Received: September 30, 2024. Revised: November 12, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Enhanced Human Activity Recognition (HAR): Leveraging Sub-Window Techniques and Feature Ratios from Triaxial Accelerometer Data

Made Liandana[1]* **Dandy Pramana Hostiadi[2]** **Nyoman Rudy Hendrawan[1]**
Gede Angga Pradipta[2] **Putu Desiana Wulaning Ayu[2]**

[1]*Department of Informatics and Computer, Institut Teknologi dan Bisnis STIKOM Bali, Bali, Indonesia*
[2]*Department of Magister Information Systems, Institut Teknologi dan Bisnis STIKOM Bali, Bali, Indonesia*
* Corresponding author's Email: liandana@stikom-bali.ac.id

Abstract: Sensor-based Human Activity Recognition has advanced rapidly, incorporating techniques such as sliding windows, feature extraction, and machine-learning parameter optimization for activity classification. Sliding windows and feature extraction are vital, as they occur early and significantly influence the classification outcomes. The proposed novel approach involves using feature ratios from three sub-windows and a static window combined with feature selection via Analysis of Variance (ANOVA) and machine learning algorithms such as Artificial Neural Networks (ANN) and Extreme Gradient Boosting (XGBoost). The primary objective of the proposed approach is to enhance the efficacy of machine-learning algorithms for recognizing human activities. The effectiveness of the proposed approach is evaluated using three datasets: FORTH-TRACE, SBHARPT, and WISDM. The experimental results indicate that the highest accuracy, precision, recall, and F1 score were achieved on the WISDM dataset, with values of 97.64%, 97.64%, 97.64%, and 97.64%, respectively, using 45 features and an Artificial Neural Network (ANN) classifier. The experiments demonstrated that an overlapping window of 25% enhanced the performance of the machine-learning model.

Keywords: Human-activity-recognition, Sliding-window, Sub-window, Feature ratio.

1. Introduction

Human Activity Recognition (HAR) has been widely applied in various fields, such as healthcare [1, 2], sports [3, 4], elderly monitoring [5], and smart homes [6, 7]. Human Activity Recognition (HAR) can be performed using sensors [8], cameras [9], or a combination of both [10]. These devices collect data representing various activities, such as walking, running, sitting, sleeping, and other daily routines, which are then analyzed and classified using machine learning algorithms. The use of cameras has limitations when dealing with moving objects because they require a broader range and a larger number of cameras [2, 11]. The use of sensor devices for activity recognition is commonly referred to as wearable sensors because these devices are worn by users on specific parts of the body, such as the chest, waist, upper and lower limbs, pockets, shoes, or attached directly to the skin [12, 13]. Wearable sensors can be an effective option for recognizing the activities of moving objects, because the devices are directly attached to the user. Sensors such as accelerometers, gyroscopes, and magnetometers can be used for this process [8, 12, 14-16]. The raw data from these sensors consist of numerical sequences, making the preprocessing lighter than image or video data.

Wearable devices used by users continuously generate a series of data. From this data stream, a specific portion must be extracted and converted into features for classification using machine-learning algorithms. The process of selecting a portion of the data is known as segmentation. Several techniques are available for segmentation, and one of the most popular is the sliding window [8]. The sliding window plays a crucial role in determining the classification results of machine-learning algorithms

[17]. This was performed prior to the feature extraction and classification stages. The window size is a critical parameter in the sliding window method. A window that is too large may capture multiple activities, whereas a window that is too small may cause the data of an activity to be fragmented [17]. Both static and dynamic window sizing approaches have been previously developed [17, 18]. However, determining the optimal sliding window size and the best way to extract features from the window to achieve improved machine-learning classification performance remains a significant challenge.

The proposed approach optimizes feature extraction, the number of features, window size, machine learning models, Artificial Neural Network (ANN), and Extreme Gradient Boosting (XGBoost). The sensor data used were sourced from a single accelerometer sensor, with the rationale of using only one accelerometer sensor to reduce processing complexity. Before the feature extraction process, the initial step involved dividing the window into sub-windows. Features were extracted from each window using statistical approaches, crossing values, and ratio values. The extracted features were then selected using an Analysis of Variance (ANOVA). The selected features were used to form feature combinations, and the effectiveness of these combinations was evaluated using Extreme Gradient Boosting (XGBoost). The contributions of the proposed approach are as follows: (1) sub-windows are generated from the current sliding window process; (2) features are extracted not only from the window but also from each sub-window; (3) the relationships between sub-windows are extracted as feature ratios; (4) the extracted features are selected using Analysis of Variance (ANOVA), and feature combinations are evaluated using an Artificial Neural Network (ANN) and Extreme Gradient Boosting (XGBoost).

The remainder of this paper is organized as follows. Section 2 discusses the related work. Section 3 elucidates the proposed model, which comprises preprocessing, feature extraction, analysis of variance (ANOVA), feature selection, construction of the machine learning model, and evaluation of the model. Section 4 delineates the experiments, including the dataset description and experimental setup. Section 5 presents the results and discussion, which encompass forming feature combinations, performance of the proposed model, and comparison with previous research. Finally, Section 6 concludes the paper.

2. Related work

Previous studies have extensively developed approaches to detect sensor-based human activities [1, 4-9, 17-19]. The framework for human activity recognition comprises data collection, pre-processing, and training [20]. Segmentation is a critical sub-step in preprocessing. Data segmentation and feature extraction have been demonstrated to be crucial components that influence the performance of machine-learning models [21]. Sliding window is a technique employed in segmentation [8, 17]. There are two categories of sliding windows: fixed and adaptive [22, 23]. The primary elements of a sliding window include samples, window sizes, and features [22].

Previous research has utilized accelerometers, gyroscopes, and pressure sensors to detect human activity [17]. In this investigation, adaptive sliding windows and multilevel machine-learning algorithms were employed. The sliding window size is expanded if the first-level machine-learning algorithm detects a transition activity. A second-level machine learning algorithm was subsequently utilized to classify the overall activity. The features employed in this study included the mean, signal energy, mean trend, window mean difference, variance trend, and window variance difference. The proposed method demonstrated an improvement in the F-score by up to 15.3% compared with the static approach. A transitional activity is identified through the utilization of two hierarchical machine-learning algorithms, which may potentially impact the computational burden and result in delayed activity recognition. The adaptive sliding window proposed by Alhammad and Al-Dossari [24] applies two parameters: peak and valley boundaries. Activity data were obtained from an accelerometer worn on the wrist. The decision to expand the window size was based on the ratio of values from the x- and z-axes or the intersection of the x- and y-axes at certain peak or valley boundaries. A threshold value was utilized as a reference to identify the peaks and valleys, with all the data required to determine the threshold. The distance value was employed to avoid multiple peaks for a single activity. The features utilized included the minimum, maximum, range, mean, standard deviation, and root mean square. The proposed method achieved an accuracy of 96.67 ± 2.7% using a Support Vector Machine (SVM) machine learning algorithm. Although this approach achieves relatively high accuracy, it necessitates the determination of a threshold value as a reference for detecting valleys and peaks.

Human Activity Recognition (HAR) for detecting drivers entering and exiting a vehicle has been investigated in previous research [18]. In that study, a dynamic window size was employed, which was adjusted to the duration of the activity. Longer activity durations resulted in larger window sizes and vice versa. The methodology utilized to calculate the duration of each activity involved identifying transitions between activities from a set of labeled activity data. A total of 120 features were extracted from the acceleration, gravitation, orientation, linear acceleration, and rotational data. The features were extracted using statistical methods, including the mean, median, minimum, maximum, standard deviation, and mean absolute deviation. The machine-learning models that demonstrated the most optimal performance on the dataset included Logistic Regression, Linear Support Vector Classifier (SVC), Decision Tree, Random Forest, Gradient Boosting, and K-Nearest Neighbor (KNN), achieving an accuracy of 100%. Although the proposed approach exhibits high accuracy, the efficacy of the method requires further evaluation on datasets with different characteristics. Furthermore, the utilization of window sizes based on activity duration may result in continuous activities, potentially leading to excessively large window sizes [23].

