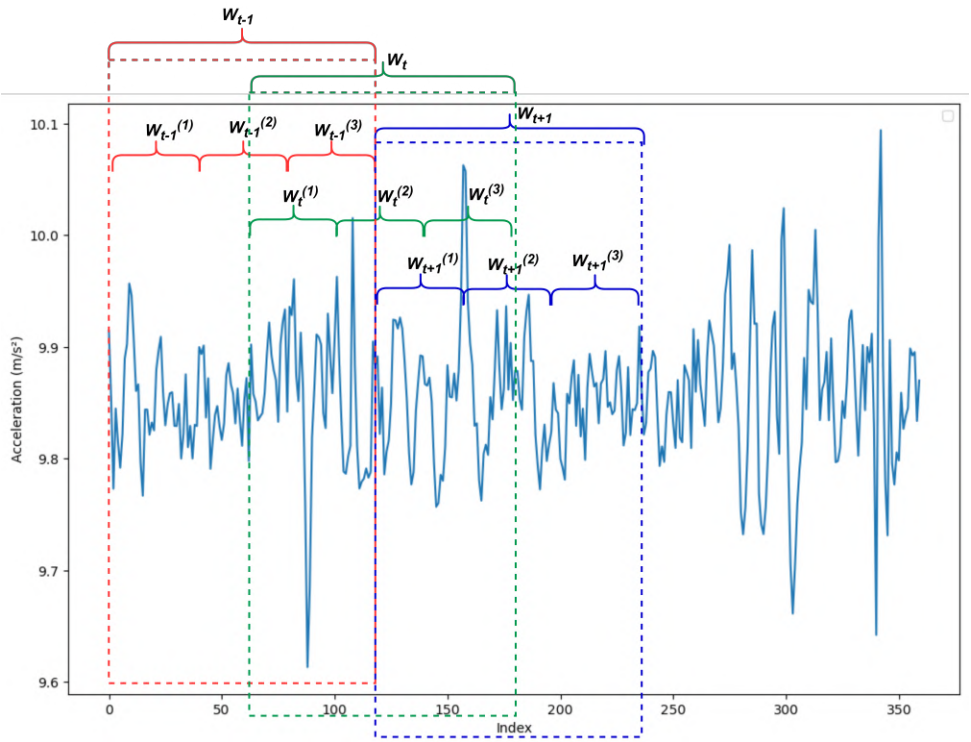



**Figure 2** Illustration of the overlapping and triplet windows

### 3.1.3 Basic feature extraction

Basic features refer to the features used to compute distance or similarity measures. These features include the minimum, maximum, mean, median, standard deviation, and interquartile range, which were computed from the SVA values. These features are extracted from each subwindow using Equations (2)–(7).

$$f_{min} = \min_{1 \le k \le n} (SVA_k) \tag{2}$$

$$f_{max} = \max_{1 \le k \le n} (SVA_k) \tag{3}$$

$$f_{\mu} = \frac{1}{n} \sum_{k=1}^{n} (SVA_k) \tag{4}$$

$$f_{med} = \begin{cases} SVA_{\frac{n+1}{2}}, & \text{if } n \text{ odd} \\ \frac{1}{2}\left(SVA_{\frac{n}{2}} + SVA_{\frac{n}{2}+1}\right), & \text{if } n \text{ even} \end{cases} \tag{5}$$

$$f_{\sigma} = \sqrt{\frac{1}{n} \sum_{k=1}^{n} (SVA_k - f_{\mu})^2} \tag{6}$$

$$f_{IQR} = Q_3(SVA) - Q_1(SVA) \tag{7}$$

where:

- $f_{min}$, $f_{max}$, $f_{\mu}$, $f_{med}$, $f_{\sigma}$, $f_{IQR}$ denote the minimum, maximum, mean, median, standard deviation, and interquartile range, respectively. The index $k$ represents the $kth$ sample within a sub-window, and $n$ denotes the number of samples in the sub-window.
- $Q_1$ represents the first quartile of the SVA values, corresponding to the value at the $25^{th}$ percentile after sorting the SVA data, and $Q_3$ represents the third quartile of the SVA values, corresponding to the value at the $75^{th}$ percentile after sorting the data.

### 3.1.4 Sub-Window Similarity

The similarity is computed using distance-based metrics, namely, the Manhattan, Euclidean, and Minkowski distances. The distances among the sub-windows $W_{t-1}^{(j)}$, $W_t^{(j)}$, and $W_{t+1}^{(j)}$ are grouped into three pairs, collectively referred to as the triplet sub-window. The triplet sub-window represents the distance between the corresponding sub-windows. The distance or similarity functions are computed using Equations (8)–(10).

$$d_{pc}^{(j)} = D\left(FW_{t-1}^{(j)}, FW_t^{(j)}\right) \tag{8}$$

$$d_{cf}^{(j)} = D\left(FW_t^{(j)}, FW_{t+1}^{(j)}\right) \tag{9}$$

$$d_{pf}^{(j)} = D\left(FW_{t-1}^{(j)}, FW_{t+1}^{(j)}\right) \tag{10}$$

$$D_{uv}^{(j)}(p) = \begin{cases} \sum_{i=1}^{n} \left|FW_{u,i}^{(j)} - FW_{v,i}^{(j)}\right|, & p = 1 \text{ (Manhattan)} \\ \sqrt{\sum_{i=1}^{n} \left(FW_{u,i}^{(j)} - FW_{v,i}^{(j)}\right)^2}, & p = 2 \text{ (Euclidean)} \\ \left(\sum_{i=1}^{n} \left|FW_{u,i}^{(j)} - FW_{v,i}^{(j)}\right|^p\right)^{\frac{1}{p}}, & p = 3 \text{ (Minkowski)} \end{cases} \tag{11}$$

where:

- $d_{pc}^{(j)}$, $d_{cf}^{(j)}$, and $d_{pf}^{(j)}$ denote the distance or similarity values between the subwindows of the past and current, current and future, and past and future windows, respectively.
- *D(.,.)* represents the distance or similarity function computed using the Manhattan, Euclidean, or Minkowski distance metrics, respectively, as defined in Equation (11).
- $FW_{t-1}^{(j)}$ denotes the set of basic features extracted from the $j$-th sub-window of the past window, defined as follows:

$$\left[f_{min}\left(W_{t-1}^{(j)}\right), f_{max}\left(W_{t-1}^{(j)}\right), f_{\mu}\left(W_{t-1}^{(j)}\right), f_{med}\left(W_{t-1}^{(j)}\right), f_{\sigma}\left(W_{t-1}^{(j)}\right), f_{IQR}\left(W_{t-1}^{(j)}\right)\right]$$

- Similarly, $FW_t^{(j)}$ represents the set of basic features extracted from the $j$-th sub-window of the current window, defined as follows:

$$\left[f_{min}\left(W_t^{(j)}\right), f_{max}\left(W_t^{(j)}\right), f_{\mu}\left(W_t^{(j)}\right), f_{med}\left(W_t^{(j)}\right), f_{\sigma}\left(W_t^{(j)}\right), f_{IQR}\left(W_t^{(j)}\right)\right]$$

- Likewise, $FW_{t+1}^{(j)}$ denotes the set of basic features extracted from the $j$-th sub-window of the future window, defined as follows:

$$\left[f_{min}\left(W_{t+1}^{(j)}\right), f_{max}\left(W_{t+1}^{(j)}\right), f_{\mu}\left(W_{t+1}^{(j)}\right), f_{med}\left(W_{t+1}^{(j)}\right), f_{\sigma}\left(W_{t+1}^{(j)}\right), f_{IQR}\left(W_{t+1}^{(j)}\right)\right]$$

- The statistical feature functions $f_{min}$, $f_{max}$, $f_{\mu}$, $f_{med}$, $f_{\sigma}$, and $f_{IQR}$ are computed using Equations (2)–(7).

