Radar Based Human Activity Recognition: from Classification to Detection

Reda El Hail[1,2][0000−0001−9369−0810], Pouya Mehrjouseresht[3][0000−0003−4248−1095], Oluwatosin John Babarinde[3][0000−0002−0089−4599], Dominique schreurs[3][0000−0002−4018−7936], and Peter Karsmakers[1,2][0000−0001−8119−6823]

[1] KU Leuven, Department of Computer Science; Leuven.AI, B-2440 Geel, Belgium
[2] Flanders Make, MPRO, B-3000 Leuven, Belgium
[3] Waves: Core Research and Engineering (WaveCoRE), Department of Electrical Engineering (ESAT), KU Leuven, B-3001 Leuven, Belgium.

Abstract. FMCW radar is emerging as a key technology for contactless Human Activity Recognition (HAR) in medical settings. While effective in controlled environments, FMCW radar-based HAR faces challenges in realistic scenarios, particularly with continuous data streams. Traditional methods often rely on segmentation, which can be cumbersome, or fail to handle transitions between activities, leading to errors. This study introduces a novel approach that enhances activity detection by refining classifier outputs with simple probabilistic reasoning that takes into account activity transition probabilities. The proposed method improves the HAR detection F1 score with 6% compared to the baseline and even slightly improves the classification accuracy which does not take into historical information about previous activities.

Keywords: Human activity recognition · FMCW radar · Machine learning · Finite State Machine · Viterbi algorithm

1 Introduction

FMCW radar technology is a significant breakthrough in contactless sensing applications for medical use-cases such as Human Activity Recognition (HAR). Especially, since this technology is unobtrusive and contactless. With radar sensors, it is possible to estimate the location of the patient inside a medical facility, as well as to perform HAR under all lighting conditions. Knowledge about the location and type of movement, can provide insights about the patient's state, her/his mobility and dangerous situations like falling, staying for a long time in the toilet or sleep problems. Because of the complex nature of the FMCW signals for HAR a multitude of data-driven methods was studied for this purpose. However, the majority of research papers studies only the classification problem of human events where a machine learning model is evaluated on predefined radar segments that contain an isolated human activity. In realistic situations, instead of a classification task, a detection task is targeted where a continuous stream of radar is acquired and subsequently fed to a detection model. Evidently, this challenges the machine learning model more compared to the classification setup. In the literature, continuous HAR (detection task) has been studied using two primary approaches. The first is a two-stage process, where the continuous data stream is initially segmented, and then each segment is classified by a machine learning model. For example, Kang et al. [3] detect motion changes using probabilistic moments of the time-range map to differentiate between windows containing a single activity and those with transitions between activities. Segments with a single activity are then classified using a Convolutional Neural Network (CNN). Another example is the work by Amin et al. [1], which introduces a two-step approach to analyze movement patterns. This method first uses the Radon transform on time-range maps to distinguish between stationary activities and translational movements, followed by applying principal component analysis to both time-range and time-Doppler maps. The resulting eigenvectors serve as input features for a k-nearest neighbors classifier. Additionally, the study proposes an ethogram model, which defines a logical framework for state transitions by narrowing down potential activities that can follow a given action, thereby enhancing the accuracy of activity prediction. However, these approaches are heavily depending on the segmentation process which is a tedious task on the FMCW radar data. The second category of approaches processes continuous radar data in a single step. For instance, Satyapreet et al. [8] applied a sliding window over the velocity and acceleration vectors of each target, which were then fed into a Long Short-Term Memory (LSTM) model. However, the sliding window method has limitations, as it doesn't account for transitions between events, leading to windows that may contain multiple activities and complicating accurate classification. Another method in this category, proposed by Prachi et al. [7], combines an unscented Kalman filter with an LSTM neural network to predict a corrected augmented state, comprising both localization parameters and a probability vector for various human activities. This approach is particularly effective in correcting model outputs during transitions between successive events, where confusion is most likely to occur. Additionally, Shrestha et al. [6] proposed using Bidirectional LSTM (Bi-LSTM) neural networks to process incoming radar data. Unlike standard LSTM models, Bi-LSTM can capture dependencies from both past and future data, and the authors demonstrated that Bi-LSTM applied to time-Doppler radar data outperformed a simple LSTM model.