Baraka and Mohd Noor [25] employed a static sliding window approach. Signals from the accelerometer, gyroscope, and magnetometer sensors were divided into adjacent windows, including the current and previous windows. Each window was further divided into three sub-windows. The sub-windows from a given window were utilized to calculate similarity values using the Euclidean distance, Manhattan (City Block) distance, and cosine similarity; this operation is referred to as the inner similarity. The inner similarity values from the current window and the previous window were utilized to determine the end of a basic activity (non-transition), the start of a transition activity, the start of a basic activity (non-transition), and the end of a transition activity; this operation is termed adjacent window dissimilarity. Inner similarity and adjacent-window dissimilarity operations were employed to identify whether a window contained a transition activity or a basic activity (non-transition). Windows containing transition activities were classified by a model for transition activities, whereas windows containing basic activities (non-transition) were classified by a model for basic activities. The two machine learning models utilized were Convolutional Neural Networks (CNN). The proposed model was evaluated using two datasets and achieved accuracies of 92.71% and 86.65% for the respective datasets. The proposed approach still necessitates threshold value determination, particularly in the calculation of adjacent-window dissimilarity. To address the limitations of threshold-based approaches, deep similarity segmentation has been introduced as a solution and the development of a previous approach [26]. This study focused on distinguishing between transition and basic activities using a combination of local and temporal features. The time-series data were divided into three consecutive windows, each of which was extracted using convolutional layers and max pooling, yielding local features. The local features from the windows were subsequently utilized to extract the temporal features. Based on the local features from several windows, the temporal features were extracted by measuring the similarity between adjacent windows. Overall, the proposed approach achieved an optimal accuracy performance of 93.35%. The proposed approach [25][26] requires 2–3 windows per activity prediction; thus, the processing complexity necessitates further evaluation.

The Human Activity Recognition (HAR) model proposed by Vidya [20] focuses on classifying activities such as bending, cycling, lying down, sitting, standing, and walking. Data obtained from multi-sensor accelerometers and Received Signal Strength (RSS) from Wireless Sensor Networks (WSN) were transformed into time and frequency domains utilizing Discrete Wavelet Transform (DWT) and Empirical Mode Decomposition (EMD). The resultant features generated encompassed entropy, energy, and statistical features (Mean, Mean Absolute Deviation, Median Absolute Deviation, Standard Deviation, L2 Norm). Feature selection was performed using Pearson's correlation, and classification was conducted using Support Vector Machine (SVM), K-nearest neighbor (KNN), Ensemble Classifier (EC), and Decision Tree (DT). The proposed approach demonstrates significant enhancement in machine learning performance, particularly for the Decision Tree (DT) classifier. A comparable study conducted by Geravesh et al. [27] classified daily activities as sitting, standing, walking, cycling, ascending stairs, and descending stairs. Six features were extracted from the combination of accelerometer and gyroscope sensors, sourced directly from each sensor axis. Multilayer Perceptron (MLP), K-nearest neighbor (KNN), Random Forest (RF), Decision Tree (DT), and Logistic Regression (LR) were employed to evaluate the extracted features. The optimal performance was achieved utilizing the Multilayer Perceptron (MLP) model. It is noteworthy that the studies conducted by [20] and Geravesh et al. [27] did not explicitly address the

Received: September 30, 2024. Revised: November 12, 2024.

sliding-window technique employed; however, both utilized a static approach.

Sliding windows and feature values are two inseparable elements [21]. Features were extracted

