# Actify

Catalin Ghitun, Carlo Alberto Alfieri, Mariapia Sorrentino
Sapienza University of Rome
{ghitun.2059802, alfieri.2263545, sorrentino.2268884}@uniroma1.it

## 1. Introduction

Human Activity Recognition (HAR) aims to automatically identify human activities from sensor data.

In this project, we address HAR using raw signals collected from smartphone inertial sensors, namely the accelerometer and gyroscope, to classify the activity performed.

We adopt a sensor-based approach instead of vision-based solutions because it is computationally more efficient, robust to environmental conditions such as lighting, and does not raise privacy concerns. Moreover, smartphones are ubiquitous and allow for low-cost and non-intrusive data collection.

This task is particularly relevant in the healthcare domain, where HAR systems can be used to monitor daily activities of elderly people, enabling the detection of abnormal behaviors and supporting assisted living applications.

## 2. Related Work

Human Activity Recognition (HAR) using smartphone sensors has been widely studied. Early works relied on classical machine learning with handcrafted features from accelerometer and gyroscope data. Recent approaches exploit deep learning, such as convolutional autoencoder-LSTM (ConvAE-LSTM) and attention-based temporal convolutional networks (TCN-Attention-HAR), to automatically extract temporal features from raw signals, achieving high accuracy on benchmark datasets [4, 5].

## 3. Dataset

In this work, we utilize the **UCI HAR Dataset** [1]. The data were collected from a group of 30 volunteers performing six different activities of daily living (ADL) while carrying a waist-mounted smartphone.

The activity labels can be categorized into two main groups based on the nature of the activity:

- **Static Activities:** Sitting, Standing, Laying.
- **Dynamic Activities:** Walking, Walking Upstairs, Walking Downstairs.

The dataset consists of a total of 10,229 samples. Following the standard protocol, the data is partitioned into 70% for the training set and 30% for the test set. Crucially, the subjects in the training set are distinct from those in the test set to ensure realistic evaluation. To maintain this subject independence during the validation phase, we employed a **GroupKFold** cross-validation strategy with 5 folds. This ensures that samples from the same subject do not appear simultaneously in both the training and validation splits.

Finally, our model input consists of a signal with **128 timesteps** and **9 channels**. Regarding preprocessing, we apply only a **channel-wise standardization** to normalize the input features.

## 4. ConvAE-LSTM

We implemented the Convolutional Autoencoder LSTM architecture based on [4], which proposes this method as an efficient unsupervised alternative to standard CNN-LSTMs for Human Activity Recognition. The architecture comprises a single convolutional block with 64 filters (kernel size 3), with maxpooling and a stride both of size 2. The resulting latent space is then fed to an LSTM, of which hyperparameters were not detailed in the study.

This model is a valuable alternative for two main reasons: unsupervised feature extraction and model complexity. The Convolutional Autoencoder, in fact, acts as an unsupervised feature extractor which, through convolution, denoises the input signal, reduces its time dimension and enriches its feature channels to feed the LSTM model with enriched signals. Thanks to this structure, this kind of model can be trained even without a fully labeled dataset but requires just some labeled samples for scoring the final LSTM classification, differently from classical CNN-LSTM architectures, requiring a fully labeled dataset, which can be expensive and hard to obtain. Additionally, in our model realization, the Convolutional Autoencoder model achieved high performance with significantly fewer parameters. In fact, our final architecture solution used 816k parameters, which are noticeably fewer than the 2.5 million used by the proposed CNN-LSTM architecture.

Furtherly, the model is particularly efficient at detecting signal spikes due to its small kernel size and aggressive downsampling. Through convolution, the model learns temporal dependencies, while maxpooling cancels the effects of small input translations, which is a crucial benefit for translation-invariant human activity signals [3].

Since no codebase was provided, we implemented the model from scratch, targeting the study's reported 0.98 test score on the UCI HAR dataset. The hyperparameters were then tested and optimized using a wandb agent running a Bayesian search for optimal hyperparameter tuning. The LSTM was structured as bidirectional (2 layers, 256 hidden units, 0.4 dropout) to capture greater temporal context and avoid overfitting. Adam was selected as the optimizer, having consistently outperformed SGD in all scenarios. Additionally, for the AE, Huber loss was picked since, having to deal with huge spikes and noises, this choice was able to provide robustness to outliers, behaving linearly for large errors, using L1 loss, and then shifting to L2 loss measure to ensure faster and more precise convergence for better weights fine-tuning. [2]

Initially, we tried a slightly different approach from the paper, opting for a lower resolution model with a wider kernel size and average pooling to avoid having results heavily impacted by the signals' noise. With this configuration, the model achieved a 0.92 test accuracy, closely matching the paper's F1-scores for dynamic activities and laying (ranging between 0.96 and 0.99) but significantly underperforming on sitting and standing (0.83 and 0.85).

This was mainly due to the average pooling and wider kernel size effects that are good to smooth signal noise and can perform well on dynamic activities, which are mostly characterized by high frequency signals, and are hence not impacted by smoothing, while on sitting and standing activities, which differ by small nuances in their signals, this smoothing effect was hampering the final classification.

