Activity Recognition with Four Accelerometers

Amirhossein Rostami[1] , Abbas Zal[2]

***Abstract*—Wearable accelerometers have become integral in health research, offering precise, continuous measurements of physical activity (PA). This study utilizes the database contains raw accelerometry data collected during outdoor walking, stair climbing, and driving for 32 healthy adults., where participants wore tri-axial accelerometers on their hips,wrists and ankles. We developed Convolutional Neural Network (CNN) and combination of CNN and GRU units for four accelerometers placed on the left wrist, left hip, left ankle, and right ankle. These models achieved high accuracy in classifying of three activities(walking, descending stairs, and ascending stairs) outperforming traditional methods.**

***Index Terms*—Physical activity, Signal Processing, Feature capturing, Convolutional Neural Networks (CNNs), Gated Recurrent Units (GRUs), Activity recognition**

I. Introduction

Wearable physical activity (PA) monitors have revolutionized health research by providing detailed, continuous, and objective measurements of individual physical activity in free-living environments. Advances in technology and the decreasing cost of wearable devices have led to their widespread use in health research, offering more accurate and reliable data than traditional subjective measurements. However, the utilization of wearable accelerometers presents several challenges, including the handling of vast data volumes, managing within- and between-subject variability, and ensuring proper calibration and synchronization of devices.

Each device collects data at a high frequency, resulting in large datasets that require sophisticated methods for storage, processing, and analysis. Moreover, significant variability exists within and between subjects due to differences in body size, musculature, and movement patterns, complicating the development of generalized models for activity recognition. The placement of sensors on different body locations, such as the hip and wrist, introduces further complexity, as the data characteristics vary with sensor location. Additionally, issues like device calibration, sampling frequency, and synchronization of multiple monitors add to the challenges, necessitating robust data processing protocols.

To address these issues, this study employs data from 32 participants that wore tri-axial accelerometers on their left hip, left wrist and ankles. By using open-source methods for data processing and summarization, the study generates meaningful insights, underscoring the importance of high-quality and reproducible data processing protocols.

Our primary objective is to leverage accelerometry data to enhance activity recognition. To achieve this, we developed and evaluated two advanced machine learning models: Convolutional Neural Network (CNN) and Combined CNN with Gated Recurrent Units (GRU) models. These models were trained on data from four body locations: left wrist, left hip, left ankle, and right ankle. Each model brings unique advantages: CNNs are adept at spatial feature extraction, and GRUs offer a simplified architecture for sequence learning with fewer parameters. By classifying activities such as walking, descending stairs, and ascending stairs, our models demonstrate their effectiveness in improving the accuracy and robustness of activity recognition in health research.

Specifically, our CNN model leverages multiple convolutional layers to extract local and global features from the accelerometer data, employing techniques such as batch normalization and dropout to enhance performance and prevent overfitting. GRU model, with its gated architecture, provides an efficient alternative for sequence modeling. The GRU models are particularly suited for time-series data, capturing the temporal dynamics essential for accurate activity classification. Combining these approaches allows for a comprehensive evaluation of their performance in recognizing physical activities from wearable sensor data.
The structure of this paper is as follows. In Section II, we review related work in the field of signals classification using deep learning. Section III The data processing pipeline will be shown. In Section IV, we detail two kind of data preprocessing used for training and evaluation. Section V presents the Learning Framework for "CNN for Multivariate Time Series Classification" and "CNN-GRU" models. In section VI presents the results of our experiments, highlighting the performance improvements achieved by our model. Finally, Section VII offers concluding remarks and discusses potential future work.

II. Related Work

Over the past ten years, many studies have focused on training time-dependent data, expressed as $p(x_n \mid x_{n-1}, x_{n-2}, \ldots)$. Recurrent Neural Networks (RNNs), especially Long Short-Term Memory (LSTM) networks [1], have become the main method for capturing long-term dependencies in sequential data. Besides RNNs, researchers have also been exploring the use of Convolutional Neural Networks (CNNs) for time series analysis, showing that they can effectively learn patterns and dependencies in sequential data.