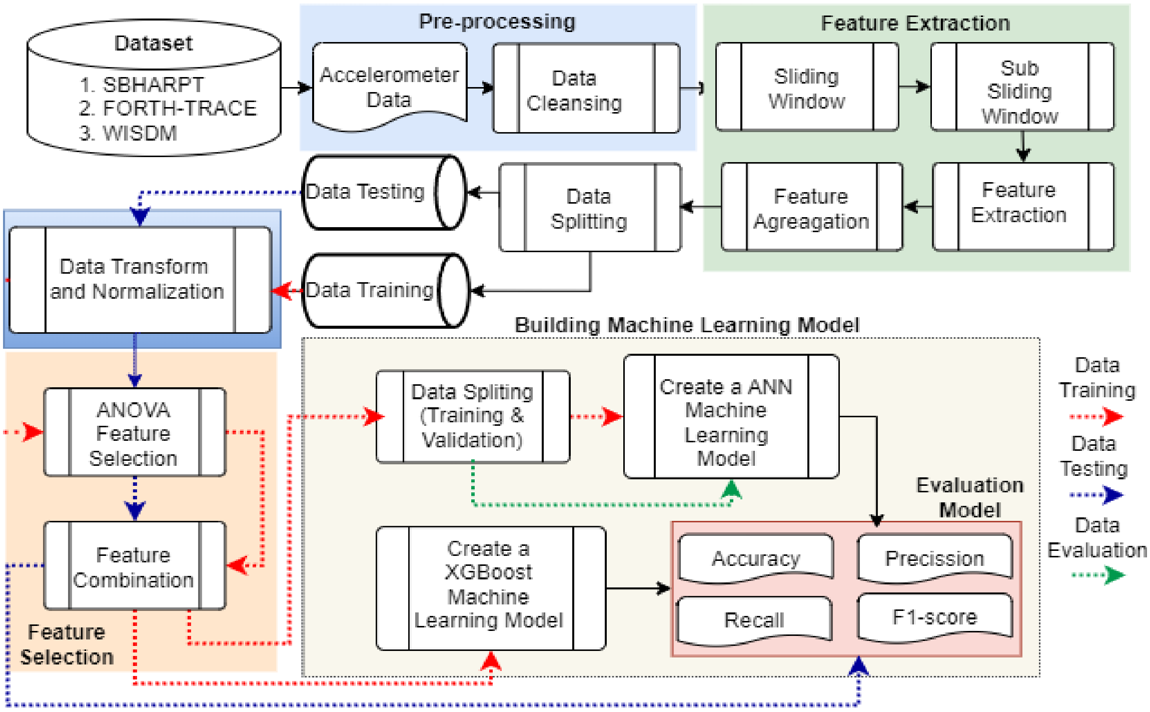


Figure. 1 Proposed Method

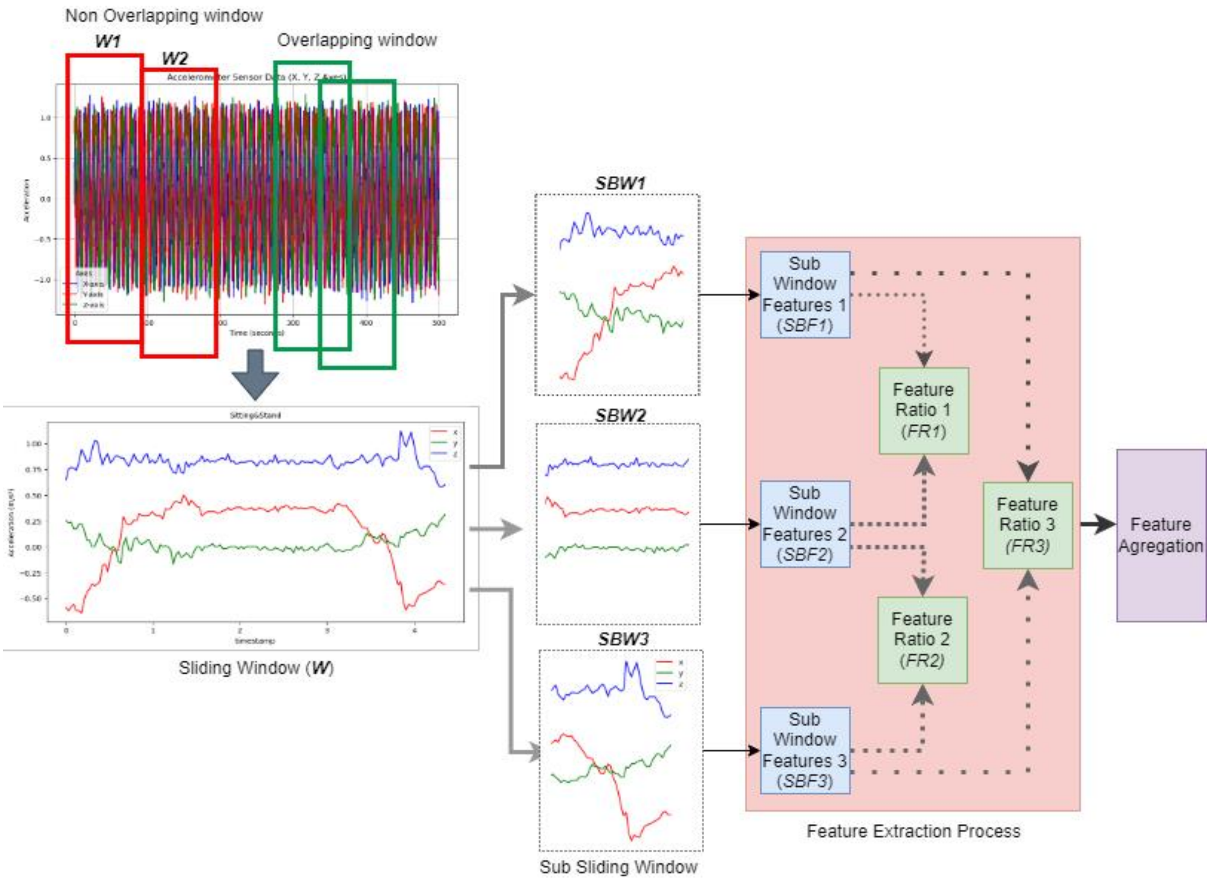


Figure. 2 Feature Extraction Process

from a window and subsequently classified using machine learning algorithms. Adaptive and static sliding windows have been proposed by several researchers [17, 18, 24-26]. The utilization of adaptive or dynamic sliding windows eliminates the necessity to determine the window size but may result in excessively large window sizes [23]. Conversely, the static approach necessitates determining a window size; however, the optimal size can be identified through the evaluation of various window sizes, as demonstrated in previous research [22]. Regarding sensor data usage, previous studies have utilized multiple sensors [17, 18, 20, 25-27], which can increase data dimensionality and potentially elevate processing complexity [20]. Previous approaches [25, 26] employed multiple windows to generate features, necessitating the extraction of more windows into the features. In the proposed approach, a single window with a static size is utilized to classify an activity. This window was divided into subwindows, and the features were subsequently extracted from each sub-window. The relationships between the subwindows were extracted as feature ratios. To reduce data dimensionality, only accelerometer sensor data were utilized. Feature selection using Analysis of Variance (ANOVA) is applied to determine the most optimal feature combinations.

3. Proposed method

This research focuses on developing a feature extraction and selection mechanism using accelerometer sensor data for human activity recognition. The proposed model consists of five main components: preprocessing, feature extraction, feature selection, building a machine learning model, and model evaluation. The proposed human activity recognition model is illustrated in Fig . 1.

3.1. Pre-processing

In this study, only a single accelerometer sensor was used; therefore, only portions of the dataset containing the accelerometer sensor data were utilized. During the data collection process, there is potential for certain data points to exhibit extreme values or missing data. Consequently, a data cleansing process is required. For excessively large values and missing data, they are replaced with the mean value of the corresponding column containing the extreme or missing values.

3.2. Feature extraction

Accelerometer sensor data may consist of multiple activities. To obtain data representing a specific activity, a sliding window process was applied using a static window. Two types of windows are used: non-overlapping and overlapping windows. An illustration of the sliding window process and the division of windows into subwindows is shown in Fig. 2. Each window representing an activity is divided into three subwindows. Each sub-window is then extracted into feature values using Eqs. (1) to (16). Table 1 provides a comprehensive breakdown of the equation, delineating its constituent elements numbered from 1 to 26.

$$min = min\left(sbw_j\right) \tag{1}$$

$$max = max\left(sbw_j\right) \tag{2}$$

$$\overline{sbw} = \frac{1}{n}\sum_{j=1}^{n}\left(sbw_j\right) \tag{3}$$

$$med = \begin{cases} sbw_{\left[\frac{n+1}{2}\right]}, if\ n\ odd \\ \frac{1}{2}\left(sbw_{\left[\frac{n}{2}\right]} + sbw_{\left[\frac{n+1}{2}\right]}\right), if\ n\ even \end{cases} \tag{4}$$

$$std = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(sbw_j - \overline{sbw}\right)^2} \tag{5}$$

$$iqr = q_3 - q_1 \tag{6}$$

3.3. Analysis of Variance (ANOVA) feature selection

Data dimensionality is a critical factor in classical machine learning [28],[29]. Feature selection can enhance machine-learning performance by reducing irrelevant and redundant data while minimizing computational resource requirements [28, 30, 31]. Numerous studies have demonstrated that feature selection utilizing Analysis of Variance (ANOVA) significantly improves machine learning performance [32-36]. Analysis of Variance (ANOVA) is a statistical method that compares the variance between groups and within groups, as expressed in Eq. (17) [19]. A description of the notation in the equation is presented in Table 1, covering entries 27 to 38.

$$F_{value} = \frac{s_{bg}^2}{s_{wg}^2} \tag{17}$$

$$\overline{Ac}_u = \frac{1}{n_u}\sum_{v=1}^{n_u} g_{uv} \quad (18)$$

$$\overline{Ac} = \frac{1}{K}\sum_{m=1}^{K} G_m$$
$$g_{uv} \in G_m \quad (19)$$

$$s_{bg}^2 = \frac{1}{\mathrm{k}-1}\sum_{i=1}^{k} n_i\left(\overline{Ac}_i - \overline{Ac}\right)^2 \quad (20)$$

$$s_{wg}^2 = \frac{1}{K-k}\sum_{i=1}^{k}\sum_{j=1}^{n_i}\left(g_{ij} - \overline{Ac}_i\right)^2 \quad (21)$$

3.4. Bulding model machine leaning and evaluation model

The extracted feature data are used to build machine learning models, namely an Artificial Neural Network (ANN) and Extreme Gradient Boosting (XGBoost). The resulting machine learning models were evaluated using four metrics: accuracy (acc), precision ($prec$), recall (rec), and F1 score ($F_{1\,score}$), using Eq. (22) through (25). These metrics are commonly used to evaluate various Human Activity Recognition (HAR) applications and other related applications [20, 42]. Accuracy measures the number of correct predictions (both positive and negative) out of the total number of predictions produced by the model. Precision is used to measure the accuracy of positive predictions, specifically the percentage of samples that are truly positive out of all the samples predicted as positive. A high precision indicates that the model rarely misclassifies negative samples as positive. Recall measures the model's ability to detect positive samples, defined as the percentage of truly positive samples out of all the actual positive samples. A high recall signifies that the model rarely misses the actual positive samples. The F1 score is the harmonic mean of the precision and recall, balancing the two metrics. This score is particularly useful when working with unbalanced data. True Positives (TP) are samples that are correctly predicted. True Negatives (TN) are negative samples that are correctly predicted. False Positives (FP) refer to samples predicted as positive but are actually negative. False Negatives (FN) refer to samples predicted as negative but are actually positive.

$$acc = \frac{TP+TN}{TP+TN+FP+FN} \quad (22)$$

$$prec = \frac{TP}{TP+FP} \quad (23)$$

$$rec = \frac{TP}{TP+FN} \quad (24)$$

$$F_{1\,score} = 2 \times \frac{prec \times rec}{prec + rec} \quad (25)$$

$$\overline{cmc} = \begin{cases} \sum_{i=1}^{n} f(sbw)_i \\ f(sbw)_i = \begin{cases} 1, if \;\; sbw_{j-1} > \overline{sbw} \;\; > sbw_j \\ 1, if \;\; sbw_{j-1} < \;\; \overline{sbw} \;\; < sbw_j \\ 0, else \end{cases} \end{cases} \quad (7)$$

$$\overline{smc} = \begin{cases} \sum_{i=1}^{n} f(sbw)_i \\ f(sbw)_i = \begin{cases} sbw_{j-1} + sbw_j, if \;\; sbw_{j-1} > \overline{sbw} \;\; > sbw_j \\ sbw_{i-1} + sbw_j, if \;\; sbw_{j-1} < \;\; \overline{sbw} \;\; < sbw_j \\ 0, else \end{cases} \end{cases} \quad (8)$$

$$cpk = \begin{cases} \sum_{j=1}^{n} f(sbw)_i \\ f(sbw)_i = \begin{cases} 1, if \;\; sbw_{j-1} \; < sbw_j \;\; > sbw_{j+1} \\ 0, else \end{cases} \end{cases} \quad (9)$$

$$spk = \begin{cases} \sum_{i=1}^{n} f(sbw)_i \\ f(sbw)_i = \begin{cases} sbw_j, if \ \ sbw_{j-1} \ < \ sbw_j \ \ > \ sbw_{j+1} \\ 0, else \end{cases} \end{cases} \tag{10}$$

$$cva = \begin{cases} \sum_{i=1}^{n} f(sbw)_i \\ f(sbw)_i = \begin{cases} 1, if \ \ sbw_{j-1} > \ sbw_j \ < \ sbw_{j+1} \\ 0, else \end{cases} \end{cases} \tag{11}$$