- $D_{uv}^{(j)}(p)$ denotes the distance function, with $p = 1$ corresponding to the Manhattan distance, $p = 2$ to the Euclidean distance, and $p = 3$ to the Minkowski distance. The indices $u, v \in \{t-1, t, t+1\}$, where $u = t-1$ and $v = t$ correspond to $d_{pc}^{(j)}$, $u = t$ and $v = t+1$ correspond to $d_{cf}^{(j)}$, and $u = t-1$ and $v = t+1$ correspond to $d_{pf}^{(j)}$.
- Furthermore, $i \in \{1, 2, \ldots, 6\}$ and $f_i \in \mathcal{F}$, where

$$\mathcal{F} = \{f_{min}, f_{max}, f_{\mu}, f_{med}, f_{\sigma}, f_{IQR}\}$$

### 3.1.5 Averaging and Normalization

The study uses only a single accelerometer sensor; therefore, normalization is performed only on a single sensor channel. There are three windows, each consisting of three sub-windows. As denoted in the previous equations and referred to as the triplet window, $d_{pc}^{(j)}$ represents the distance or similarity between the past and current windows (pc), $d_cf^{(j)}$ represents the distance or similarity between the current and future windows (cf), and $d_pf^{(j)}$ represents the distance or similarity between the past and future windows (pf), with j=1,2,3. Because each triplet window produces three distance or similarity values, their average value is computed using Equation (12). Only three aggregated distance or similarity values were obtained. A normalization process is applied to scale the values between 0 and 1 to ensure that the similarity values lie within a consistent range, as computed using Equation (13).

$$\bar{d}^{(j)} = \frac{1}{3}\left(d_{pc}^{(j)} + d_{cf}^{(j)} + d_{pf}^{(j)}\right) \tag{12}$$

$$\hat{d}_{norm}^{(j)} = \frac{\bar{d}^{(j)} - d_{min}}{d_{max} - d_{min}} \tag{13}$$

where:

- $\bar{d}^{(j)}$ denotes the average distance or similarity value for a given triplet,
- $d_{min}$ represents the minimum value among all $\bar{d}^{(j)}$, and
- $d_{max}$ represents the maximum value among all $\bar{d}^{(j)}$.
- $\hat{d}_{norm}^{(j)}$ denotes the normalized average value, scaled to lie within the range of 0–1.

### 3.1.6 Voting and Base-Window Update

To perform voting and update the base window, Equations (14)–(18) are applied. The similarity indicator for each triplet window is determined based on the similarity indicator $I_j$, which is derived from the normalized value $\hat{d}_{norm}^{(j)}$. Smaller values of $\hat{d}_{norm}^{(j)}$ approaching zero, indicate a higher degree of similarity. A voting process is conducted using the three values of $\hat{d}_{norm}^{(j)}$. If the similarity votes, denoted as $votes_{sim}$, are greater than or equal to 2, then the window update process is performed by multiplying the normalized average value $avg_{norm}$ with the base window *($W_0$)* and then adding the base window *($W_0$)* itself. The maximum width of the updated window ($W_{new}$) was limited to twice the size of the base window *($W_0$)*. This constraint was introduced to prevent continuous window growth, which could increase computational complexity.

$$I_j = \begin{cases} 1, & \text{if } \hat{d}_{norm}^{(j)} < \tau \\ 0, & \text{else} \end{cases} \tag{14}$$

$$votes_{sim} = \sum_{j=1}^{n} I_j \tag{15}$$

$$avg_{norm} = \frac{1}{n}\sum_{j=1}^{n} \hat{d}_{norm}^{(j)} \tag{16}$$

$$W_{new} = \begin{cases} \min\left(\max\left(W_0, [W_0 + W_0 \cdot avg_{norm}]\right), cap\right), & \text{if } votes_{sim} \geq 2 \\ W_0, & \text{if } votes_{sim} = 1 \\ W_0, & \text{if } votes_{sim} < 1 \end{cases} \tag{17}$$

$$cap = max_{fact} \cdot W_0 \tag{18}$$

where:

- $I_j$ denotes the similar triplet window indicator based on the predefined similarity threshold $\tau$. A value of $I_j = 1$ indicates that the similarity condition has been satisfied. The parameter $\tau$ represents the majority threshold, which is used as the similarity criterion.
- $votes_{sim}$ denotes the number of indicators $I_j$ that satisfy the similarity condition.
- $avg_{norm}$ represents the average value of $\hat{d}^{(j)}_{norm}$. Because three values of $\hat{d}^{(j)}_{norm}$ are obtained, their mean is computed and subsequently used to determine the update rate of the newly adjusted window.
- $W_{new}$ denotes the updated window length, with the minimum window size defined as the base window ($W_0$), which is set to 120 samples.
- The parameter *cap* represents the maximum allowable window length, whereas $max_{fact}$ denotes the scaling factor used to define this upper bound. In this study, $max_{fact}$ was limited to 2 to prevent excessive window expansion.

### 3.2 Feature extraction and model machine learning

The data sequence generated from the sliding window process using the semi-adaptive sliding window technique was subsequently subjected to feature extraction and classified using ML algorithms. At this stage, the features are extracted from the three axes of the accelerometer sensor (x, y, and z). These procedures are described in detail in Sections 3.2.1 and 3.2.2.

#### 3.2.1 Feature Extraction

Features are extracted from the statistical and frequency domains. The statistical features include the minimum, maximum, mean, median, standard deviation, and interquartile range, which are computed using (2) through (7), and the average energy and the number of signal direction changes, which are calculated using (19) and (20), respectively. Unlike the similarity computation stage, these statistical features are not derived from the Signal Vector Magnitude (SVA); instead, they are extracted independently from each accelerometer axis (x, y, and z). Frequency-domain features, including the dominant frequency, highest energy intensity, and total spectral energy, are extracted using the fast Fourier transform (FFT), which are computed using Equations (21)–(23). All feature extraction equations were applied independently to the three axes of the accelerometer (x, y, and z).

$$rms(w) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} w_i^2} \tag{19}$$

$$zcr(w) = \frac{1}{N-1}\sum_{i=1}^{N-1} 1(w_i \cdot w_{i+1} < 0) \tag{20}$$

$$f_{peak} = \arg\max_{f_k} P(f_k), \qquad P(f_k) = |S(f_k)|^2 \tag{21}$$

$$P_{peak} = \max_{f_k} P(f_k) \tag{22}$$

$$P_{band} = \sum_{k=1}^{K} P(f_k) \tag{23}$$

where:

- $w_i$ and $w_{i+1}$ denote the sensor acceleration values at the $i$-th sample and the subsequent sample, respectively.
- $N$ is the total number of samples in a single window.
- $rms(w)$ denotes the average signal energy within the corresponding time window.
- The notation $w_i \cdot w_{i+1} < 0$ indicates a Boolean condition for two consecutive samples $w_i$ and $w_{i+1}$. The condition is true when the signs of $w_i$ and $w_{i+1}$ are opposite and false otherwise.
- The indicator function $1(\cdot)$ takes the value of 1 when a sign change occurs (from positive to negative or vice versa) and 0 otherwise.
- $zcr(w)$ denotes the number of signal direction changes, useful for identifying dynamic activities.
- $S(f_k)$ represents the Fast Fourier Transform (FFT) of the signal $w$.
- $P(f_k)$ denotes the power spectrum at frequency $f_k$.
- $f_{peak}$ is the dominant frequency at which the signal energy reaches its maximum value.
- $P_{peak}$ represents the highest energy intensity in the frequency domain.
- $K$ denotes the total number of frequency components obtained from the FFT.
- Finally, $P_{band}$ represents the total spectral energy over the entire frequency band of the signal.

#### 3.2.2 Model machine learning

In this study, the Extreme Gradient Boosting (XGBoost) and Light Gradient Boosting Machine (LightGBM) algorithms were employed to classify human activities based on features extracted from the statistical and frequency domains. Each windowed segment was analyzed to extract statistical features, including the minimum, maximum, mean, median, standard deviation, interquartile range, root mean square, and zero-crossing rate. In addition, we extracted frequency-domain features, such as the spectral peak frequency, spectral peak power, and spectral bandpower. Then, these features were used as inputs to the XGBoost and LightGBM models for activity classification. The performance of the ML models was evaluated using several standard metrics, namely, accuracy, precision, recall, and F1-score.

#### 3.2.3 Analysis of variance-based feature selection

Appropriate feature selection can improve the performance of machine learning algorithms by retaining only the most relevant features (Wang et al., 2021). Analysis of variance (ANOVA) is one of the most popular and effective feature selection techniques (Pathan et al., 2022; Wang et al., 2023a; Wang et al., 2022). ANOVA compares the variance between groups with that within groups (Liandana et al., 2024). Each feature subset generated through ANOVA-based feature selection was evaluated using machine learning algorithms.

## 4. Experimental Setup

### 4.1 Dataset

The effectiveness of the proposed model was evaluated using four public datasets. The datasets used in this study include SBHARPT (Reyes-Ortiz et al., 2016), FORTH-TRACE (Karagiannaki et al., 2016), WISDM (WISDM Lab, n.d.), and PAMAP2 (Reiss and Stricker, 2012). Although these datasets contain data collected from multiple sensors, only accelerometer sensor data were used in this study to reduce the dimensionality of the data. A brief description of each dataset is provided below.