Our activity recognition model leverages a Convolutional Neural Network (CNN) architecture, drawing significant inspiration from recent advancements in the field of activity

[1] email: amirhossein.rostami@studenti.unipd.it
[2] email: abbas.zal@studenti.unipd.it

recognition. Specifically, Our model design is influenced by methodologies described in the works of Rizal et al. (2023) [2]. These sources provided essential insights into optimizing window sizes and step sizes for segmenting accelerometer signals, as well as implementing effective CNN layers and configurations for robust feature extraction and classification. The signals is obtained from a fleet of gas sensors that measure and track quantities such as oxygen and sound. At first, they analyze the time series data to understand the effect of different parameters, such as the sequence length, when training our models. FCNs are advantageous due to their ability to capture local and global features of multivariate time series data without pooling operations, preserving the sequence length. They consist of several convolutional blocks followed by Global Average Pooling (GAP) to reduce parameters and improve efficiency. The CNN architecture of our model includes multiple convolutional layers designed to extract features from the accelerometer data, leveraging the insights for robust feature extraction and classification. Another model that is appropriate for our activity recognition is CNN - GRU model. Our model design is influenced by methodologies described in the works of [3].There are many methods to diagnose heart disease; the most effective way is to analyze electrocardiogram (ECG) signals. Generally, the automatic classification techniques based on ECG analysis consist of three steps: data preprocessing, feature extraction, and classification. This study designed eight hybrid model architectures using several types of deep neural networks, including Convolution Neural Network (CNN), Gated Recurrent Unit (GRU). this models combine Convolutional Neural Networks (CNNs) for spatial feature extraction with Gated Recurrent Units (GRUs) for sequential data processing.

III. PROCESSING PIPELINE

The dataset used in this project consists of measurements from four accelerometers worn on different parts of the body [4].The data is structured to include various variables that capture the activity and gravitational acceleration in different axes for each accelerometer. The data format includes the following variables:

activity: Type of activity

time_s: Time from device initiation (seconds)

lw_x, lw_y, lw_z: Left wrist x, y, and z-axis measurements (gravitational acceleration)

lh_x, lh_y, lh_z: Left hip x, y, and z-axis measurements (gravitational acceleration)

la_x, la_y, la_z: Left ankle x, y, and z-axis measurements (gravitational acceleration)

ra_x, ra_y, ra_z: Right ankle x, y, and z-axis measurements (gravitational acceleration)

The data processing pipeline, as illustrated in Figure 1, involves the following steps: For the modeling process, two different preprocessing are employed, each utilizing the entire dataset without splitting it. The details of these preprocessing steps are discussed in the following sections (IV) and (IV).

Both models proceed to their respective learning frameworks, where deep learning algorithms are trained on the preprocessed data. The training process involves feeding the data into the models and optimizing the parameters to accurately predict the activities. We will describe the models in detail.(V) and V

After training, each model is used for classification. The output of the classification process is the type of activity being performed. The possible outputs for our project are:

- **0**: Walking
- **1**: Descending stairs
- **2**: Ascending stairs

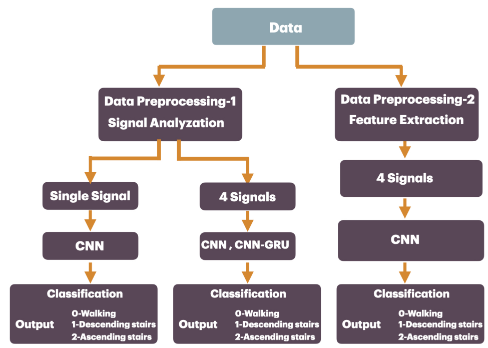

Fig. 1: Processing pipeline

IV. SIGNALS AND FEATURES

In this section, we describe the two main data preprocessing strategies applied to the accelerometer data to prepare it for training various models.

Data Combination

The first step involved loading and combining the datasets from individual participants into a single comprehensive dataset. Each participant's data was stored in a separate CSV file. We combined these files into one large dataset (with preserving the time column), which now contains approximately 2 million rows. This combined dataset includes accelerometer signals from four different detectors (left wrist, left hip, left ankle, right ankle) and corresponding activity labels.

