tinyRadar: LSTM-based Real-time Multi-target Human Activity Recognition for Edge Computing

Satyapreet Singh Yadav, Shreyansh Anand, Adithya M D, Dasari Sai Nikitha, Chetan Singh Thakur
{satyapreets, shreyansha, adithyamd, dasaris, csthakur}@iisc.ac.in
Department of Electronic Systems Engineering, Indian Institute of Science, Bangalore, India, 560012

***Abstract*—Wireless Human Activity Recognition (HAR) has emerged as a vital technology with wide-ranging applications, including healthcare, aged care, and child monitoring. Radar-based HAR systems, grounded in electromagnetic principles, offer resilience to lighting variations and uphold user privacy by efficiently processing sparse point cloud data. These systems demonstrate robust performance even in obstructed environments. Nonetheless, existing radar-based HAR methods face a limitation in relying on fixed time windows for data classification. This approach may not be the most adaptable, especially when monitoring individuals of various ages, from children to the elderly, who perform activities at different speeds. This paper introduces "tinyRadar," a novel system that capitalizes on the capabilities of the Texas Instruments IWR6843 radar for sensing and the Raspberry Pi 4 for executing Long Short-Term Memory (LSTM) inference. tinyRadar is trained on activities of varying durations, enabling it to cater to different human activity speeds. Remarkably, tinyRadar achieves 93% real-time inference accuracy in recognizing eight distinct activity classes, classifying each activity frame within 10 ms, with a compact model size of 311 KB.**

***Index Terms*—human activity recognition, IWR6843 radar, Raspberry Pi 4, long short-term memory, edge computing**

I. Introduction

There is a growing demand for a real-time surveillance system that enables continuous monitoring of human activities. This necessity arises from the desire to enhance the safety and well-being of vulnerable populations, including the elderly and individuals with physical challenges who reside independently in their homes [1], [2]. Additionally, such a system serves as a valuable tool for detecting any unauthorized human presence within restricted areas. Vision-based systems offer the highest accuracy but raise privacy concerns [3]. Additionally, these systems involve substantial computational intricacies and are susceptible to issues like partial obstruction [4]. In contrast, wearable device-based systems provide the benefit of privacy preservation. However, their suitability for long-term activity recognition applications is challenged by inherent limitations. These include the risk of device loss, maintenance demands, limited battery life, and the potential for wearer discomfort [5], [6].

In contrast to various other sensor technologies, radar-based systems harness electromagnetic principles unaffected by factors such as lighting conditions and other fluctuations [7]. The unique ability of high-frequency radar waves to penetrate materials like curtains, paper, fog, and smoke ensures their resilience in obstructed environments, making them particularly robust [8]. These radar systems also prioritize user privacy by efficiently processing sparse point cloud data, all while maintaining high computational efficiency [9], [10]. Recent technological advancements in the semiconductor sector have enabled the miniaturization of mmWave radar hardware, resulting in smaller and more compact solutions [11]. Coupled with substantial enhancements in machine learning algorithms, these systems have become highly compatible with resource-constrained hardware, making mmWave radar an appealing option for implementing HAR systems.

Many studies within the realm of mmWave radar-based HAR systems have primarily focused on the extraction of features related to range, velocity, and angular information. For instance, research using Deep Convolutional Neural Networks (DCNNs) has employed 2D features like micro-doppler spectrograms [9] or 3D point cloud data [12]–[14] for HAR classification. It's worth noting that in CNN-based approaches, the observation window is typically fixed, and this fixed window must accommodate a wide range of activity speeds. As a result, it may require a longer observation window to capture slow-paced activities, such as when an elderly individual is going to bed, in contrast to activities occurring over shorter time intervals, like a fall event, which necessitates a shorter observation window. Researchers have also investigated SlowFast CNN networks to handle activities of varying speeds, where a single frame is processed by two CNN networks, one for a longer window and another for a shorter window. While this approach improves accuracy, it comes with increased computational complexity and memory usage [15]. Alternatively, some studies have investigated using CNNs for spatial decoding, followed by Temporal Convolutional Networks (TCN) or LSTM for temporal decoding to handle activities of different speeds. This approach processes Range-Angle and Range-Velocity spectrograms with CNNs and uses LSTM to decode frame-to-frame temporal relations [16], [17]. However, this model also tends to be memory-intensive.