1. The SBHARPT dataset (Reyes-Ortiz et al., 2016) contains accelerometer and gyroscope data collected using a smartphone at a sampling frequency of 50 Hz. This dataset includes three categories of body postures: static postures (standing, sitting, and lying), dynamic activities (walking, walking upstairs, and walking downstairs), and transitional postures (standing to sitting, sitting to standing, sitting to lying, lying to sitting, standing to lying, and lying to standing). In total, 12 activities were recorded from 30 participants aged 19–48 years.
2. The FORTH-TRACE dataset (Karagiannaki et al., 2016) comprises accelerometer, gyroscope, and magnetometer data sampled at 51.2 Hz. Data were collected from 15 participants who performed seven primary activities and nine postural transition activities. The selected activities included standing, sitting, walking, stair climbing, and transitions between standing and sitting positions. These activities were chosen because of the limitations of accelerometer and gyroscope sensors in detecting non-physical activities, such as speaking (Munoz-Organero et al., 2022). Posture transitions between dynamic movements, such as stair climbing followed by walking, may lead to classification errors (Baraka and Mohd Noor, 2023).
3. The WISDM Dataset consists of accelerometer data collected from Android devices and serves as a benchmark dataset in human activity recognition studies by the Wireless Sensor Data Mining Laboratory (WISDM Lab: Dataset, n.d.). Data were obtained from 36 participants performing various activities, including sitting, jogging, walking upstairs and downstairs, standing, and walking. Acceleration signals were recorded at a sampling interval of 50 ms.
4. The PAMAP2 Dataset (Reiss and Stricker, 2012) records 18 physical activities performed by nine participants, comprising eight males and one female. Data were acquired using inertial measurement units (IMUs) and heart rate sensors. The IMU setup included three accelerometers, one gyroscope, and one magnetometer with a sampling frequency of 100 Hz. The IMU sensors were positioned at three body locations: the wrist, ankle, and chest. Each participant performed 12 activity protocols, including lying, sitting, standing, walking, running, cycling, Nordic walking, ironing, vacuum cleaning, rope jumping, stair ascent, and stair descent. In addition, the participants performed several supplementary activities, such as watching television, using a computer, driving a car, folding laundry, cleaning the house, and playing soccer.

### 4.2 Parameter Setup

To evaluate the performance of the proposed model, we employed two machine learning algorithms, namely, Extreme Gradient Boosting (XGBoost) and Light Gradient Boosting Machine (LightGBM). These algorithms were implemented using the Python programming language with scikit-learn, XGBoost, and LightGBM libraries as the primary tools. Table 1 lists the machine learning parameters used in this study.

**Table 1** Setup of machine learning parameters

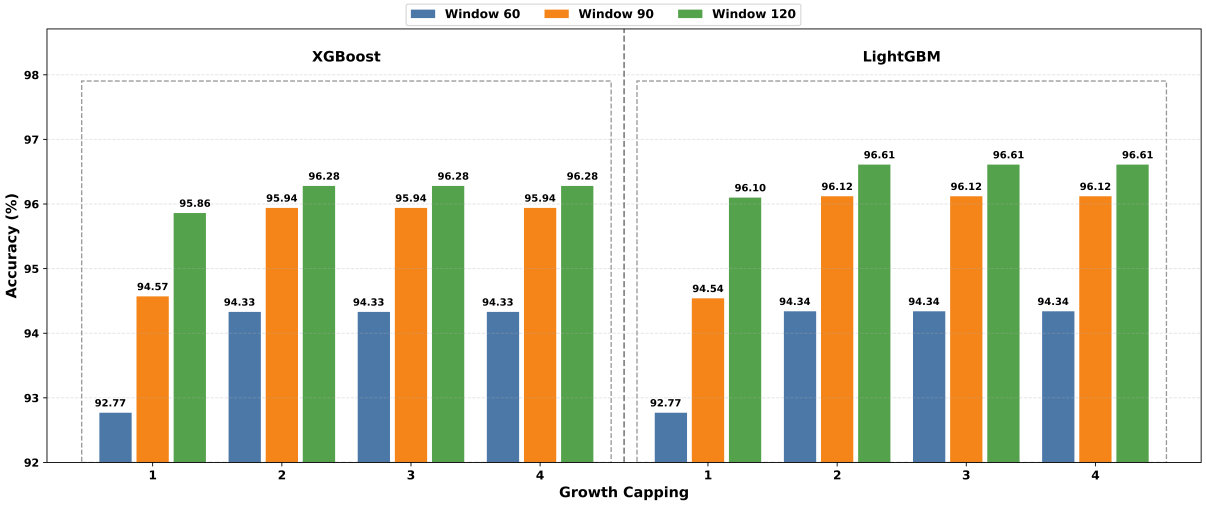
| Model | Library | Parameters and values |
|---|---|---|
| XGB | XGBoost | XGBClassifier(n_estimators=300, learning_rate=0.1, max_depth=6, subsample=0.8, colsample_bytree=0.8, random_state=42, eval_metric="mlogloss", use_label_encoder=False |
| LGBM | LightGBM | LGBMClassifier(n_estimators=300, learning_rate=0.1, max_depth=-1, subsample=0.8, colsample_bytree=0.8, random_state=42) |

## 5. Results and Discussion

The windows generated by the semi-adaptive sliding window process were subjected to feature extraction, resulting in 11 features for each sensor axis, yielding 33 features. According to equation (14), the similarity indicator is determined based on the majority threshold ($\tau$), which ranges from 0.0 to 1.0. The comparison between $\hat{d}^{(j)}_{norm}$ and the majority threshold ($\tau$) determines the similarity decision, where smaller values of $\hat{d}^{(j)}_{norm}$ indicate a higher degree of similarity. However, the accelerometer sensor data representing different activities exhibit varying magnitudes depending on the recorded motions. Consequently, the values of each sub-window within a window also vary when computing the nearest distance or similarity. Therefore, the selection of the majority threshold ($\tau$) must be carefully evaluated to identify its optimal value for the application. In this study, the majority threshold ($\tau$) was evaluated incrementally from 0.1 to 1.0. For each evaluated threshold value, feature selection was performed using ANOVA to determine the optimal number of features. The Manhattan, Euclidean, and Minkowski distance metrics were also evaluated to compute the similarity values.

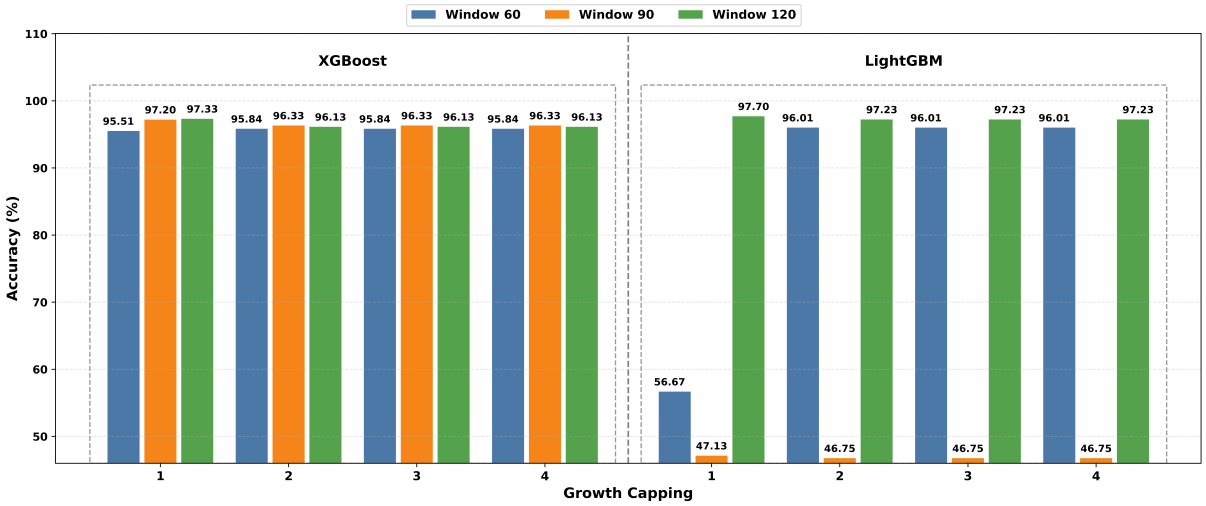
### 5.1 Evaluation of the Base Window Size and Growth Capping

The base window length and maximum window expansion limit (growth-capping) are critical components in determining the temporal data representation quality. To identify the optimal configuration, an evaluation was conducted using base window sizes of 60, 90, and 120, along with growth-capping values of 1–4. All features obtained from the feature extraction stage were used. Tables S1–S4 and Figure 3 present the results of this evaluation. Based on the FORTH_TRACE dataset's experimental results (Table S1 and Figure 3 (a)), increasing the growth-capping value from 1 to 2 consistently improved all evaluation metrics, with the best performance achieved at a window size of 120 (accuracy of 96.61 and F1-score of 96.56). In contrast, growth-capping values of 3 and 4 did not yield further improvements, indicating a plateau in performance. For the SBHARPT dataset (Table S2 and Figure 3 (b)), although growth-capping 1 achieves the highest performance under certain configurations, particularly with XGBoost at a window size of 120, this configuration exhibits considerable variability across models, as evidenced by the significant performance degradation of LightGBM at window sizes of 60 and 90. Consequently, it is considered less stable than growth-capping 2. A window size of 120 consistently provided the highest and most stable performance.