$$sva = \begin{cases} \sum_{i=1}^{n} f(sbw)_i \\ f(sbw)_i = \begin{cases} sbw_j, if \ \ sbw_{j-1} > \ sbw_j \ < \ sbw_{j+1} \\ 0, else \end{cases} \end{cases} \tag{12}$$

$$min_{cros}^{c} = \begin{cases} \begin{cases} sbw_j^s + \ sbw_{j-1}^s \ , if \ \ sbw_j^s > \ \min{(sbw_j^r)} \ \ > \ sbw_{j-1}^s \\ sbw_j^s + \ sbw_{j-1}^s, if \ \ sbw_j^s < \ \ \min{(sbw_j^1)} \ < \ sbw_{j-1}^s \\ 0, else \end{cases} \\ combinations = \begin{cases} r = 1 \ and \ s = 2 \\ r = 1 \ and \ s = 3 \\ r = \ 2 \ and \ \ s = 3 \end{cases} \end{cases} \tag{13}$$

$$max_{ratio}^{c} = \begin{cases} \frac{\max(sbw_j^r)}{\max(sbw_j^s) + \max(sbw_j^t)} \\ combinations = \begin{cases} r = 1, s = 2 \ and \ t = 3 \\ r = 2, s = 1 \ and \ t = 3 \\ r = 3, s = 1 \ and \ t = 2 \end{cases} \end{cases} \tag{14}$$

$$mean_{ratio}^{c} = \begin{cases} \frac{mean(sbw_j^r)}{\mathrm{mean}(sbw_j^s) + \mathrm{mean}(sbw_j^t)} \\ combinations = \begin{cases} r = 1, s = 2 \ and \ t = 3 \\ r = 2, s = 1 \ and \ t = 3 \\ r = 3, s = 1 \ and \ t = 2 \end{cases} \end{cases} \tag{15}$$

$$iqr_{ratio}^{c} = \begin{cases} \frac{iqr(sbw_j^r)}{iqr(sbw_j^s) + \mathrm{iqr}(sbw_j^t)} \\ combinations = \begin{cases} r = 1, s = 2 \ and \ t = 3 \\ r = 2, s = 1 \ and \ t = 3 \\ r = 3, s = 1 \ and \ t = 2 \end{cases} \end{cases} \tag{16}$$

Received: September 30, 2024. Revised: November 12, 2024.

Table 1. List of notation description

No	Notation	Description
1	min	minimum
2	max	maximum
3	$\overline{sbw}$	mean
4	med	median
5	std	standard deviation
6	iqr	interquartil
7	sbw_j	the series of acceleration data within a sub-window
8	j	the index of each data point
9	q_1	the lower quartile or first quartile
10	q_3	the upper or third quartile
11	$\overline{cmc}$	count-mean-crossing
12	$\overline{smc}$	sum-mean-crossing
13	cpk	count-peak
14	spk	sum-peak
15	cva	count-valley
16	sva	sum-valley
17	min_{cros}^{c}	the min-crossing for each sub-window. This notation represents the intersection between the minimum value (min) and the acceleration values of other sub-windows.
18	max_{ratio}^{c}	the ratio between the maximum value of a sub-window and the sum of the maximum values of the other two sub-windows.
19	$mean_{ratio}^{c}$	the ratio between the mean value of a sub-window and the sum of the mean values of the other two sub-windows
20	iqr_{ratio}^{c}	the ratio between the interquartile range of a sub-window and the sum of the interquartile ranges of the other two sub-windows
21	sbw_j^1, sbw_j^2, sbw_j^3	the first sub-window, the second sub-window, and the third sub-window
22	c	the index of the sub-window combinations, with values of 1, 2, and 3.
23	r *and* s	the three sub-windows used, namely: the first sub-window, the second sub-window, and the third sub-window, denoted as sbw_j^1, sbw_j^2, sbw_j^3
24	$f(sbw)_i$	the function that represents the return value
25	i	the index of the function that represents the return value
25	n	n is the length or the total number of elements in the function $f(sbw)_i$
27	F_{value}	the Analysis of Variance (ANOVA) **value (**the ratio of the variance between groups to the variance within groups)
28	s_{bg}^2	the variance between groups
29	s_{wg}^2	the variance within groups
30	$\overline{Ac_u}$	the mean of each group
31	u	the index of each group, u ranges from 1 to k
32	$\overline{Ac}$	the overall mean of all samples
33	n_u	the length of each group
34	g_{uv}	the sample data from each group
35	v	the index of each value within each group
36	G_m	refers to all sample data
37	k	the number of groups formed from all sample data
38	K	the total number of samples

4. Experiments description

4.1. Dataset description

This study uses three datasets: SBHARPT [37, 38], FORTH-TRACE [39], and WISDM [40]. These datasets are used to test and evaluate the effectiveness of the proposed approach. The SBHARPT dataset was collected using accelerometer and gyroscope sensors embedded in a smartphone. The smartphone was worn on the waist of the user at a sampling rate of 50 Hz. Data were collected from 30 subjects aged 19–48 years. Twelve activities were recorded,

Received: September 30, 2024. Revised: November 12, 2024.

Table 2. Summary of the Architecture & Parameters for the Artificial Neural Network (ANN)

Parameter	Value
Input Features	depends on the feature combination
Output Units	depends on data
Activation Functions	ReLU (hidden layers), Softmax (output layer)
Dropout Rate	0.1 (10%) after each dense layer
Optimizer	Adam (lr=0.00155, beta_1=0.9, beta_2=0.999, decay=0.00014)
Loss Function	Categorical Crossentropy
Metrics	Accuracy
Epochs	100
Validation Split	0.2 (20% of training data)
Early Stopping	Based on validation accuracy, patience=100 epochs
Model Checkpoint	Saves the best model based on validation accuracy

including three static postures (standing, sitting, and lying), three dynamic activities (walking, walking downstairs, and walking upstairs), and six postural transitions (stand-to-sit, sit-to-stand, sit-to-lie, lie-to-sit, stand-to-lie, and lie-to-stand).

The FORTH-TRACE dataset was collected from 15 subjects using an accelerometer, gyroscope, and magnetometer sensors with a fixed sampling frequency of 51.2 Hz. The subjects performed 16 types of activities, consisting of seven basic activities and nine postural transition activities. Accelerometer and gyroscope sensors are unable to detect speaking activities because these sensors only capture physical movements [41]. Therefore, activities involving speaking such as walking while talking were excluded from the experiments. This dataset also includes postural transition activities that occur between dynamic activities, such as climbing stairs and walking. Previous studies have shown that such activities can lead to classification errors [25]. Consequently, the selected activities in the FORTH-TRACE dataset were standing, sitting, walking, climbing stairs, standing to sitting, and sitting to standing.

The WISDM dataset was recorded by the Wireless Sensor Data Mining Lab and serves as a publicly available benchmark dataset for Human Activity Recognition (HAR) [40]. This dataset was gathered by performing a series of specific daily activities with 36 subjects. Participants placed an Android phone in their front pocket and engaged in various activities, including sitting, jogging, climbing stairs, descending stairs, standing, and walking for a set duration. An integrated 3-axis accelerometer (x, y, and z) was used to measure the changes in linear acceleration, providing important information about human movement and activity patterns every 50 ms.

4.2. Experimental setup

The FORTH-TRACE and SBHARPT datasets consist of accelerometers, gyroscopes, and magnetometer sensors. However, in the proposed model, only the data were used. Therefore, the data were filtered to include only those from the accelerometer sensor. However, all the data in the WISDM dataset come from the accelerometer sensor, but some values need to be cleaned, such as those that could not be defined as numbers and excessively large values (infinity). These invalid values were cleaned and imputation was performed using the mean value of each column. The training and testing data were split in a ratio of 80:20, with 80% used for training and 20% for testing. In this study, the Keras library was used for the Artificial Neural Network (ANN) classifier, which consists of six hidden layers with a decreasing number of neurons: 512, 256, 128, 64, 32, and 16. The details of the ANN architecture used are shown in Table 1. For the Extreme Gradient Boosting (XGBoost) classifier, the Scikit-Learn framework was used with the function XGBClassifier() and the parameters n_estimators=100 and eval_metric='mlogloss.' Both classifiers were implemented using the Python programming language.