A common limitation in the studies mentioned is the lack of a comparative analysis between model performance in classification versus detection scenarios when a model is deployed in an environment that is different to that used to acquire training data. Evaluating a model on data from a different environment enables to better assess the generalization capabilities of the HAR models. This research proposes a novel approach to HAR that enhances CNN based classifier probability outputs with simple probabilistic reasoning taking into account activity transition probabilities. The latter enables fitting a detection within a broader context. The proposed approach intentionally employs a relatively simple CNN architecture, as the limited amount of available data is insufficient to train more complex models with recurrent layers which would easily lead to overfitting.

In the next section, we will outline the methodologies used in this study, followed by a detailed description of the experimental setup. The results and discussion will be presented in the final two sections.

2 Methods

In this section, first an overview is provided about the radar signal processing that is required to obtain time-Doppler feature maps. It is then explained how CNNs utilize these features to continuously predict the activity class probabilities. Subsequently, a post-processing procedure based on simple probabilistic reasoning is explained to clean up the activity estimates. Finally, the metrics that will later be used for performance assessment are explained.

2.1 Radar signal processing

A standard signal processing pipeline is applied to extract time-Doppler features. The raw IQ data is processed by a 2D Fast Fourier Transform (FFT) to derive a range-Doppler matrix per antenna pair. Next, all range-Doppler maps are accumulated resulting in a single range-Doppler matrix with an improved Signal to Noise Ratio (SNR). Subsequently, the static object reflections are removed by subtracting a running mean value over the Doppler axis. The latter range-Doppler map is transformed to a logarithmic scale with radix 10. Then, it is further processed by taking the maximum over the range dimension to end-up with a Doppler vector. This process is repeated for each newly received radar pattern (a so called frame). As a result a continuous stream of Doppler vectors is generated. Some example time-Doppler streams for a number of human activities are shown in figure 1.

2.2 Convolutional neural networks

There are now a plethora of deep learning architectures that have been developed for a wide range of applications and problem domains. These include CNNs, LSTM models, and Bi-LSTM networks, amongst others. While LSTM networks can be effective for processing time-series data like radar features, they typically require a sizable amount of training data to be optimized well. Since we do not have access to a sufficient volume of data in this study, we opted to use a relatively simple CNN model.

CNNs are a popular choice for classification and detection tasks across diverse applications. They combine automatic feature extraction from the input using kernel convolutions with classification using fully connected layers in a single model. This model can be trained in an end-to-end fashion. As the input passes through different convolutional layers, the CNN model can learn different features, from low-level features in early layers to more complex ones in deeper

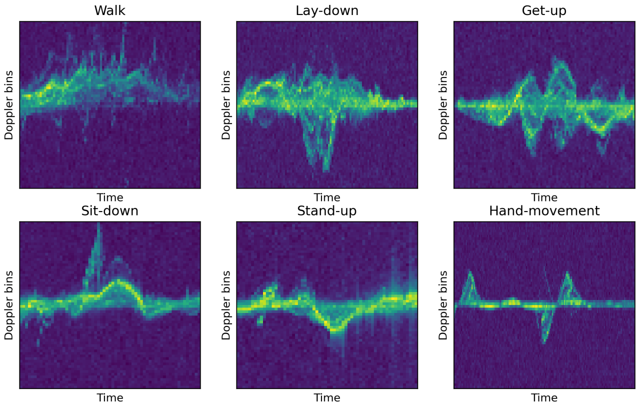

Fig. 1. Time-Doppler maps of different activities.

layers. Features can describe patterns across both the Doppler and time dimension. In this work the CNN model will estimate a vector with activity (class) probabilities based on the contents of a sliding window (with a certain time horizon, e.g. 2s) that moves over the continuous stream of Doppler vectors.

2.3 Post-processing

As the estimated class probabilities are likely to be noisy, as indicated earlier, an FSM-based post-processing method is proposed.