Differently, the small kernel and maxpooling approach (the paper structure) resulted in better performance, scoring better on sitting and standing classification, with F1 scores of 0.85 and 0.88.

This discrepancy highlights the benefits and downsides of the smoothing effect provided by a wide kernel and average pooling, which while is beneficial for denoising high-frequency dynamic signals, it can be hampering the classification of static activities like sitting and standing activities, which differ by small nuances in their signals, and can hence suffer from this smoothing effect which can delete some of their small differences.

Ultimately, both configurations achieved strong performance, almost reaching the target scores. This dual implementation highlighted the trade-offs between high and low-resolution architectures: while the paper's high-resolution structure offers precision, the lower-resolution variant can still be considered a viable, robust solution, especially for scenarios involving high noise due to low-quality signals.

# 5. CNN-LSTM

To address the limitations of reconstruction-based approaches in distinguishing static classes, we propose a direct **End-to-End CNN-LSTM** architecture via Cross-Entropy Loss, forcing the network to learn filters that maximize class separability while discarding non-discriminative static noise.

## 5.1. Feature Extraction (CNN)

The encoder is designed to transform the low-dimensional raw signal ($N \times 9 \times 128$) into a high-level discriminative representation. To balance model capacity and generalization, we designed two Convolutional Blocks with specific parameters:

- **Temporal Resolution:** We selected a kernel size of 5, allowing the network to capture local temporal dependencies across sensor channels without over-smoothing the signal.
- **Noise Reduction:** We selected **Max Pooling with a stride of 2** to progressively downsample the time dimension ($128 \rightarrow 64 \rightarrow 32$), suppressing the high-frequency noise typical of raw accelerometer data while retaining the main activity peaks.
- **Regularization Strategy:** To prevent overfitting due to the small subject pool (30 volunteers), we applied an aggressive regularization, including **Batch Normalization** and high **Dropout** rates ($p \approx 0.4$) in each convolutional block. This ensures that the learned features represent the activity itself rather than the specific individual movement styles.

## 5.2. Sequence Modeling (BI-LSTM)

The compressed feature sequence is permuted to shape ($N \times 32 \times 128$) to align with the input requirements. Then the sequence is processed by a **Bidirectional LSTM** (2 layers, 256 hidden units) to capture long-term dependencies. This approach allows the model to aggregate information from both past and future states before passing the resulting summary vector $(N, 512)$ to a fully connected Dense layer with Softmax activation for the final 6-class classification.

# 6. Experimental Results & Discussion

## 6.1. Validation Strategy

Following the **GroupKFold** protocol defined in Section 3, the proposed CNN-LSTM model achieved an average validation accuracy of **94.70% ± 2.91%** and evaluation on the unseen Test Set yielded a final accuracy of **97.25%**.
This represents an improvement over the baseline **ConvAE** approach, which achieved an average validation accuracy of **0.9171 % ± 0.0517%** and a test accuracy of **93.68%**.

### 6.2. Comparison

The most significant improvement was the performance on **Static Classes** (*Sitting vs. Standing*).
CNN-LSTM reduced misclassifications between these two classes to only 52 samples. While ConvAE still struggled to denoise the similarity of these two postures, CNN-LSTM learned to identify subtle axis-orientation shifts, ignoring the static components.

This confirms that for this kind of classification task, where classes share significant structural similarities, a direct approach yields major results.

### 6.3. Conclusion & Future Work

In this work, we demonstrated that a direct CNN–LSTM architecture achieves higher accuracy for Human Activity Recognition compared to the baseline approach, successfully mitigating the “Sitting vs. Standing” ambiguity.
Nevertheless, there is still room for improvement. Future work will focus on replacing the LSTM component with a temporal attention mechanism,which can allow to better classify different acitivicties characterized by a stronger and longer temporal dependency. scoring differently only the most relevant time steps for the activity classification.

## References

[1] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L. Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2013. URL: https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones.

[2] PyTorch Contributors. torch.nn.HuberLoss — PyTorch Documentation, 2025. Accessed: 2025-12-23. URL: https://docs.pytorch.org/docs/stable/generated/torch.nn.HuberLoss.html.

[3] Charissa Ann Ronao and Sung-Bae Cho. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Systems with Applications*, 59:235–244, 2016. URL: https://www.sciencedirect.com/science/article/abs/pii/S0957417416302056, doi:10.1016/j.eswa.2016.04.032.

[4] Dipanwita Thakur, Suparna Biswas, Edmond S. L. Ho, and Samiran Chattopadhyay. ConvAE-LSTM: Convolutional Autoencoder LSTM for Smartphone-Based HAR. *IEEE Access*, 2022.

[5] Wei Xiong and Zifan Wang. TCN-Attention-HAR: Human Activity Recognition based on Attention Mechanism Time Convolutional Network. *Scientific Reports*, 14:7414, 2024. doi:10.1038/s41598-024-57912-3.