5. Result and discussion

5.1. Forming feature combinations

The features generated from the feature-extraction process resulted in a dimensionality of 144. Based on these features, those that could significantly impact the machine learning model were selected using Analysis of Variance (ANOVA) feature selection. In this study, each feature was ranked based on the Analysis of Variance (ANOVA) feature selection calculation. Feature combinations were then formed. For instance, the first feature combination is taken from the feature with the highest rank; the second combination consists of the first- and second-ranked features; and the third combination includes the first-, second-, and third-ranked features. In total, there were 144 feature combinations, as listed in Table 2. Each feature combination is then evaluated using Artificial Neural Network (ANN) and Extreme Gradient Boosting (XGBoost) classifiers.

5.2. Performance of the proposed model

Each dataset consisted of six experiments, with each experiment based on the window length. Three experiments were conducted with window sizes of 60, 90, and 120 and three experiments with overlapping windows. These overlapping windows are denoted as 60:45, 90:67, and 120:90, respectively. This notation represents a 25% overlap. For example, in the notation 60:45, number 60 indicates that the current window is taken from sample points 1 to 60, while the next window starts from sample point 45 to sample point 105. The overlap sizes can be denoted using Eqs. (26) to (28), where W_{start} represents the starting sample point of a window, Wl_{i-1} is the size of the previous window, W_{end} Wend is the ending sample point of the window, and k_w represents the window sizes to be evaluated (60, 90, 120).

$$W_{start} = Wl_{i-1} - (0.25 \times Wl_{i-1}) \quad (26)$$

$$W_{end} = W_{start} + k_w \quad (27)$$

$$Wl = W_{end} - W_{start} \quad (28)$$

Fig. 3 shows the accuracy of each experiment in relation to the number of features. The results indicate that the number of features affects the performance of machine learning algorithms, Artificial Neural Network (ANN), and Extreme Gradient Boosting (XGBoost). Overall, the accuracy increased when the number of features exceeded 20. In the FORTH-TRACE dataset, the optimal performance of the Artificial Neural Network (ANN) and Extreme Gradient Boosting (XGBoost) algorithms is achieved using an overlapping window with a size of 120:90. For the SBHARPT dataset, the optimal performance for both machine-learning algorithms was obtained with an overlapping window of size 90:67. Meanwhile, for the WISDM dataset, the optimal performance for both machine learning algorithms was achieved with an overlapping window size of 120:90.

Tables 4 to 6 present the optimal accuracy, precision, recall, and F1-score values for each dataset and classifier. For the FORTH-TRACE dataset, the accuracy, precision, recall, and F1-score for the Artificial Neural Network (ANN) classifier were 94.54%, 94.5%, 94.54%, and 94.47%, respectively. For the SBHARPT dataset, the accuracy, precision, recall, and F1-score were 93.65%, 93.30%, 93.65%, and 93.40%, respectively. For the WISDM dataset, the accuracy, precision, recall, and F1-score were all 97.64%. For the Extreme Gradient Boosting (XGBoost) classifier, the accuracy, precision, recall, and F1-score are 93.82%, 93.85%, 93.82%, and 93.72% for the FORTH-TRACE dataset; 93.12%, 93.23%, 93.12%, and 93.06% for the SBHARPT dataset; and 96.26%, 96.22%, 96.26%, and 96.17% for the WISDM dataset.

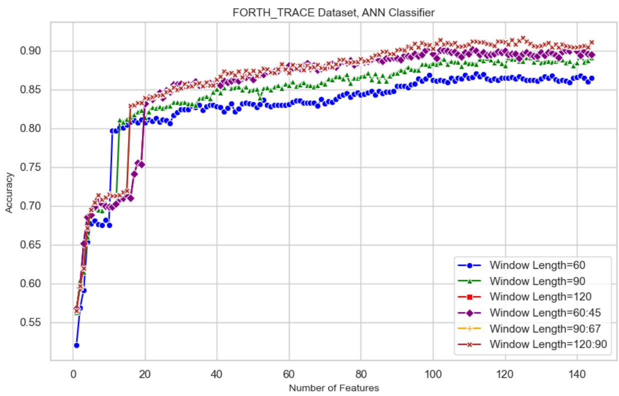
Fig. 4 shows a comparison of the number of features and accuracy for each experiment. For the FORTH-TRACE dataset, the optimal performance of the Artificial Neural Network (ANN) was achieved with 130 features, whereas the optimal performance of Extreme Gradient Boosting (XGBoost) was achieved with 139 features. For the SBHARPT dataset, the optimal performance of the Artificial Neural Network (ANN) was reached with 74 features, and for Extreme Gradient Boosting (XGBoost) with 120 features. In the experiments using the WISDM dataset, the optimal performance for the Artificial Neural Network (ANN) was achieved with 45 features, whereas the optimal performance for Extreme Gradient Boosting (XGBoost) was achieved with 88 features. The smallest number of features

Table 3. Examples of feature combination formation

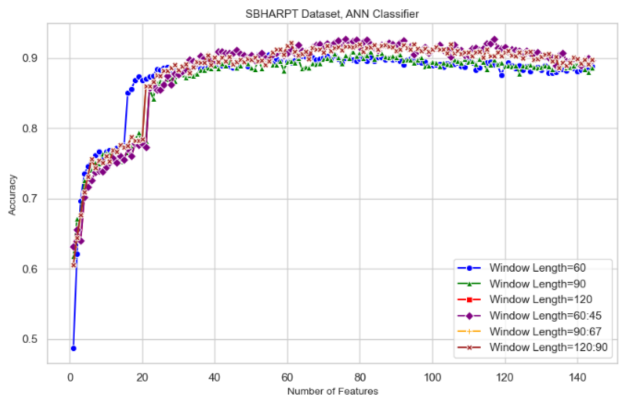
Feature index	Feature name	F value	Feature ranking	Combination of features (index view)
0	w2_y_count_cross_mean	6632.79811121	1	0
1	'w1_y_count_cross_mean	6549.3589873	2	0,1
2	w3_y_count_cross_mean	6466.77070179	3	0,1,2
…	…	…	…	…
…	…	…	…	…
143	w2_y_mean_ratio	14.97278221	144	0,1,2,…,143

Received: September 30, 2024. Revised: November 12, 2024.

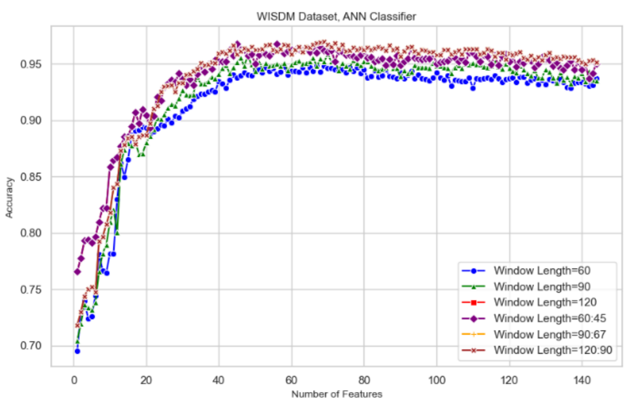
that resulted in the highest performance was achieved using the WISDM dataset and Artificial Neural Network (ANN) with 45 features and an accuracy of 97.64%. Across all experiments, the proposed model was able to improve machine learning performance, with the highest performance obtained using the WISDM dataset and Artificial Neural Network (ANN) algorithm. Moreover, overlapping windows provide better performance than non-overlapping windows.