This method processes the data by weighing the class probabilities with activity transition probabilities. In this way impossible activity transitions (e.g. walk directly after sleeping without getting-up in between) are filtered out and less likely sequences of activities are less likely to be predicted. For this purpose, transition probabilities are represented by a finite state machine which is used together with the sequence of predicted activity probability vectors (output of CNN) in the Viterbi algorithm to predict the most likely sequence of activities. Note that this work targeted an online method which can only observe past measurements which matches the properties of the Viterbi algorithm.

Finite state machine (FSM) [4] represent a fundamental concept in automata theory. They can be used to represent systems with discrete and deterministic behavior. The key characteristics of an FSM are the finite states, the initial states, the inputs and transitions which define rules to move from one state (activity in our case) to another based on the inputs. FSMs find widespread usage in different applications ranging from text processing and pattern recognition to digital circuit design.

The Viterbi algorithm [2] is a dynamic programming algorithm used for calculating the most likely sequence of hidden states (activities) from a sequence of observations (activity probabilities). It is used frequently in the context of hidden Markov models. It calculates the most likely sequence of activities. It does this by recursively considering the current observation and the probabilities from the previous state, and the transition probabilities. To find the most likely sequence of states, it then works backwards through these calculated probabilities.

In this work, we propose the following procedure: A CNN translates the sequence of Doppler vectors (generated for each radar frame) into a sequence of activity probability vectors. A sliding window of a specified length (e.g., 40 frames) then moves across this sequence with a defined hop size (e.g., 10 frames). Each sliding window is processed by the Viterbi algorithm to identify the most likely sequence of activities. For the Viterbi algorithm, a trellis diagram is constructed, where each column represents the possible activities at a specific time frame, along with their corresponding probabilities. The number of columns in the trellis corresponds to the number of activity probability vectors within the sliding window. In Figure. 2 an example trellis diagram is given. Considering a matrix $T \in \mathbb{R}^{c,n}$ with c the number of activities and n the number of time frames the values in T can be calculated using the following equation:

$$T_{i,j} = \max_k (T_{k,j-1} A_{k,j} P_{i,j}), \tag{1}$$

where $T_{i,j}$ indicates the probability of being in state i at time step j, $\forall k, T_{k,0} = 1/c$, $A_{k,j}$ is the transition probability from state k to state j and $P_{i,j}$ is the probability generated by the CNN activity classifier for event i at time step j. The transition probabilities $A_{i,j}$ are expected to satisfy the constraint $\forall j, \sum_{i=1}^{c} A_{i,j} = 1$. Note that is process is repeated after shifting the sliding window with a certain hop size.

Once the trellis data structure T has been constructed for the given sliding window a backward trace is used to determine the most likely sequence (path through the trellis diagram) of activities.

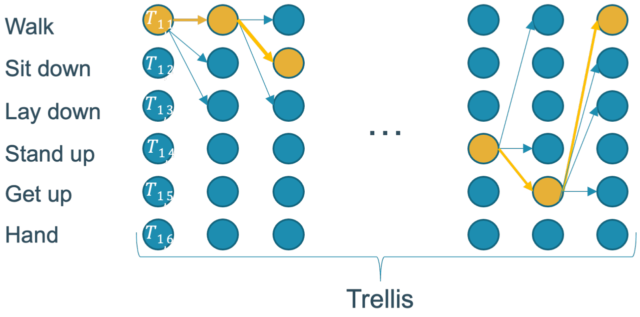

Fig. 2. Example trellis diagram with some best path found with the Viterbi algorithm.

2.4 Activity-based metrics

To evaluate the proposed detection methodology, an activity-based evaluation method was used[4]. In such method, the subjectivity of the temporal boundaries is alleviated by allowing some degree of misalignment called a collar between the compared activities in the reference and system output [5]. An activity instance is counted as a true positive if it has the same label as the corresponding reference activity, and its temporal boundaries lie within the permitted temporal collar with respect to the reference activity. The condition is used here for both onset and offset. Fig. 3 illustrates the possible cases when both onset and offset conditions are used with a collar of x ms. Activity-based evaluation is more intuitive for humans interpreting the result of an activity detection system, because it expresses the performance in terms of activity instances being detected.