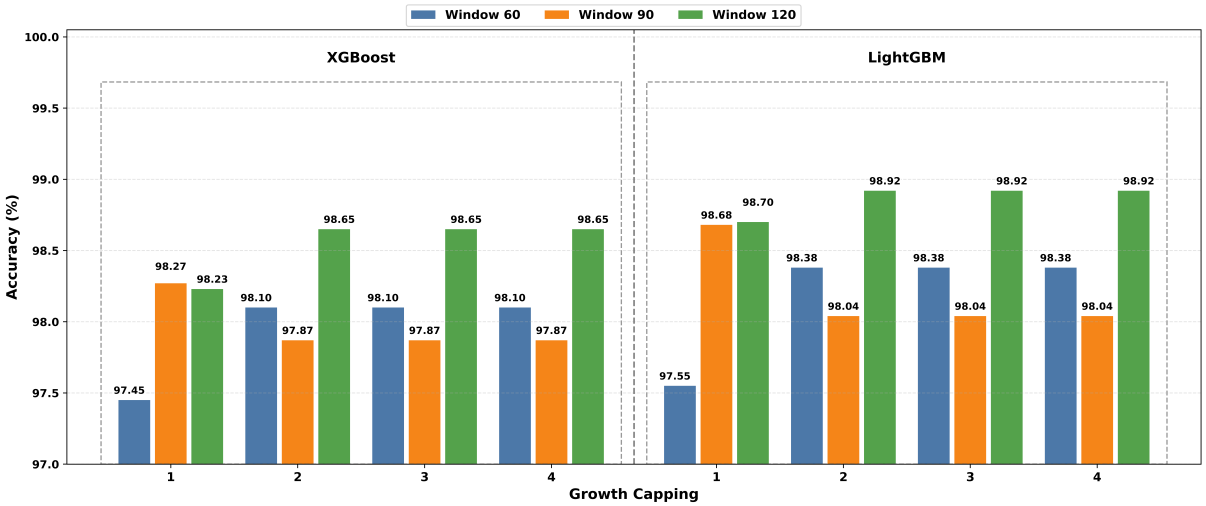



**(a)**

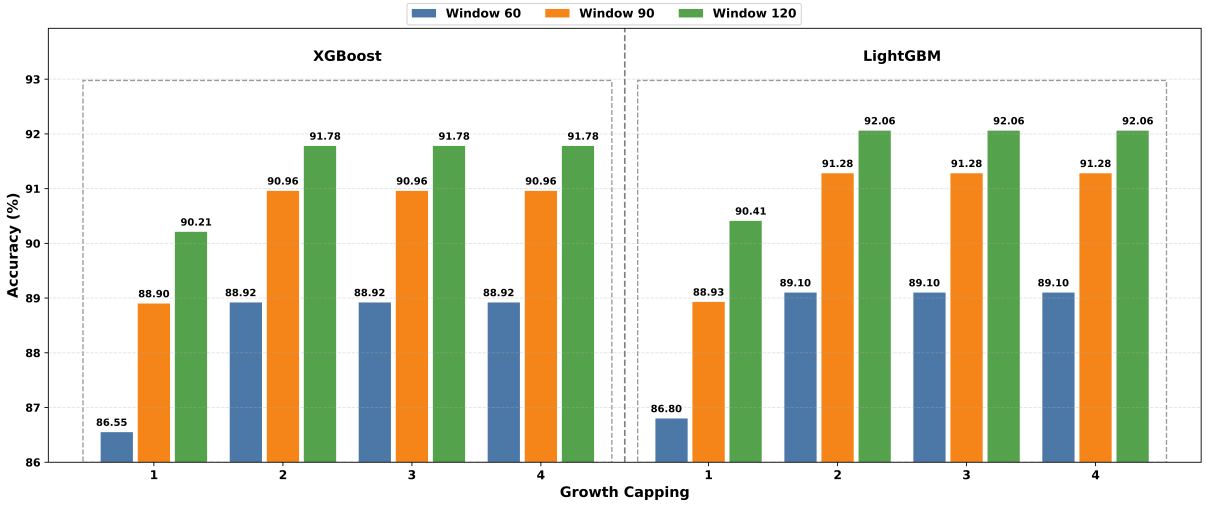
**Figure 3** Evaluation of the Base Window Size and Growth Cap. (a) FORTH TRACE Dataset; (b) SBHARPT Dataset; (c) WISDM Dataset; (d) PAMA2 Dataset




(b)




(c)




(d)

**Figure 3** Evaluation of the Base Window Size and Growth Cap. (a) FORTH TRACE Dataset; (b) SBHARPT Dataset; (c) WISDM Dataset; (d) PAMA2 Dataset (cont.)

The combination of growth-capping 2 and window size 120 yielded the best overall performance on the WISDM dataset (Table S3 and Figure 3 (c)), with all evaluation metrics

approaching 98.92. This further confirms that increasing the growth-capping value to 3 and 4 does not provide any additional benefits. A similar trend was observed in the PAMA2 dataset (Table S4 and Figure 3 (d)), where a substantial performance improvement was achieved when increasing growth-capping from 1 to 2, with the best results obtained at a window size of 120 (accuracy reaching 92.06), while performance remained unchanged for growth-capping values of 3 and 4. Overall, despite the exception observed in the SBHARPT dataset, the combination of a growth-capping of two and a window size of 120 can be identified as the most optimal and robust configuration. This combination consistently demonstrates a strong and reliable performance across most datasets while maintaining stability across different models. Therefore, the growth-capping values of 3 and 4 were not selected because they did not provide significant performance improvements (i.e., they exhibited a plateau effect) and only introduced additional computational complexity without meaningful gains in model performance.

### 5.2 Evaluation Based on Threshold Majority ($\tau$), Number of Features, and Metric Distancecs

After determining the optimal base window size and growth-capping parameter, further evaluation was conducted on the majority threshold (($\tau$)), number of features, and distance metrics. Tables S5–S12 present a summary of the evaluation process. Based on the FORTH-TRACE dataset's experimental results (Tables S5 and S6 and Figure 4), the best performance was achieved using the XGBoost classifier with a threshold (($\tau$)) value of 1, 33 selected features, and the Manhattan distance metric. This configuration resulted in accuracies, precisions, recalls, and F1-scores of 97.26%, 97.21%, 97.26%, and 97.20%, respectively. The LightGBM classifier achieved its best performance using the same Manhattan distance metric and a threshold ($\tau$) value of 1, with 32 selected features, yielding accuracies, precisions, recalls, and F1-scores of 97.26%, 97.21%, 97.26%, and 97.22%, respectively.