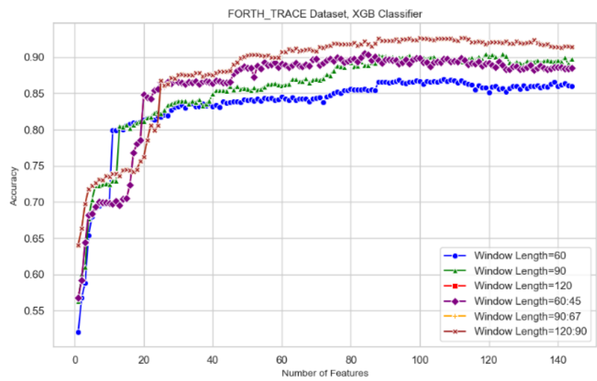
(a)



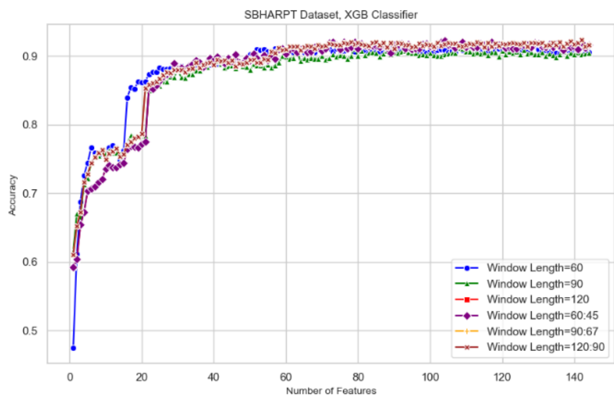
(b)



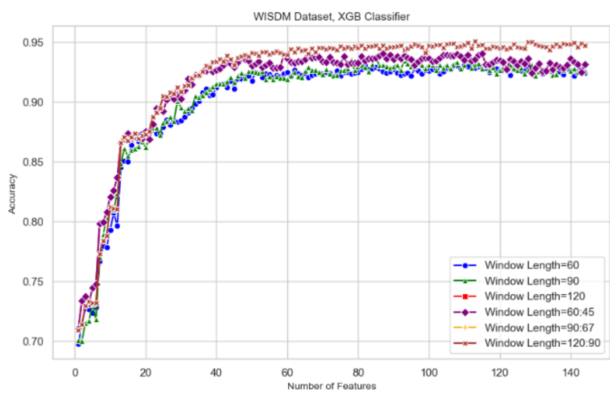
(c)



(d)



(e)



(f)

Figure. 3 Accuracy based on the number of features: (a) FORTH_TRACE Dataset, ANN Classifier, (b) SBHARPT Dataset, ANN Classifier, (c) WISDM Dataset, ANN Classifier, (d) FORTH_TRACE Dataset, XGBoost Classifier, (e) SBHARPT Dataset, XGBoost Classifier, and (f) WISDM Dataset, XGBoost Classifier

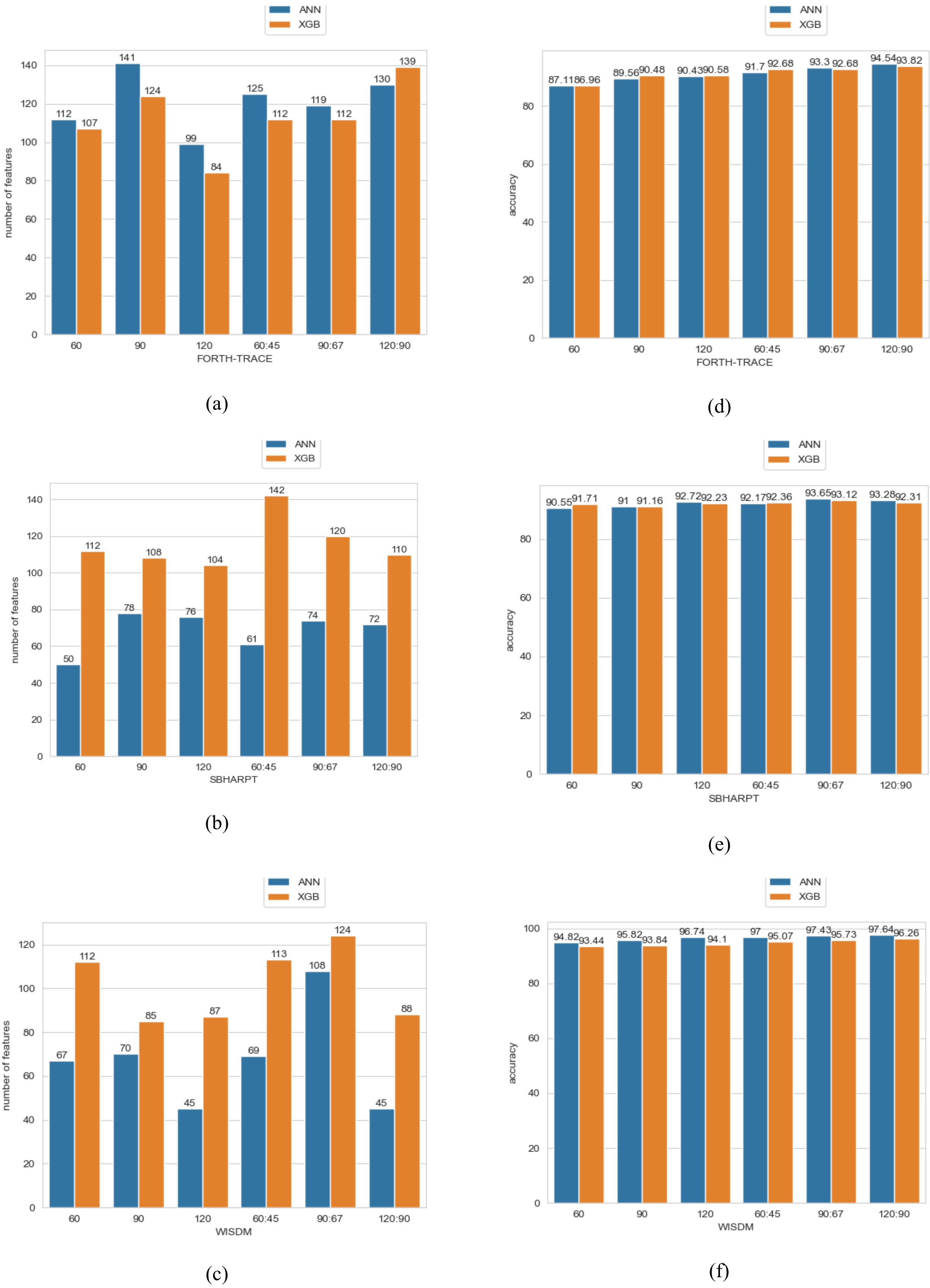


Figure. 4 Comparison of the number of features and accuracy for each experiment: (a) Optimal number of features (FORTH-TRACE), (b) Optimal number of features (SBHARPT), (c) Optimal number of features (WISDM), (d) Optimal accuracy (FORTH-TRACE), (e) Optimal accuracy (SBHARPT), and (f) Optimal accuracy (WISDM)

Received: September 30, 2024. Revised: November 12, 2024.

Table 4. Evaluation Metrics for the FORTH-TRACE Dataset

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-score(%)	The optimal number of features
Window size 60, Overlap 0					
ANN	87.11	86.77	87.11	86.68	112
XGBoost	86.96	87.38	86.96	86.35	107
Window size 90, Overlap 0					
ANN	89.56	89.24	89.56	89.30	141
XGBoost	90.48	90.40	90.48	90.11	124
Window size 120, Overlap 0					
ANN	90.43	90.30	90.43	90.10	99
XGBoost	90.58	90.72	90.60	90.26	84
Window size 60, Overlap 45					
ANN	91.70	91.37	91.70	91.40	125
XGBoost	92.68	92.78	92.68	92.48	112
Window size 90, Overlap 67					
ANN	93.30	93.14	93.30	93.20	119
XGBoost	92.68	92.77	92.68	92.48	112
Window size 120, Overlap 90					
ANN	**94.54**	**94.50**	**94.54**	**94.47**	130
XGBoost	**93.82**	**93.85**	**93.82**	**93.72**	139

Table 5. Evaluation Metrics for the SBHARPT Dataset

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-score(%)	The optimal number of features
Window size 60, Overlap 0					
ANN	90.55	90.28	90.55	90.32	50
XGBoost	91.71	91.43	91.71	91.53	112
Window size 90, Overlap 0					
ANN	91.00	91.39	91.00	90.70	78
XGBoost	91.16	91.30	91.16	91.15	108
Window size 90, Overlap 0					
ANN	92.72	92.90	92.72	92.70	76
XGBoost	92.23	92.21	92.23	92.13	104
Window size 60, Overlap 45					
ANN	92.17	92.20	92.17	92.13	61
XGBoost	92.36	92.28	92.36	92.27	142
Window size 90, Overlap 67					
ANN	**93.65**	**93.30**	**93.65**	**93.40**	74
XGBoost	**93.12**	**93.23**	**93.12**	**93.06**	120
Window size 120, Overlap 90					
ANN	93.28	93.24	93.28	93.21	72
XGBoost	92.31	92.53	92.31	92.24	110