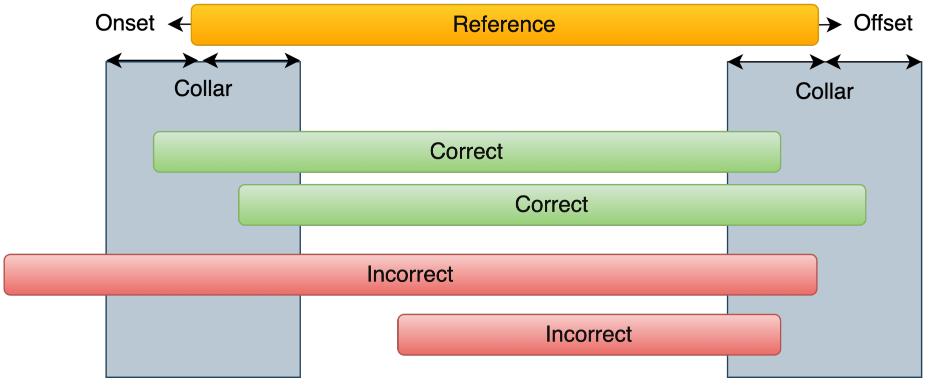

Fig. 3. Temporal misalignment's between ground truth and model predictions.

Using the previous methodology a confusion matrix with True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) can be calculated. Based on these the recall, precision and F1-score can be calculated as follows.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}},$$
$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

[4] Note that an activity typically lasts longer than the duration of the sliding window. As a result, a single activity will span multiple CNN predictions, allowing these predictions to be evaluated individually. In activity-based evaluation, individual predictions are first combined into distinct activity events, which are then compared to the ground truth.

3 Experimental setup

In this section, we first provide a brief overview of our HAR dataset, followed by an explanation of our experimental setup.

3.1 Data set

The dataset was collected in two distinct environments designed to simulate different settings. Environment A, resembling a hospital room, measured 2.3 m x 5.8 m x 3.25 m and was equipped with medical beds, a table, and medical equipment. The radar system was installed on the ceiling, providing coverage of the room. The two beds were positioned at a 45° angle relative to the radar's line of sight, with a chair and table setup placed between them, allowing for varied activity scenarios. Environment B was arranged to represent an ambient assisted living situation, with similar dimensions of 2.3 m x 5.8 m x 3.25 m. This room featured a sofa, TV, a table with four chairs, and a sink, creating a more cluttered and spatially constrained environment compared to Environment A. The radar was again mounted on the ceiling, capturing data across a room filled with more furnishings and reduced open space, which introduced additional complexity to the activity scenarios.

Participants in both environments were asked to perform a sequence of activities without specific instructions on how to execute them, ensuring no concurrent activities occurred. Each activity was annotated using camera images, marking the on- and offset times on the time-Doppler maps. This setup allowed for detailed analysis of activities considered being: walk, sit-down, lay-down, stand-up, get-up and hand-movement. Each had varying numbers of occurrence recorded in each environment:

- Environment A: In this setting, 10 participants each performed 4 different scenarios, repeating each scenario 5 times. Every scenario consisted of a series of activities. The dataset includes a total of 246 "walk" activities, 96 "sit-down," 96 "stand-up," 42 "hand-movement," 108 "lay-down," and 108 "get-up" activities.
- Environment B: In this setting, 5 participants each performed a sequence of activities, repeating them 5 times. This dataset contains 78 "walk" activities, 46 "sit-down," 48 "stand-up," 25 "hand-movement," 25 "lay-down," and 24 "get-up" activities.

3.2 CNN classifier

The CNN architecture used in this work consists of 4 convolutional layers with an increasing number of filters (8, 16, 32, 64 respectively), followed by max pooling along the frequency axis for the first 3 layers. After the convolutional layers, there are 2 fully connected layers with 32 perceptrons each, and the output layer uses a softmax activation for multi-class classification probabilities. The model was trained using categorical cross-entropy loss and the Adam optimizer, with an initial learning rate of 0.0001 that was dynamically reduced during training. Regularization techniques such as dropout, batch normalization, and L2 regularization were used to prevent overfitting. Importantly, the main purpose of this work is to compare the classification and detection performance with and without post-processing, so the CNN classifier model was kept fixed across all experiments to ensure that any differences in performance could be attributed to the post-processing techniques rather than variations in the underlying model.