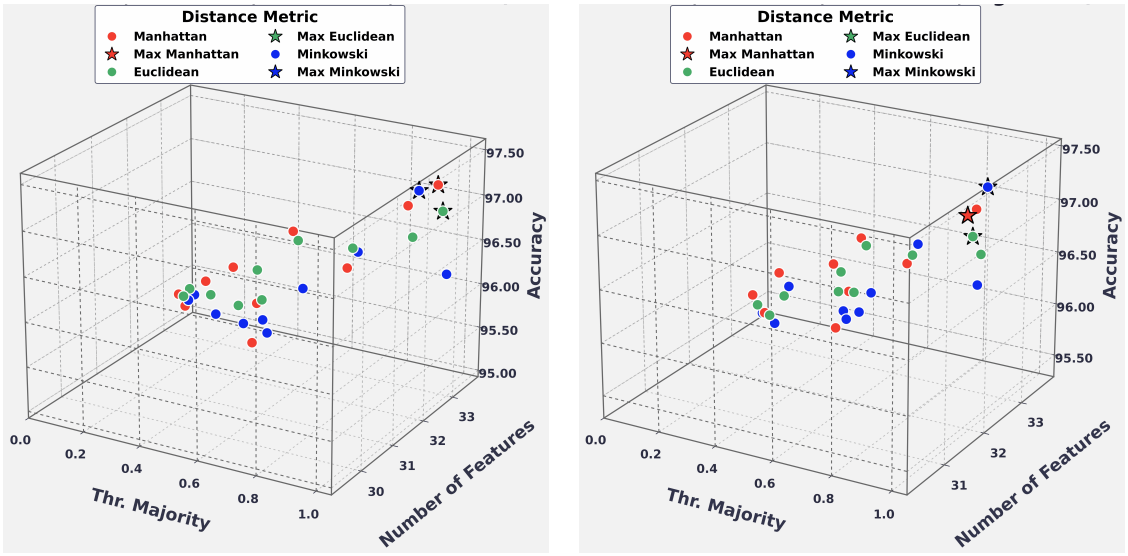



**Figure 4** Accuracy Comparison of Distance Metrics on the FORTH TRACE Dataset. (a) XGBoost; (b) LightGBM

Tables S7 and S8 present the experimental results for the SBHARPT dataset and Figure 5. The best performance was achieved using the XGBoost classifier with a threshold ($\tau$) value of 0.3, 27 selected features, and the Minkowski distance metric, resulting in an accuracy, precision, recall, and F1-score of 98.09%, 98.01%, 98.09%, and 98.00%, respectively. In contrast, the LightGBM classifier demonstrated superior performance using the Minkowski distance metric, a threshold ($\tau$) value of 0.3, and 29 selected features, achieving accuracy, precision, recall, and F1-score of 98.15%, 98.17%, 98.15%, and 98.13%, respectively.

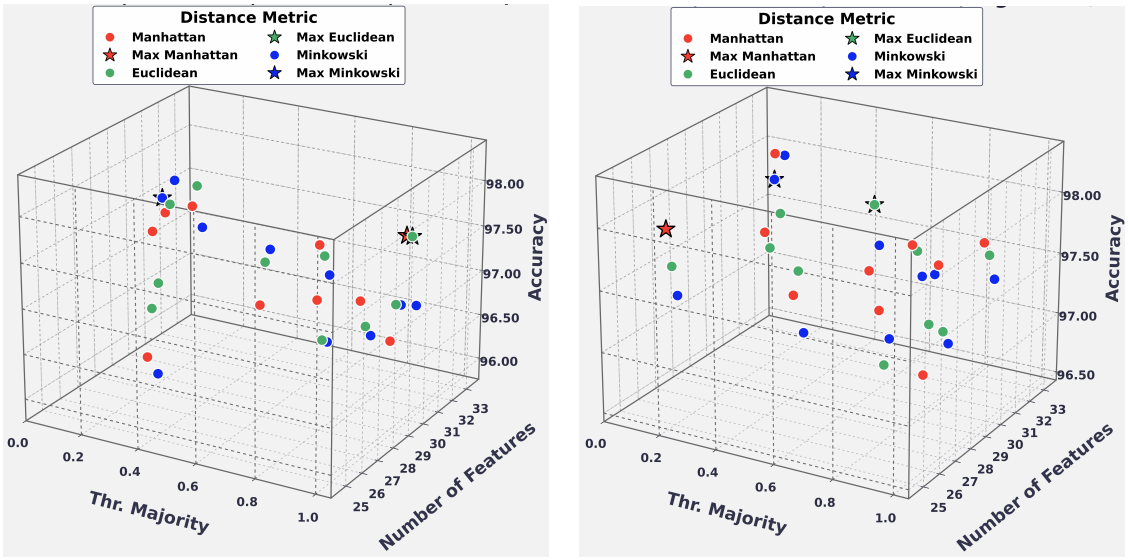



**Figure 5** Accuracy Comparison of Distance Metrics on the SBHARPT Dataset. (a) XGBoost; (b) LightGBM

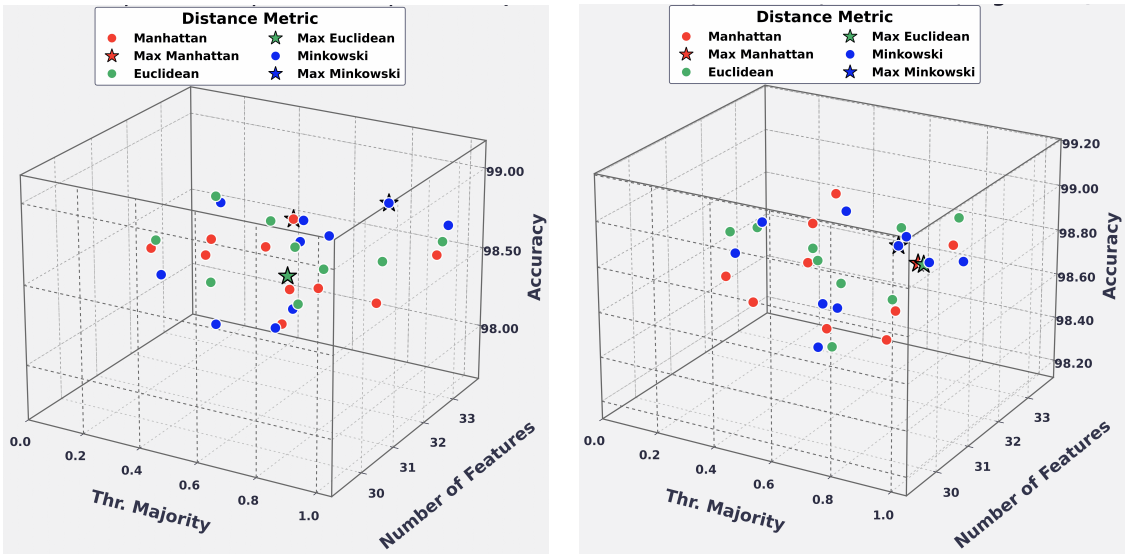



**Figure 6** Accuracy Comparison of Distance Metrics on the WISDM Dataset. (a) XGBoost; (b) LightGBM

For the WISDM dataset, as shown in Tables S9 and S10 and illustrated in Figure 6, the XGBoost classifier achieved optimal performance with a threshold ($\tau$) value of 0.9, 32 selected features, and the Minkowski distance metric, resulting in accuracy, precision, recall, and F1-score of 98.97%, 98.98%, 98.97%, and 98.97%, respectively. In contrast, the LightGBM classifier achieved a higher performance with a threshold ($\tau$) value of 0.9, 30 selected features, and the Minkowski distance metric, yielding an accuracy, precision, recall, and F1-score of 99.06%, 99.06%, 99.06%, and 99.06%, respectively.

Furthermore, the experimental results on the PAMA2 dataset, as presented in Tables S11 and S12 and illustrated in Figure 7, indicate that the XGBoost classifier achieved optimal performance with a threshold ($\tau$) value of 0.9, 31 selected features, and the Minkowski distance metric, resulting in accuracy, precision, recall, and F1-score of 92.21%, 92.52%, 92.21%, and 92.28%, respectively. For the same dataset, the LightGBM classifier demonstrated superior performance with a threshold ($\tau$) value of 0.9, 30 selected features, and the Minkowski distance

metric, achieving an accuracy, precision, recall, and F1-score of 92.43%, 92.73%, 92.43%, and 92.49%, respectively.

Based on the variation in the similarity threshold ($\tau$), a consistent pattern can be observed across multiple datasets, where the model performance tends to improve as $\tau$ increases. Lower threshold values ($\tau = 0.1$–$0.3$) result in relatively lower accuracy, whereas higher threshold values ($\tau = 0.8$–$1.0$) yield more stable and optimal performance. This indicates that stricter similarity criteria lead to more consistent window segmentation, thereby enhancing classification performance. In terms of similarity metrics, no single metric consistently outperforms others across all datasets. In certain cases, the Manhattan metric shows superior performance, whereas the Euclidean and Minkowski metrics provide a more stable or higher performance under different conditions. Notably, the Minkowski metric often achieves competitive results as a generalized form of distance measurement because of its flexibility. The effectiveness of similarity metrics is highly dependent on the data characteristics. Furthermore, the number of selected features varied across different threshold values, indicating an interaction between the feature selection process and the window size parameters. This implies that the optimal configuration is not governed by a single parameter but rather by the combined effect of the similarity threshold, distance metric, and feature set. Overall, the choice of similarity threshold and distance metric strongly influenced the performance of the proposed method.