Table 6. Evaluation Metrics for the WISDM Dataset

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-score(%)	The optimal number of features
Window size 60, Overlap 0					
ANN	94.82	94.73	94.82	94.76	67
XGBoost	93.44	93.26	93.44	93.22	112
Window size 90, Overlap 0					
ANN	95.82	95.78	95.82	95.79	70
XGBoost	93.84	93.71	93.84	93.63	85
Window size 90, Overlap 0					
ANN	96.74	96.70	96.74	96.71	45
XGBoost	94.10	92.00	94.07	93.84	87
Window size 60, Overlap 45					
ANN	97.00	97.00	97.00	97.00	69
XGBoost	95.07	95.00	95.07	95.00	113
Window size 90, Overlap 67					
ANN	97.43	97.42	97.43	97.43	108
XGBoost	95.73	95.65	95.73	95.67	124
Window size 120, Overlap 90					
ANN	**97.64**	**97.64**	**97.64**	**97.64**	45
XGBoost	**96.26**	**96.22**	**96.26**	**96.17**	88

Table 7. Comparison with Previous Research

Reference	Dataset	Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Baraka dan Mohd Noor [25]	FORTH-TRACE	static window, 2 window, 3 sub-window, similarity-based	86.65	-	-	-
	SBHARPT		92.71	-	-	-
Baraka dan Mohd Noor [26]	FORTH-TRACE	Deep similarity-based	84.96	-	-	-
	SBHARPT		93.35	-	-	-
Shi et al. [43]	WISDM	Multichannel Convolutional Neural Network	-	95.53	94.83	95.18
Akter et al. [44]	WISDM	Attention-Mechanism-Based Deep Learning Feature Combination	93.89	-	-	-
Zhang et al. [45]	WISDM	CNN-GRU	97.18	-	-	97.17
Proposed Model	FORTH-TRACE	Static window, 1 window, 3 sub-window, feature ratio	94.54	94.50	94.54	94.47
	SBHARPT		93.65	93.30	93.65	93.40
	WISDM		97.64	97.64	97.64	97.64

5.3. Comparison with previous research

To evaluate the effectiveness of the proposed method, three datasets were employed: FORTH-TRACE, SBHARPT, and WISDM, along with two machine-learning techniques: Artificial Neural Networks (ANN) and Extreme Gradient Boosting (XGBoost). The primary objective of this study was to assess the effectiveness of the proposed approach by comparing it with previous studies, particularly those utilizing the same datasets. Although previous studies have evaluated various datasets [43-45], only the performance results on the same dataset, specifically the WISDM, were used for comparison to ensure a relevant and meaningful evaluation.

As shown in Table 7, for the FORTH-TRACE dataset, the proposed model achieved an accuracy 7.89% higher than that of Baraka and Mohd Noor [25], and 9.58% higher than Baraka and Mohd Noor [26]. For the SBHARPT dataset, the proposed model had an accuracy 0.94% higher than Baraka and Mohd Noor [25] and 0.30% higher than Baraka and Mohd Noor [26]. For the WISDM dataset, the proposed model achieved precision, recall, and F1-score improvements of 2.11%, 2.81%, and 2.46%, respectively, compared to the approach used by Shi et al. [43]. The proposed approach also achieved accuracies 3.75% and 0.46% higher than those of Akter et al. [44] and Zhang et al. [45], respectively. Overall, the proposed approach was able to improve the accuracy, precision, recall, and F1-score compared with previous approaches

6. Conclusion

To optimize the window size, features, and number of features to improve the performance of Human Activity Recognition (HAR) models, the proposed model consists of a single window, three sub-windows, and feature ratios. Additionally, to identify the most influential feature combinations for the classification results, Analysis of Variance (ANOVA) feature selection and the machine learning algorithms Artificial Neural Network (ANN) and Extreme Gradient Boosting (XGBoost) were applied. The FORTH-TRACE, SBHARPT, and WISDM datasets are used to assess the effectiveness of the proposed model. The experimental results showed the highest accuracy, precision, recall, and F1-score of 97.64% with 45 features and an Artificial Neural Network (ANN) classifier. The experiments also demonstrated that overlapping windows significantly influenced the performance of the machine learning models. The limitation of the proposed approach is that it still uses a static sliding window, which requires an initial determination of the window size. Therefore, future developments could employ a dynamic sliding window, eliminating the need for predefined window sizes, particularly overlapping windows.

Conflicts of Interest

The author(s) declares no conflict of interest.

Author Contribution

Conceptualization, Liandana and Hostiadi; methodology, Liandana and Hendrawan; software, Liandana, Hostiadi, and Pradipta; validation, Liandana, Hostiadi, Ayu; formal analysis, Liandana, Hostiadi, Pradipta, and Hendrawan; investigation, Liandana, Hostiadi, Pradipta, and Ayu; resources, Liandana, Hostiadi, Hendrawan, and Pradipta; data curation, Liandana, Hostiadi, Hendrawan, Pradipta, and Ayu; writing—original draft preparation, Liandana, Hostiadi, Pradipta, and Ayu; writing—review and editing, Hostiadi, Hendrawan, and Pradipta; visualization, Liandana, Hostiadi, Hendrawan; supervision, Hostiadi, Pradipta, and Ayu; project administration, Liandana. All authors have read and approved the final manuscript.

Acknowledgments

The authors acknowledge the financial support provided by the Directorate General of Academic Affairs for Research, Technology, and Community Service (DPRM) KEMENDIKBUDRISTEK Indonesia for this research through the Program Funding (Fundamental Regular) for the 2024 Fiscal Year. This work was supported by Institut Teknologi dan Bisnis STIKOM Bali.

References

[1] G. Bhat, Y. Tuncel, S. An, H. G. Lee, and U. Y. Ogras, "An ultra-low energy human activity recognition accelerator for wearable health applications", *International Journal of ACM Transactions on Embedded Computing Systems*, Vol. 18, No. 5s, 2019.

[2] F. Serpush, M. B. Menhaj, B. Masoumi, and B. Karasfi, "Wearable Sensor-Based Human Activity Recognition in the Smart Healthcare System", *International Journal of Comput Intell Neurosci*, Vol. 2022, 2022.

[3] K. Host and M. Ivašić-Kos, "An overview of Human Action Recognition in sports based on Computer Vision", *International Journal of Heliyon*, Vol. 8, No. 6, p. e09633, 2022.

[4] M. Tabish, Z. ur R. Tanooli, and M. Shaheen, "Activity recognition framework in sports videos", *International Journal of Multimed Tools Appl*, pp. 1–23, 2021.

[5] L. Schrader, A. Vargas Toro, S. Konietzny, S. Rüping, B. Schäpers, M. Steinböck, C. Krewer, F. Müller, J. Güttler, and T. Bock, "Advanced Sensing and Human Activity Recognition in Early Intervention and Rehabilitation of Elderly People", *International Journal of Popul Ageing*, Vol. 13, No. 2, pp. 139–165, 2020.

[6] G. Chimamiwa, M. Alirezaie, F. Pecora, and A. Loutfi, "Multi-sensor dataset of human activities in a smart home environment", *International Journal of Data Brief*, Vol. 34, p. 106632, 2021.

[7] L. G. Fahad and S. F. Tahir, "Activity recognition and anomaly detection in smart homes", *International Journal of Neurocomputing*, Vol. 423, pp. 362–372, 2021.

[8] N. Tran Thi Hong, G. L. Nguyen, N. Quang Huy, D. Viet Manh, D. N. Tran, and D. T. Tran, "A low-cost real-time IoT human activity recognition system based on wearable sensor and the supervised learning algorithms", *International Journal of Measurement*, Vol. 218, p. 113231, 2023.

[9] G. A. S. Surek, L. O. Seman, S. F. Stefenon, V. C. Mariani, and L. dos S. Coelho, "Video-Based Human Activity Recognition Using Deep Learning Approaches", *International Journal of Sensors*, Vol. 23, Page 6384, Vol. 23, No. 14, p. 6384, 2023.

[10] M. Ehatisham-Ul-Haq, A. Javed, M. A. Azam, H. M. A. Malik, A. Irtaza, I. H. Lee, and M. T. Mahmood, "Robust Human Activity Recognition Using Multimodal Feature-Level Fusion", *IEEE Access*, Vol. 7, pp. 60736–60751, 2019.

[11] A. Hussain, T. Hussain, W. Ullah, and S. W. Baik, "Vision Transformer and Deep Sequence Learning for Human Activity Recognition in Surveillance Videos", *International Journal of Comput Intell Neurosci*, Vol. 2022, pp. 1–10, 2022.

[12] H. Wang, J. Zhao, J. Li, L. Tian, P. Tu, T. Cao, Y. An, K. Wang, and S. Li, "Wearable Sensor-Based Human Activity Recognition Using Hybrid Deep Learning Techniques", *International Journal of Security and Communication Networks*, Vol. 2020, 2020.