Initially, we combined data for all participant, as in [5] suggested, in order to remove gravitational and orientations, we computed the amplitude of the signals in three axes using the formula:

$$\text{Amplitude(t)} = \sqrt{x(t)^2 + y(t)^2 + z(t)^2}$$

We also removed unwanted labels (driving and unkown activities) [5]. These steps were applied to all four signals.

DATA PREPROCESSING-1

For this preprocessing, Two approaches were used to create the input for training:

1- single signal selection

2- Combination of all four signals

Window Selection

Specifically, we used a window selection approach with two parameters: window size and step size. In this approach, data from each accelerometer signal was divided into windows of varying sizes (256, 512, and 1024 samples), corresponding to 2.56 seconds, 5.12 seconds, and 10.24 seconds at a sampling frequency of 100Hz in which the next window index is selected by step size. It is also important to note that we ensured the activity type (label) remained constant within each window. The next steps are applying the Baseline Wander Removal and Normalization to all signals.

Baseline Wander Removal

Baseline wander, a low-frequency noise in accelerometer data, was removed using median filters to estimate and subtract the baseline from the original signal.

Baseline Estimation using Median Filters: For a signal $x[n]$, the median filter with window size $W = 2k + 1$ computes:

$$\tilde{x}[n] = \text{median}\{x[n-k], \ldots, x[n], \ldots, x[n+k]\} \quad (1)$$

Baseline Removal: The baseline-corrected signal $y[n]$ is obtained by:

$$y[n] = x[n] - \tilde{x}[n] \quad (2)$$

This method enhances signal clarity, improving feature extraction and classification accuracy.

Normalization

Normalization adjusts the scale of data to ensure consistent signal amplitudes across sensors and activities, reducing outlier impact and scaling data to a consistent range.

Compute the 99th and 5th Percentiles: For a signal $x[n]$ with $n = 1, 2, \ldots, N$: Let P_{99} and P_5 be the 99th and 5th percentiles of $x[n]$.

Normalize the Signal: The normalized signal $y[n]$ is obtained by:

$$y[n] = \frac{x[n]}{P_{99} - P_5}$$

This ensures consistent signal amplitude, reducing outlier impact and improving model performance and robustness.

DATA PREPROCESSING-2

To enhance the dataset and extract more informative features for model training, a second preprocessing method was applied. This method involved the following steps:

Window Selection and Feature Extraction

In this approach, data from each accelerometer signal was divided into non-overlapping windows (without step size) of varying sizes (as described in Section IV). For each window, several statistical and frequency domain features were extracted to capture the characteristics of the signal. These features include: normal and DCT/FFT features of a signal. Table 1

TABLE 1: Extracted Features

mean	fft_mean	dct_mean
std	fft_std	dct_std
skew	fft_skew	dct_skew
kurtosis	fft_kurtosis	dct_kurtosis
min	fft_min	dct_min
max	fft_max	dct_max
fft_energy	range	fft_range
dct_range	dct_energy	height

Feature Normalization

After extracting features, we normalize each using the Median Absolute Deviation (MAD) [6] [5]. For a feature w, calculate $MAD(w)$ as:

$$MAD(w) = 1.4826 \times \text{median}(|w - \text{median}(w)|)$$

Normalize by:

$$z^* = \frac{w - \text{median}(w)}{MAD(w)}$$

Dataset Preparation

The extracted features were compiled into a new dataset for each signal and window size combination. This dataset includes the calculated statistical and frequency domain features for each window, providing a comprehensive set of attributes for model training.

V. LEARNING FRAMEWORK

CNN for Multivariate Time Series Classification

Our activity recognition model leverages a Convolutional Neural Network (CNN) architecture designed to classify activities based on accelerometer signals. The following sections describe the structure and components of the CNN model used in this study.