3.3 Finite state machine

In order to enable FSM-based post-processing an FSM containing the transition probabilities between activities is defined. The FSM is given in (Fig. 4). For demonstration purposes the transitions between the activities were defined using prior knowledge about the activity sequences that occurred during the data collection campaign. For instance, the transition from lay-down to walk was not present during the data collection since a person first stands up before the person starts walking. Although in practice after walking a person can make hand movements this transition was not added to the FSM as it did not occur in the data set. The transition probability of the self-loops were set to a relatively high value of 0.8 since an activity is more likely to be followed by itself in the next frame, given that each radar frame and hence corresponding probability vector corresponds to a time window of 50 ms.

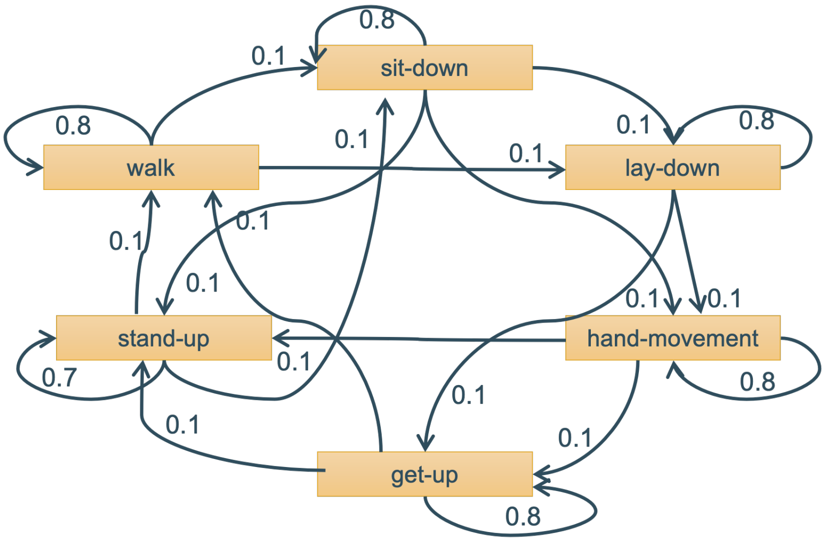

Fig. 4. State automata for human movement events

4 Results and Discussion

The proposed methodology was evaluated in a cross-domain setting, wherein the model is initially trained using data from one environmental context and subsequently assessed on a distinct, unobserved environment. This approach allows to better gauge the model's generalizability and robustness when confronted with novel environmental conditions. Table 1 shows classification and detection results. It reports the mean and standard deviation over 10 runs for each case. The classification results are obtained when prior knowledge about the onset and offset times is available for the CNN model. Detection results show the performance of the algorithm when it is fed with a continuous stream of Doppler vectors without prior knowledge concerning the onset and offset times of activities. Note that, as indicated before, for the latter approach collars were used to allow some acceptable misalignment between the ground truth and predicted onset and offset times of activities. More specifically, a tolerance of 5 radar frames (0.25 seconds) for both the on- and offset times was allowed. As a baseline, a method that applied majority voting on a sliding window of 40 CNN model output vectors with a hop size of 40 was used. For the proposed detection method a sliding window size of 100 frames and a hop size of 40 frames was used. A hop size of 40 was selected as this is close to the shortest event (sit-down) which has an average duration of approximately 30 frames. Analysis of the results reveal that the F1 scores obtained for both environmental contexts (AAL and Hospital) drop when comparing the classification performance with the baseline detection performance. However, when historical context is added by the simple Viterbi based reasoning approach that is proposed the performance again is comparable or slightly better compared to the classification performance.

Furthermore, a systematic ablation study was conducted to evaluate the impact of sliding window lengths for the Viterbi algorithm implementation, as well as different hop sizes between consecutive sliding windows. In table 2 the results indicate that for both the AAL and Hospital scenarios, the best F1 performance is achieved with a sliding window size of 100 and a hop size of 40.

5 Conclusion

In this work we proposed a method that enhances FMCW based HAR by refining classifier outputs with simple probabilistic reasoning taking into account activity transition probabilities. The approach demonstrated a slight improvement in the F1-score when compared to classification tasks which do not take into account information about prior activities. The latter, for example, enables implausible human activity transitions to be eliminated. In this study the HAR detection models were evaluated on data that was recorded in a different environment compared to that used to generate the training data. While not explained in this article this causes the performance to drop compared to when the model is evaluated on data from the same environment. Future work will study means to make HAR detection methods more robust to changes in the environment.