[13] X. Wang, H. Yu, S. Kold, O. Rahbek, and S. Bai, "Wearable sensors for activity monitoring and motion control: A review", *International Journal of Biomimetic Intelligence and Robotics*, Vol. 3, No. 1, p. 100089, 2023.

[14] A. A. Badawi, A. Al-Kabbany, A. Al-Kabbany, A. Al-Kabbany, and H. A. Shaban, "Sensor Type, Axis, and Position-Based Fusion and Feature Selection for Multimodal Human Daily Activity Recognition in Wearable Body Sensor Networks", *International Journal of Healthc Eng*, Vol. 2020, 2020.

[15] J. Chen, Y. Sun, and S. Sun, "Improving Human Activity Recognition Performance by Data Fusion and Feature Engineering", *International Journal of Sensors*, Vol. 21, Page 692, Vol. 21, No. 3, p. 692, 2021.

[16] E. Fridriksdottir and A. G. Bonomi, "Accelerometer-Based Human Activity Recognition for Patient Monitoring Using a Deep Neural Network", *International Journal of Sensors*, Vol. 20, Page 6424, Vol. 20, No. 22, p. 6424, 2020.

[17] C. Ma, W. Li, J. Cao, J. Du, Q. Li, and R. Gravina, "Adaptive sliding window based activity recognition for assisted livings", *International Journal of Information Fusion*, Vol. 53, pp. 55–65, 2020.

[18] A. Hirawat, S. Taterh, and T. K. Sharma, "A dynamic window-size based segmentation technique to detect driver entry and exit from a car", *International Journal of King Saud University - Computer and Information Sciences*, Vol. 34, No. 10, pp. 8514–8522, 2022.

[19] M. Liandana, D. P. Hostiadi, and G. A. Pradipta, "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*, Vol. 17, No. 2, pp. 235–250, 2024, doi: 10.22266/ijies2024.0430.21.

[20] B. Vidya and S. P, "Wearable multi-sensor data fusion approach for human activity recognition using machine learning algorithms", *International Journal of Sens Actuators A Phys*, Vol. 341, p. 113557, 2022.

[21] H. Qian, S. J. Pan, and C. Miao, "Weakly-supervised sensor-based activity segmentation and recognition via learning from distributions", *International Journal of Artif Intell*, Vol. 292, p. 103429, 2021.

[22] M. Jaén-Vargas, K. M. R. Leiva, F. Fernandes, S. B. Goncalves, M. T. Silva, D. S. Lopes, and J. J. S. Olmedo, "Effects of sliding window variation in the performance of acceleration-based human activity recognition using deep

learning models", *International Journal of PeerJ Comput Sci*, Vol. 8, p. e1052, 2022.

[23] L. Sun, X. Yang, and C. Hu, "DSWHAR: A Dynamic Sliding Window Based Human Activity Recognition Method", In: *Proc. of 2022 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Autonomous and Trusted Vehicles, Scalable Computing and Communications, Digital Twin, Privacy Computing, Metaverse, SmartWorld/UIC/ATC/ScalCom/DigitalTwin/PriComp/Metaverse 2022*, pp. 1421–1426, 2022.

[24] N. Alhammad and H. Al-Dossari, "Dynamic Segmentation for Physical Activity Recognition Using a Single Wearable Sensor", *International Journal of Applied Sciences*, Vol. 11, Page 2633, Vol. 11, No. 6, p. 2633, 2021.

[25] A. M. A. Baraka and M. H. Mohd Noor, "Similarity Segmentation Approach for Sensor-Based Activity Recognition", *International Journal of IEEE Sens J*, Vol. 23, No. 17, pp. 19704–19716, 2023.

[26] A. R. Baraka and M. H. Mohd Noor, "Deep similarity segmentation model for sensor-based activity recognition", *International Journal of Multimed Tools Appl*, pp. 1–24, 2024.

[27] S. Geravesh and V. Rupapara, "Artificial neural networks for human activity recognition using sensor based dataset", *International Journal of Multimed Tools Appl*, Vol. 82, No. 10, pp. 14815–14835, 2023.

[28] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective", *International Journal of Neurocomputing*, Vol. 300, pp. 70–79, 2018.

[29] W. Albattah, R. U. Khan, M. F. Alsharekh, and S. F. Khasawneh, "Feature Selection Techniques for Big Data Analytics", *International Journal of Electronics*, Vol. 11, Page 3177, Vol. 11, No. 19, p. 3177, 2022.

[30] L. Wang, S. Jiang, and S. Jiang, "A feature selection method via analysis of relevance, redundancy, and interaction", *International Journal of Expert Syst Appl*, Vol. 183, p. 115365, 2021.

[31] L. Gao and W. Wu, "Relevance assignation feature selection method based on mutual information for machine learning", *International Journal of Knowl Based Syst*, Vol. 209, p. 106439, 2020.

[32] S. Canzhuang and F. Yonge, "Identification of Disordered Regions of Intrinsically Disordered Proteins by Multi-features Fusion", *International Journal of Curr Bioinform*, Vol. 16, No. 9, pp. 1126–1132, 2021.

[33] Y. H. Wang, Y. F. Zhang, Y. Zhang, Z. F. Gu, Z. Y. Zhang, H. Lin, and K. J. Deng, "Identification of adaptor proteins using the ANOVA feature selection technique", *International Journal of Methods*, Vol. 208, pp. 42–47, 2022.

[34] J. Wang, P. Xu, X. Ji, M. Li, and W. Lu, "Feature Selection in Machine Learning for Perovskite Materials Design and Discovery", *International Journal of Materials*, Vol. 16, Page 3134, Vol. 16, No. 8, p. 3134, 2023.

[35] M. S. Pathan, A. Nag, M. M. Pathan, and S. Dev, "Analyzing the impact of feature selection on the accuracy of heart disease prediction", *International Journal of Healthcare Analytics*, Vol. 2, p. 100060, 2022.

[36] Y. Kang, C. Ma, C. Xu, L. You, and Z. You, "Prediction of drilling fluid lost-circulation zone based on deep learning", *International Journal of Energy*, Vol. 276, p. 127495, 2023.

[37] J. L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, "Transition-Aware Human Activity Recognition Using Smartphones", *International Journal of Neurocomputing*, Vol. 171, pp. 754–767, 2016.

[38] "Smartphone-Based Recognition of Human Activities and Postural Transitions - UCI Machine Learning Repository", Accessed: Mar. 03, 2024. [Online]. Available: https://archive.ics.uci.edu/dataset/341/smartphone+based+recognition+of+human+activities+and+postural+transitions

[39] K. Karagiannaki, A. Panousopoulou, and P. Tsakalides, "A benchmark study on feature selection for human activity recognition", In: *Proc. of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 105–108, 2016.

[40] "WISDM Lab: Dataset." Accessed: Sep. 21, 2024. [Online]. Available: https://www.cis.fordham.edu/wisdm/dataset.php

[41] M. Munoz-Organero, I. D. Luptáková, M. Kubovčík, and J. Pospíchal, "Wearable Sensor-Based Human Activity Recognition with Transformer Model", *International Journal of Sensors*, Vol. 22, Page 1911, Vol. 22, No. 5, p. 1911, 2022.

[42] M. Mardanpour, M. Sepahvand, F. Abdali-Mohammadi, M. Nikouei, and H. Sarabi, "Human activity recognition based on multiple inertial sensors through feature-based knowledge distillation paradigm", *International Journal of Inf Sci (N Y)*, Vol. 640, p. 119073, 2023.

Received: September 30, 2024. Revised: November 12, 2024.

[43] W. Shi, X. Fang, G. Yang, and J. Huang, “Human Activity Recognition Based on Multichannel Convolutional Neural Network With Data Augmentation”, *IEEE Access*, Vol. 10, pp. 76596–76606, 2022.

[44] M. Akter, S. Ansary, M. A. M. Khan, and D. Kim, “Human Activity Recognition Using Attention-Mechanism-Based Deep Learning Feature Combination”, *International Journal of Sensors*, Vol. 23, Page 5715, Vol. 23, No. 12, p. 5715, 2023.

[45] C. Zhang, K. Cao, L. Lu, and T. Deng, “A multi-scale feature extraction fusion model for human activity recognition”, *Scientific Reports*, Vol. 12, No. 1, pp. 1–13, 2022.