The CNN model is composed of multiple layers designed to extract features from the accelerometer data and perform activity classification. We used in all layer 1D filters. The architecture can be summarized as follows:

- **Input Layer:** The input shape depends on the data preprocessing type.
- **First Convolutional Layer:**
  - Applies 64 filters with a kernel size of 3.
  - Uses ReLU (Rectified Linear Unit)
  - Followed by Batch Normalization
  - Max Pooling with a pool size of 2
  - Dropout with a rate of 0.3
- **Second Convolutional Layer:**
  - Applies 128 filters with a kernel size of 3 and ReLU activation.
  - Followed by Batch Normalization, Max Pooling with a pool size of 2, and Dropout with a rate of 0.3.
- **Third Convolutional Layer:**
  - Uses 256 filters with a kernel size of 3 and ReLU activation.

- Followed by Batch Normalization, Max Pooling with a pool size of 2, and Dropout with a rate of 0.3.
- **Fourth Convolutional Layer:**
  - Applies 512 filters with a kernel size of 3 and ReLU activation.
  - Followed by Batch Normalization, Max Pooling with a pool size of 2, and Dropout with a rate of 0.3.
- **Flattening Layer:**
  - Converts the multi-dimensional output from the convolutional layers into a single-dimensional vector.
- **Fully Connected Layers:**
  - A Dense layer with 512 units and ReLU activation.
  - Dropout with a rate of 0.5
  - Another Dense layer with 256 units and ReLU activation, followed by Dropout with a rate of 0.5.
- **Output Layer:**
  - A final Dense layer with the number of neurons equal to the number of activity classes (3 in this case).
  - Uses Softmax activation to output the probability distribution over the activity classes.

Training and Optimization

To enhance the training process and improve the model's generalization capabilities, we employ early stopping and learning rate reduction techniques:

- **Early Stopping:** Monitors the validation loss and stops training if the loss does not improve for 10 epochs (patience), restoring the best model weights.
- **Reduce - LR:** Reduces the learning rate by a factor of 0.2 if the validation loss plateaus for 5 epochs, allowing the model to converge more effectively. The minimum learning rate is set to 1×10^{-5}.

The model is compiled using the Adam optimizer and trained with a sparse categorical cross-entropy loss function. We balance the class weights to address the issue of imbalanced data Figure 2, ensuring that the model does not bias towards the more frequent activities.

This CNN model is designed to efficiently process and classify accelerometer data into different activity types, leveraging the power of deep learning to improve accuracy and robustness. The model's architecture and training framework are optimized to handle the specific characteristics of the accelerometer signals and provide reliable activity recognition.

CNN and GRU Model

The following sections describe the structure and components of these model.

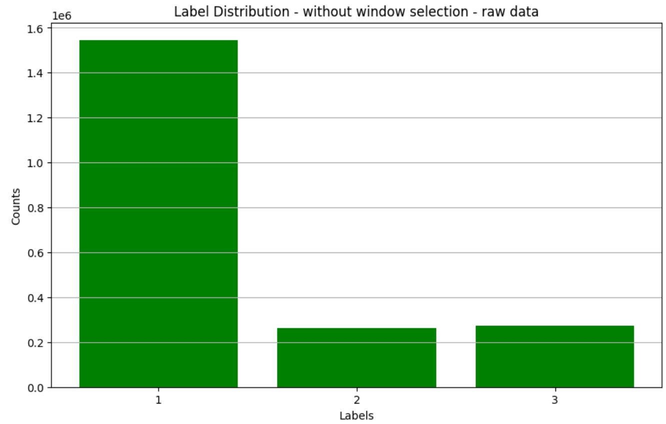

Fig. 2

Model Architecture

The CNN-GRU model is composed of multiple layers designed to extract features from the accelerometer data and perform activity classification. The architecture can be summarized as follows:

- **Input Layer:** The input shape is (*input size*, 4), where *input size* corresponds to the window size.
- **First Convolutional Layer:**
  - Applies 64 filters with a kernel size of 3.
  - Uses ReLU (Rectified Linear Unit) activation.
  - Max Pooling with a pool size of 2
- **Second Convolutional Layer:**
  - Applies 128 filters with a kernel size of 3 and ReLU activation.
  - Max Pooling with a pool size of 2.
  - Dropout with a rate of 0.5
- **First GRU Layer:**
  - Contains 128 units.
  - Return sequences set to true to pass outputs to the next GRU layer.
- **Second GRU Layer:**
  - Contains 128 units.
- **Fully Connected Layer:**
  - A Dense layer with 64 units and ReLU activation.
  - Dropout with a rate of 0.5.
- **Output Layer:**
  - A final Dense layer with the number of neurons equal to the number of activity classes.
  - Uses Softmax activation to output the probability distribution over the activity classes.

VI. Results

Results for CNN Model for preprocessing - 1

To analyze the accelerometer signals, we segmented the data into windows, each containing a fixed number of samples. In this part we will check the results for CNN model for preprocessing - 1 in two approach as mentioned(1-

single signal selection 2- Combination of all four signals).

single signal selection:
As we mentionedIV, there are two parameters in the window selection part. Selecting the step size requires careful consideration. After extensive testing, we found that one of the best step size is equal to the window size, except for a window size of 1024. In this case, setting the step size to 1024 causes to have less signals for training. The activity label for each window was assigned based on the activity during that period. We used different window sizes (256, 512, 1024 samples) to evaluate the model's performance. We trained the model with different window sizes for 4 signals separately . The results are reported in following Tables various statuses. Table 2, 3, 4

TABLE 2: Window size 256 - step size 256

signal	Final Accuracy	Precision	Recall	F1 Score
Left wrist	0.8743	0.8782	0.8743	0.8752
Left hip	0.9760	0.9758	0.9760	0.9758
Left ankle	0.9729	0.9730	0.9729	0.9729
Right ankle	0.9904	0.9903	0.9904	0.9903

TABLE 3: Window size 512 - step size 512

Signal	Final Accuracy	Precision	Recall	F1 Score
Left wrist	0.8560	0.8300	0.8560	0.8315
Left hip	0.9579	0.9563	0.9579	0.9561
Left ankle	0.9897	0.9897	0.9897	0.9896
Right ankle	0.9962	0.9962	0.9962	0.9962

TABLE 4: Window size 1024 - step size 256

Signal	Final Accuracy	Precision	Recall	F1 Score
Left wrist	0.8736	0.8832	0.8736	0.8688
Left hip	0.9852	0.9854	0.9852	0.9854
Left ankle	0.9956	0.9956	0.9956	0.9956
Right ankle	0.9961	0.9962	0.9961	0.9961

Now , If we compare the result of [Fadel19] with our model's result5,(for example we did it for 4 signals at window size 1024) we can extract this point that Our CNN model surpasses the Classification Tree approach in [Fadel19] for several reasons. Firstly, it automatically extracts complex features from raw signals, eliminating the need for manual feature engineering. Secondly, CNNs effectively handle non-linear relationships within the data due to their hierarchical structure and non-linear activation functions. Thirdly, regularization techniques, such as dropout.

Combination of all four signals:
We also wanted to test the model with all signals combined.

TABLE 5: Comparison of F1 Scores

Signal\window size 1024	CNN F1 Score	Classification Tree F1 Score
left wrist	0.86	0.80
Left hip	0.98	0.85
Left ankle	0.99	0.85
Right ankle	0.99	0.83

For this purpose, we first concatenated the data from different signals and then fed the combined data into the model. The results were better because optimal sensor placement often involves a combination of multiple sensors.(you can see the result in table 6) For example, placing sensors on both wrists, both ankles, and the pelvis provides a comprehensive dataset that enhances the classification accuracy of various activities. This combination leverages the rich information from these key points to provide a more complete picture of body movement [7].

TABLE 6: Window size 256 for combined data (weighted avg)

Signal	Precision	Recall	F1 Score
All signals	0.96	0.96	0.95

Results for CNN Model for preprocessing - 2

Now, We want to present the outcomes of the second data preprocessing strategy applied to our accelerometer dataset. This approach aimed to extract a comprehensive set of statistical and frequency domain features . The preprocessing were explained in detail in signals and features sectionIV.