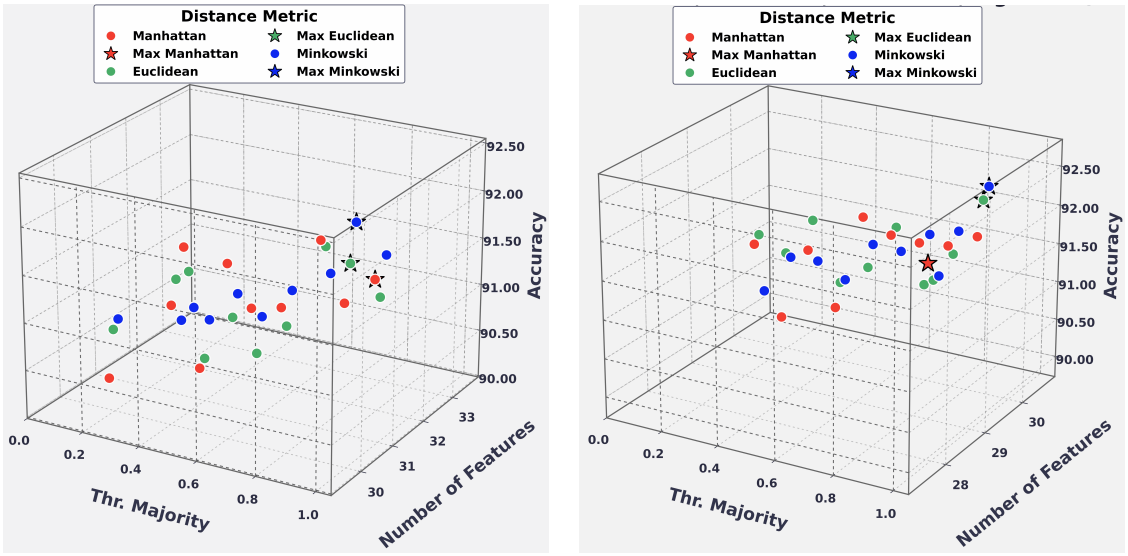



**Figure 7** Accuracy Comparison of Distance Metrics on the PAMA2 Dataset. (a) XGBoost; (b) LightGBM

Figure 8 presents a comparison of the optimal performance of XGBoost and LightGBM for each dataset. The optimal values were derived from the highest accuracy results reported in Tables S5–S12. The comparison indicates that when using the Minkowski distance metric, LightGBM achieves higher accuracy than XGBoost, particularly for the SBHARPT, WISDM, and PAMA2 datasets. However, the statistical significance test using the Wilcoxon signed-rank test, as reported in Table 2, shows that the performance difference is not statistically significant because the ptwo-side value is greater than 0.05. This indicates that both ML algorithms can be regarded as equivalent alternatives for the evaluated datasets, as the observed performance differences are not statistically significant.

### 5.3 Evaluation of robustness and CPU execution time

An additional experiment was conducted by simulating orientation shifts in the test data at angles of 1°, 4°, 7°, and up to 15° to evaluate the robustness of the proposed model. These perturbations were applied to the sensor values along the x, y, and z axes, which directly affected the

extracted features, including the minimum, maximum, median, and interquartile values. This scenario was designed to represent real-world conditions, particularly variations in the sensor position or orientation during practical use. The models evaluated in this scenario correspond to those that achieved the best performance in terms of accuracy, precision, recall, and F1-score (Tables S5–S12). These models were then re-evaluated using the perturbed test data, and the performance differences between the original and perturbed conditions were calculated and denoted as Δacc, Δprec, Δrec, and ΔF1. Δ values close to zero indicate higher robustness, as they reflect the model's ability to maintain a consistent performance despite input data variations.

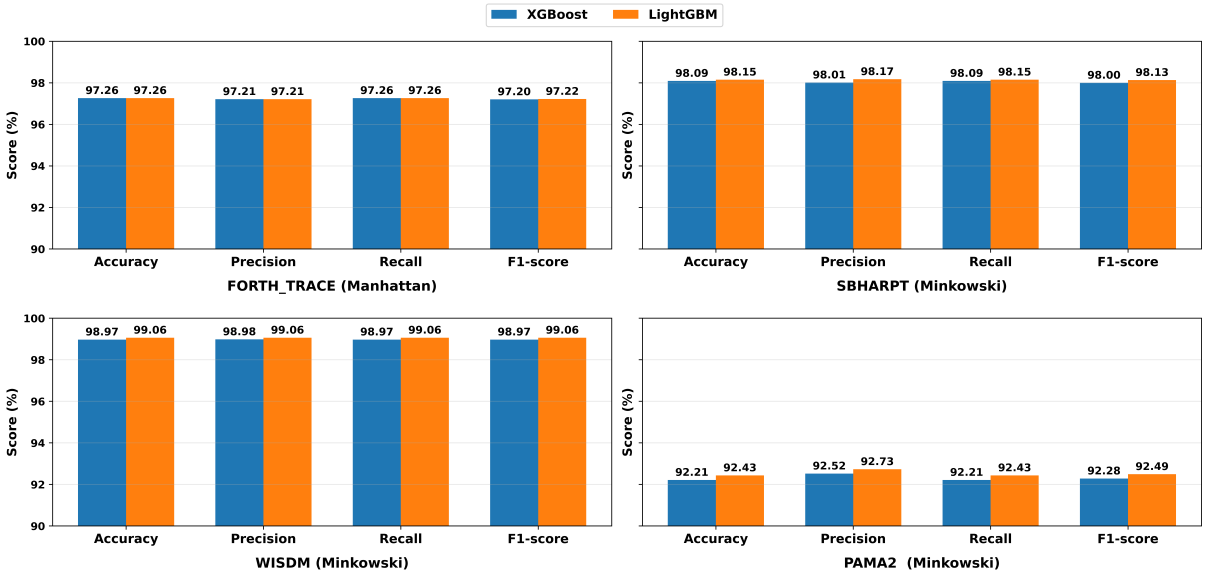



**Figure 8** Comparison between XGBoost and LightGBM (Optimal Performance)

**Table 2** Statistical significance test using the Wilcoxon signed-rank test

| Dataset | XGBoost | LightGBM | XGBoost-LightGBM | Rank Absolute |
|---|---|---|---|---|
| FORTH_TRACE | 97.26 | 97.26 | 0.00 | |
| SBHARPT | 98.09 | 98.15 | -0.06 | 1 |
| WISDM | 98.97 | 99.06 | -0.09 | 2 |
| PAMA2 | 92.21 | 92.43 | -0.22 | 3 |
| $\alpha = 0.05$, $p_{\text{two-side}} = 0.25$, result: not significant | | | | |

Based on the results presented in Tables S14–S21, a consistent pattern can be observed across all datasets, where increasing the rotation angle leads to progressively larger degradation in the model performance. In general, Δ values remained close to zero only at small rotation angles, specifically within the range of 1°–4°, indicating that the model was capable of maintaining stable performance under minimal sensor perturbations. At rotation angles of $\leq$ 4°, the performance degradation remains relatively small (with Δacc generally below 2%), and can therefore be considered an acceptable tolerance range for orientation variation. However, at rotation angles of $\geq$ 7°, a more significant and consistent decline in performance across all datasets indicated that the model began to lose its ability to preserve stable feature representations. Therefore, the proposed model's robustness threshold can be identified as approximately 4°, where the Δ values remain relatively small and close to zero. Overall, the proposed approach demonstrates good robustness to minor variations in the sensor orientation (up to approximately ±4°), whereas larger perturbations result in a more pronounced performance degradation.

**Table 3** Evaluation of Performance Against Prior Work

| Reference | Dataset | Sliding Window Methods | Acc (%) | Prec (%) | Rec (%) | F1(%) | Descript. |
|---|---|---|---|---|---|---|---|
| Baraka and Mohd Noor, 2023 | FORTH-TRACE | static sliding window | 86.65 | - | - | - | - |
| | SBHARPT | | 92.71 | - | - | - | - |
| Baraka and Mohd Noor, 2024 | FORTH-TRACE | dynamic sliding window | 84.96 | - | - | - | - |
| | SBHARPT | | 93.35 | - | - | - | - |
| Shi et al., 2022 | WISDM | static sliding window | - | 95.53 | 94.83 | 95.18 | - |
| Akter et al., 2023 | WISDM | static sliding window | 93.89 | - | - | - | - |
| Zhang et al., 2022 | WISDM | static sliding window | 97.18 | - | - | 97.17 | - |
| Pan et al., 2025 | PAMA2 | static sliding window | 73.34 | - | - | - | - |
| Yu and Al-Qaness, 2025 | PAMA2 | static sliding window | 89.72 | - | - | - | - |
| Liandana et al., 2025 | FORTH-TRACE | static sliding window | 94.54 | 94.50 | 94.54 | 94.47 | - |
| | SBHARPT | | 93.65 | 93.30 | 93.65 | 93.40 | - |
| Proposed Model | FORTH-TRACE | semi-adaptive sliding window | 97.26 | 97.21 | 97.26 | 97.26 | XGBoost/LightGBM |
| | SBHARPT | | 98.15 | 98.18 | 98.15 | 98.13 | LightGBM |
| | WISDM | | 99.06 | 99.06 | 99.06 | 99.06 | LightGBM |
| | PAMA2 | | 92.43 | 92.73 | 92.43 | 92.49 | LightGBM |