Table 1. This table shows both classification and detection results. For each result an evaluation was performed on data from a setting from which no data was used during training. With respect to detection the proposed method was compared to a baseline that only uses majority voting to merge CNN predictions to activity events prior to perform an activity-based assessment. For detection a sliding window of 100 frames and a hop size of 40 frames was used.

	F1	**Recall**	**Precision**
Environment/type	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$
AAL/classification	72.7 ± 4.4	73.7 ± 3.7	**75.6 ± 2.4**
AAL/detection-baseline	69.5 ± 0.2	**79.6 ± 0.3**	62.0 ± 0.4
AAL/detection-proposed	**75.1 ± 0.2**	78.8 ± 0.2	71.8 ± 0.2
Hospital/classification	80.1 ± 2.8	82.3 ± 2.5	**83.4 ± 2.9**
Hospital/detection-baseline	74.3 ± 0.1	84.3 ± 0.1	66.4 ± 0.1
Hospital/detection-proposed	**80.7 ± 0.3**	**85.3 ± 0.1**	77.3 ± 0.8

Table 2. Ablation study concerning the impact of the sliding window and hop size on the performance of the proposed method.

		F1	**Recall**	**Precision**
Window Size	**Hop Size**	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$
AAL				
50	20	67.8 ± 0.11	84.3 ± 0.15	56.7 ± 0.10
100	10	62.6 ± 0.10	84.1 ± 0.16	49.9 ± 0.07
100	20	70.4 ± 0.08	84.3 ± 0.12	60.4 ± 0.07
100	**40**	**75.1± 0.17**	78.8 ± 0.26	71.8 ± 0.15
150	20	67.8 ± 0.11	84.3 ± 0.15	56.7 ± 0.10
Hospital				
50	20	73.0 ± 0.04	85.5 ± 0.12	63.7 ± 0.03
100	10	65.9 ± 0.02	86.4 ± 0.05	53.3 ± 0.03
100	20	72.5 ± 0.04	86.2 ± 0.06	62.6 ± 0.04
100	**40**	**80.7 ± 0.31**	85.3 ± 0.15	77.3 ± 0.79
150	20	72.9 ± 0.05	85.3 ± 0.12	63.7 ± 0.03

Acknowledgment

The work was supported by project NextPerception funded from H2020 ECSEL-2019-2-RIA Joint Undertaking (GA No. 876487, and CZ GA No. 8A20007, by project DistriMuSe (No 101139769) funded from HORIZON-KDT-JU-2023-2-RIA Joint Undertaking (GA No. 101139769).

References

1. Moeness G Amin and Ronny G Guendel. Radar classifications of consecutive and contiguous human gross-motor activities. *IET Radar, Sonar & Navigation*, 14(9):1417–1429, 2020.
2. G.D. Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.
3. Sung-wook Kang, Min-ho Jang, and Seongwook Lee. Identification of human motion using radar sensor in an indoor environment. *Sensors*, 21(7), 2021.
4. D. Lee and M. Yannakakis. Principles and methods of testing finite state machines-a survey. *Proceedings of the IEEE*, 84(8):1090–1123, 1996.
5. Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. Metrics for polyphonic sound event detection. *Applied Sciences*, 6:162, 2016.
6. Aman Shrestha, Haobo Li, Julien Le Kernec, and Francesco Fioranelli. Continuous human activity classification from fmcw radar with bi-lstm networks. *IEEE Sensors Journal*, 20(22):13607–13619, 2020.
7. Prachi Vaishnav and Avik Santra. Continuous human activity classification with unscented kalman filter tracking using fmcw radar. *IEEE Sensors Letters*, 4(5):1–4, 2020.
8. Satyapreet Singh Yadav, Shreyansh Anand, Adithya M D, Dasari Sai Nikitha, and Chetan Singh Thakur. tinyradar: Lstm-based real-time multi-target human activity recognition for edge computing. In *2024 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2024.