The CNN model's performance was evaluated using the same metrics as in the first preprocessing method: accuracy, precision, recall, and F1 score. The results for different window sizes are summarized in the following table.

TABLE 7: Performance for Data Preprocessing - 2

Window Size	Final Accuracy	Precision	Recall	F1 Score
256	0.98	0.95	0.95	0.95
512	0.99	0.99	0.97	0.98
1024	0.98	0.97	0.91	0.94

Results for CNN and GRU Model

We used data preprocessing - 1 for preparing data for training IV. The results for different window sizes and step size for combined 4 signals are summarized in the following table:

TABLE 8: Performance for CNN - GRU Model

Window Size - step size	Final Accuracy	Precision	Recall	F1 Score
256 - 256	0.98	0.98	0.98	0.98
512 - 512	0.98	0.98	0.98	0.98
1024 - 256	0.98	0.99	0.98	0.98

VII. Concluding Remarks

we evaluated the performance of various body-worn sensors in recognizing physical activities such as walking, ascending stairs, and descending stairs. Our findings indicate that sensors placed on the ankles and hips provided the most reliable data, outshining wrist sensors. This highlights the importance of sensor placement in achieving accurate activity recognition. Additionally, combining data from all sensor locations resulted in the highest accuracy in activity classification, demonstrating the value of a multi-sensor approach in human activity recognition.

The Convolutional Neural Network (CNN) model, along with the Convolutional Neural Network (CNN) and Gated Recurrent Units (GRU) models, developed in this study outperformed traditional methods by effectively handling large data volumes, variability, and complex patterns inherent in accelerometer data. These deep learning models, particularly the CNN-GRU architectures, leveraged their ability to capture temporal dependencies and non-linear patterns, resulting in superior performance.

This research highlights the significant potential of wearable accelerometers in advancing health research, emphasizing the importance of sophisticated data processing and machine learning techniques in harnessing the full potential of accelerometry data. The successful application of CNN , CNN-GRU models demonstrates the crucial role of deep learning in enhancing activity recognition, paving the way for more accurate and reliable health monitoring systems.

Potential Future Work

Sensor Placement Optimization: Future studies can explore optimal sensor placement strategies. The placement of sensors for activity recognition plays a crucial role. In our project, various activities were monitored, For the types of activities we examined, sensors placed on the ankles and hips provided the best accuracy. Thus, the placement of sensors for activity recognition is closely related to the type of activity being performed.

Incorporation of Additional Sensors: As observed, the combination of four signals enhanced the accuracy of activity recognition. Future experiments would benefit from using more sensor placements to recognize different kinds of activities.

References

[1] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *arXiv preprint arXiv:1402.1128*, 2014.

[2] A. Rizal and Author3, "Multivariate time-series classification using memory and attention for long and short-term dependence," *arXiv preprint arXiv:2307.02253*, 2023.

[3] Z. Oleiwi, E. AlShemmary, and S. Al-Augby, "Developing hybrid cnn-gru arrhythmia prediction models using fast fourier transform on imbalanced ecg datasets," *Mathematical Modelling of Engineering Problems*, vol. 11, no. 2, pp. 413–429, 2022.

[4] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Accelerometry for walking, climbing stairs, and driving," 2023.

[5] W. F. Fadel, J. K. Urbanek, S. R. Albertson, X. Li, A. K. Chomistek, and J. Harezlak, "Differentiating between walking and stair climbing using raw accelerometry data," *Statistics in Biosciences*, vol. 11, no. 2, pp. 334–354, 2019.

[6] L. Xiao, B. He, A. Koster, P. Caserotti, B. Lange-Maia, N. Glynn, T. Harris, and C. Crainiceanu, "Movement prediction using accelerometers in a human population," *Biometrics*, vol. 72, no. 2, pp. 513–524, 2016.

[7] O. Konak, A. Wischmann, R. van de Water, and B. Arnrich, "Pose recognition and analysis," 2023.