In addition to evaluating the classification performance, the computational resource utilization, particularly the CPU execution time measured in seconds, was analyzed to compare the efficiencies of the static sliding window method and the proposed semi-adaptive sliding window approach. Experiments were performed on a laptop equipped with an AMD Ryzen 7 6800H processor (3.20 GHz, 8 cores, 16 logical processors) and 16 GB main memory. The evaluation exclusively focuses on the sliding window process and feature extraction for each dataset. Based on the results presented in Table S13, the proposed semi-adaptive sliding window approach consistently improves the classification performance across all datasets and models, as evidenced by the increases in accuracy, precision, recall, and F1-score compared with the static sliding window method. However, these performance gains are accompanied by an increase in CPU execution time, ranging from 20.59% to 82.08%. This increase can be attributed to the additional computational complexity introduced by the proposed mechanism, such as the calculation of the similarity between sub-windows and the adaptive window updating process, which is absent in the static approach. However, the increase in computational cost exhibited a pattern that was proportional to the characteristics of the datasets. Datasets such as WISDM and FORTH_TRACE demonstrated higher increases in execution time, whereas SBHARPT showed a relatively lower increase. This variation is likely influenced by dataset complexity, including factors such as the number of participants involved in data collection, the presence of activity transition data, and the sensors' sampling rates. Overall, these findings indicate a balanced trade-off between performance and computational efficiency, where a relatively controlled increase in the execution

time yields consistent and significant improvements in classification performance. Therefore, the proposed approach remains viable, particularly for applications that prioritize accuracy, with further optimization potential to enhance the efficiency of real-time implementations.

## 6. Comparison of Performance with Previous Research

The proposed model's performance is compared with those of previous studies, as presented in Table 3. Only prior studies that employed the same dataset were considered to ensure a fair and relevant comparison. For the FORTH-TRACE dataset, the proposed model achieves an accuracy of 97.26%, corresponding to performance improvements of 10.61%, 12.30%, and 2.72% compared to Baraka and Mohd Noor, 2023, Baraka and Mohd Noor, 2024, and 2.72% compared to Liandana et al., 2025, respectively. For the SBHARPT dataset, the proposed model attains the highest accuracy of 98.15%, yielding improvements of 5.44%, 4.80%, and 4.50% over the results reported by Baraka and Mohd Noor, 2023, Baraka and Mohd Noor, 2024, and Liandana et al., 2025, respectively. The proposed model achieved an accuracy of 99.06% on the WISDM dataset, representing improvements of 5.17% over Akter et al., 2023 and 1.88% over Zhang et al., 2022. Meanwhile, for the PAMA2 dataset, the proposed model reaches an accuracy of 92.43%, providing improvements of 19.09% compared to Pan et al., 2025 and 2.71% compared to Yu and Al-Qaness, 2025. Overall, the proposed model achieves competitive performance and, in several cases, demonstrates improved results compared with previous approaches on the same datasets. However, given the potential differences in preprocessing pipelines, segmentation settings, and evaluation procedures used across studies, these comparisons should be interpreted with caution.

## 7. Conclusions

This study proposes a semi-adaptive sliding window framework that integrates static and dynamic approaches to improve human activity recognition performance. The proposed method exploits temporal information by considering the current, past, and future windows, where each window is divided into three sub-windows. Window updates are adaptively performed based on the similarity among pairs of subwindows forming a triplet subwindow and are constrained by a growth-capping mechanism to limit excessive window expansion. Experimental results on the FORTH-TRACE, SBHARPT, WISDM, and PAMA2 datasets demonstrated that the proposed model achieved a higher performance and consistently outperformed previous studies using the same datasets. This indicates that similarity-based semi-adaptive sliding window segmentation effectively captures temporal activity patterns and enhances the performance of human activity recognition. In this study, the majority threshold ($\tau$) and window growth-capping factor were experimentally determined. Although this approach is effective, parameter selection remains empirical and does not incorporate automatic optimization or adaptive learning mechanisms that enable dynamic adjustment based on data characteristics. Therefore, automatic optimization techniques or adaptive learning strategies should be developed to dynamically determine the majority threshold and growth-capping parameters. Such approaches are expected to enhance the flexibility and generalization capability of the proposed method, particularly when applied to datasets with diverse activity characteristics and signal patterns.

## Acknowledgements

The authors would like to acknowledge the Ministry of Higher Education, Science, and Technology of Indonesia (Kemdiktisaintek) for funding this research through the Fundamental Research Grant (Regular Scheme) 2025. The authors express their sincere gratitude to the Institut Teknologi dan Bisnis STIKOM Bali for its support in facilitating this research.

## Author Contributions

Liandana contributed to the research conceptualization, data processing, model development, experimental evaluation, and manuscript writing. Pradipta contributed to the conceptualization, data processing, model evaluation, and manuscript writing. Ayu contributed to the conceptualization, data processing, model evaluation, and manuscript writing. Hostiadi contributed to the conceptualization, data processing, model evaluation, and manuscript writing.

## Conflict of Interest

The authors declare no conflicts of interest.

## References

Akter, M., Ansary, S., Khan, M. A. M., & Kim, D. (2023). Human activity recognition using attention-mechanism-based deep learning feature combination. *Sensors*, *23*(12), 5715. https://doi.org/10.3390/S23125715

Baraka, A. M. A., & Mohd Noor, M. H. (2023). Similarity segmentation approach for sensor-based activity recognition. *IEEE Sensors Journal*, *23*(17), 19704–19716. https://doi.org/10.1109/JSEN.2023.3295778

Baraka, A. R., & Mohd Noor, M. H. (2024). Deep similarity segmentation model for sensor-based activity recognition. *Multimedia Tools and Applications*, 1–24. https://doi.org/10.1007/S11042-024-18933-2/METRICS

Chandramouli, N. A., Natarajan, S., Alharbi, A. H., Kannan, S., Khafaga, D. S., Raju, S. K., Eid, M. M., & El-kenawy, E. S. M. (2024). Enhanced human activity recognition in medical emergencies using a hybrid deep cnn and bi-directional lstm model with wearable sensors. *Scientific Reports*, *14*(1), 1–24. https://doi.org/10.1038/s41598-024-82045-y

Chen, S. Y., & Lin, C. L. (2024). Wi-fi-based human activity recognition for continuous, whole-room monitoring of motor functions in parkinson's disease. *IEEE Open Journal of Antennas and Propagation*, *5*(3), 788–799. https://doi.org/10.1109/OJAP.2024.3393117

Dilshad Ansari, M., Benhaili, Z., Visutsak, P., Bangkok, N., Ashaq Hussain Bhat, T., Noori, F. M., Deeptha, R., Ramkumar, K., Venkateswaran, S., Mehedi Hassan, M., Rafiul Hassan, M., & Zia Uddin, M. (2024). Enhancing human activity recognition for the elderly and individuals with disabilities through optimized internet-of-things and artificial intelligence integration with advanced neural networks. *Frontiers in Neuroinformatics*, *18*.

Guerra, B. M. V., Torti, E., Marenzi, E., Schmid, M., Ramat, S., Leporati, F., & Danese, G. (2023). Ambient assisted living for frail people through human activity recognition: State-of-the-art, challenges and future directions. *Frontiers in Neuroscience*, *17*, 1256682. https://doi.org/10.3389/FNINS.2023.1256682

Hirawat, A., Taterh, S., & Sharma, T. K. (2022). A dynamic window-size based segmentation technique to detect driver entry and exit from a car. *Journal of King Saud University - Computer and Information Sciences*, *34*(10), 8514–8522. https://doi.org/10.1016/J.JKSUCI.2021.08.028

Host, K., & Ivašić-Kos, M. (2022). An overview of human action recognition in sports based on computer vision. *Heliyon*, *8*(6), e09633. https://doi.org/10.1016/J.HELIYON.2022.E09633

Jaén-Vargas, M., Leiva, K. M. R., Fernandes, F., Goncalves, S. B., Silva, M. T., Lopes, D. S., & Olmedo, J. J. S. (2022). Effects of sliding window variation in the performance of acceleration-based human activity recognition using deep learning models. *PeerJ Computer Science*, *8*, e1052. https://doi.org/10.7717/PEERJ-CS.1052

Jeon, S., Lee, Y. S., & Son, S. H. (2023). Cascade windows-based multi-stream convolutional neural networks framework for early detecting in-sleep stroke using wristbands. *IEEE Access*, *11*, 84944–84956. https://doi.org/10.1109/ACCESS.2023.3301872

Karagiannaki, K., Panousopoulou, A., & Tsakalides, P. (2016). A benchmark study on feature selection for human activity recognition. *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 105–108. https://doi.org/10.1145/2968219.2971421

Liandana, M., Hostiadi, D. P., Hendrawan, N. R., Pradipta, G. A., Desiana, P., & Ayu, W. (2025). Enhanced human activity recognition (har): Leveraging sub-window techniques and feature ratios from triaxial accelerometer data. *International Journal of Intelligent Engineering and Systems*, *18*(1). https://doi.org/10.22266/ijies2025.0229.34

Liandana, M., Hostiadi, D. P., & Pradipta, G. A. (2024). A new approach for human activity recognition (har) using a single tri-axial accelerometer based on a combination of three feature subsets. *International Journal of Intelligent Engineering and Systems*, *17*(2), 235–250. https://doi.org/10.22266/ijies2024.0430.21

Lu, N., Yan, T., Zhu, S., Qian, J., & Han, M. (2025). Deep feature unsupervised domain adaptation for time-series classification. *IEEE Transactions on Artificial Intelligence*, *6*(3), 725–737. https://doi.org/10.1109/TAI.2024.3491948

Ma, C., Li, W., Cao, J., Du, J., Li, Q., & Gravina, R. (2020). Adaptive sliding window based activity recognition for assisted livings. *Information Fusion*, *53*, 55–65. https://doi.org/10.1016/J.INFFUS.2019.06.013

Machado, J., Antosz, K., Mazurkiewicz, D., Ren, Y., Rea, P., El Abdi, R., Ranga, M., Kumar Manupati, V., Villani, E., & Park, K. (2022). Wearable sensor for forearm motion detection using a carbon-based conductive layer-polymer composite film. *Sensors*, *22*(6), 2236. https://doi.org/10.3390/S22062236

Mekruksavanich, S., & Jitpattanakul, A. (2022). Deep residual network for smartwatch-based user identification through complex hand movements. *Sensors*, *22*(8), 3094. https://doi.org/10.3390/S22083094

Minh Dang, L., Min, K., Wang, H., Jalil Piran, M., Hee Lee, C., & Moon, H. (2020). Sensor-based and vision-based human activity recognition: A comprehensive survey. *Pattern Recognition*, *108*, 107561. https://doi.org/10.1016/J.PATCOG.2020.107561

Munoz-Organero, M., Luptáková, I. D., Kubovčík, M., & Pospíchal, J. (2022). Wearable sensor-based human activity recognition with transformer model. *Sensors*, *22*(5), 1911. https://doi.org/10.3390/S22051911

Pan, J., Hu, Z., Zhang, L., & Cai, X. (2025). Multi-channel time series decomposition network for generalizable sensor-based activity recognition. *IEEE Transactions on Automation Science and Engineering*, *22*, 8150–8161. https://doi.org/10.1109/TASE.2024.3480119

Pathan, M. S., Nag, A., Pathan, M. M., & Dev, S. (2022). Analyzing the impact of feature selection on the accuracy of heart disease prediction. *Healthcare Analytics*, *2*, 100060. https://doi.org/10.1016/J.HEALTH.2022.100060

Petz, P., Eibensteiner, F., & Langer, J. (2021). Sensor shirt as universal platform for real-time monitoring of posture and movements for occupational health and ergonomics. *Procedia Computer Science*, *180*, 200–207. https://doi.org/10.1016/J.PROCS.2021.01.157

Qian, H., Pan, S. J., & Miao, C. (2021). Weakly-supervised sensor-based activity segmentation and recognition via learning from distributions. *Artificial Intelligence*, *292*, 103429. https://doi.org/10.1016/J.ARTINT.2020.103429

Rahayu, Y., Rosdiansyah, Hilmi, M. F., & Odih, T. (2021). Wearable antenna for time-domain breast tumor detection. *International Journal of Technology*, *12*(6), 1101–1111. https://doi.org/10.14716/IJTECH.V12I6.5187

Reiss, A., & Stricker, D. (2012). Introducing a new benchmarked dataset for activity monitoring. *Proceedings - International Symposium on Wearable Computers, ISWC*, 108–109. https://doi.org/10.1109/ISWC.2012.13

Reyes-Ortiz, J. L., Oneto, L., Samà, A., Parra, X., & Anguita, D. (2016). Transition-aware human activity recognition using smartphones. *Neurocomputing*, *171*, 754–767. https://doi.org/10.1016/J.NEUCOM.2015.07.085

Rustam, F., Reshi, A. A., Ashraf, I., Mehmood, A., Ullah, S., Khan, D. M., & Choi, G. S. (2020). Sensor-based human activity recognition using deep stacked multilayered perceptron model. *IEEE Access*, *8*, 218898–218910. https://doi.org/10.1109/ACCESS.2020.3041822

Shi, W., Fang, X., Yang, G., & Huang, J. (2022). Human activity recognition based on multi-channel convolutional neural network with data augmentation. *IEEE Access*, *10*, 76596–76606. https://doi.org/10.1109/ACCESS.2022.3192452

Sun, L., Yang, X., & Hu, C. (2022). Dswhar: A dynamic sliding window based human activity recognition method. *Proceedings - 2022 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Autonomous and Trusted Vehicles, Scalable Computing and Communications, Digital Twin, Privacy Computing, Metaverse*, 1421–1426. https://doi.org/10.1109/SMARTWORLD-UIC-ATC-SCALCOM-DIGITALTWIN-PRICOMP-METAVERSE56740.2022.00205

Vidya, B., & Sasikumar, P. (2022). Wearable multi-sensor data fusion approach for human activity recognition using machine learning algorithms. *Sensors and Actuators A: Physical*, *341*, 113557. https://doi.org/10.1016/J.SNA.2022.113557

Wang, H., Zhao, J., Li, J., Tian, L., Tu, P., Cao, T., An, Y., Wang, K., & Li, S. (2020). Wearable sensor-based human activity recognition using hybrid deep learning techniques. *Security and Communication Networks*, *2020*. https://doi.org/10.1155/2020/2132138

Wang, J., Xu, P., Ji, X., Li, M., & Lu, W. (2023a). Feature selection in machine learning for perovskite materials design and discovery. *Materials*, *16*(8), 3134. https://doi.org/10.3390/MA16083134

Wang, K., Saragadam, A., Kaur, J., Dogra, A., Cao, S., Ghafurian, M., Butt, Z. A., Abhari, S., Chumachenko, D., & Morita, P. P. (2025). A contactless method for recognition of daily living activities for older adults based on ambient assisted living technology. *Internet of Things*, *30*, 101502. https://doi.org/10.1016/J.IOT.2025.101502

Wang, L., Jiang, S., & Jiang, S. (2021). A feature selection method via analysis of relevance, redundancy, and interaction. *Expert Systems with Applications*, *183*, 115365. https://doi.org/10.1016/J.ESWA.2021.115365

Wang, Y., Xu, H., Liu, Y., Wang, M., Wang, Y., Yang, Y., Zhou, S., Zeng, J., Xu, J., Li, S., & Li, J. (2023b). A novel deep multifeature extraction framework based on attention mechanism using wearable sensor data for human activity recognition. *IEEE Sensors Journal*, *23*(7), 7188–7198. https://doi.org/10.1109/JSEN.2023.3242603

Wang, Y. H., Zhang, Y. F., Zhang, Y., Gu, Z. F., Zhang, Z. Y., Lin, H., & Deng, K. J. (2022). Identification of adaptor proteins using the anova feature selection technique. *Methods*, *208*, 42–47. https://doi.org/10.1016/J.YMETH.2022.10.008

WISDM Lab. (n.d.). Dataset [Retrieved September 21, 2024, from https://www.cis.fordham.edu/wisdm/dataset.php].

Yu, X., & Al-Qaness, M. A. A. (2025). Human activity recognition using deep residual convolutional network based on wearable sensors. *IEEE Journal of Biomedical and Health Informatics*, *29*(3), 1950–1958. https://doi.org/10.1109/JBHI.2024.3510860

Zhang, C., Cao, K., Lu, L., & Deng, T. (2022). A multi-scale feature extraction fusion model for human activity recognition. *Scientific Reports*, *12*(1), 1–13. https://doi.org/10.1038/s41598-022-24887-y