In contrast to CNNs, LSTM networks excel in capturing temporal dependencies among events, making them well-suited for understanding the time-dependent aspects of human activities. Prior studies have employed LSTM or CNN-LSTM techniques for HAR, which utilize time sequences, such as micro-doppler spectrograms, often overlooking either the angle or velocity information. Also, they can be resource-intensive and incompatible for edge computing [18], [19]. Additionally, LSTM-based classification on point cloud data

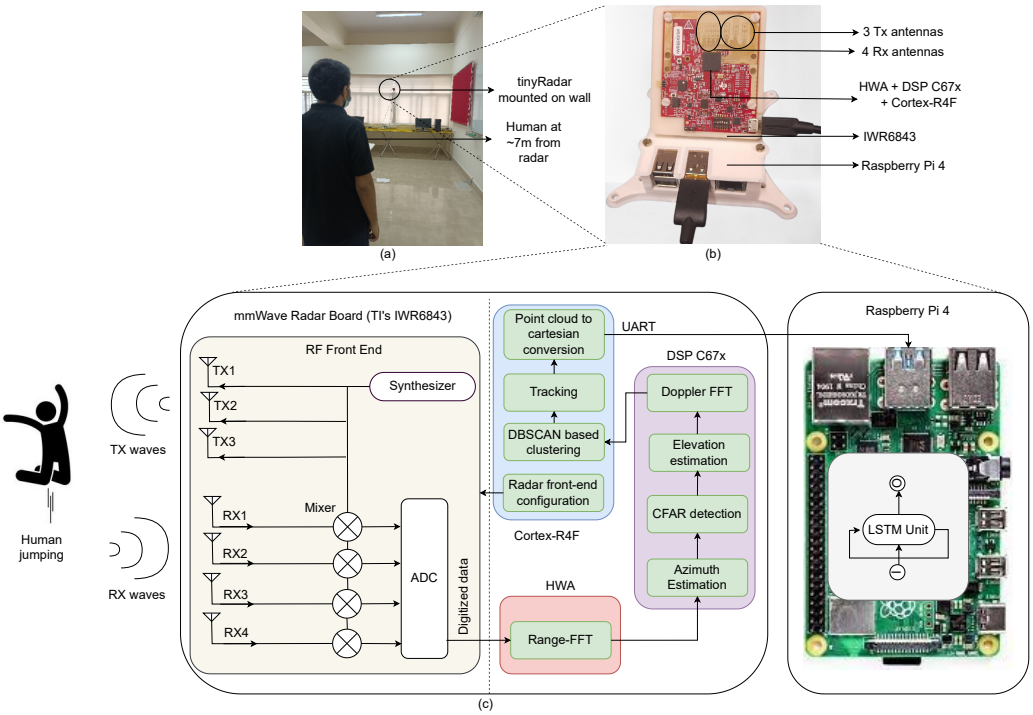

Fig. 1. Block diagram of tinyRadar based HAR system (a) tinyRadar mounted on a wall with human standing at rest. (b) tinyRadar comprising of IWR6843 radar board connected via a USB cable to Raspberry Pi 4 module. (c) Block diagram indicating the RF front-end of the radar board, followed by on-board signal processing for feature generation. The LSTM-based inference is performed on Raspberry Pi 4 by transferring data over the UART interface.

is also conducted, but with fixed time windows [13].

In this study, we propose "tinyRadar", a novel system that combines the IWR6843 [20] radar sensor with the Raspberry Pi 4 [21] to facilitate the process of multi-target HAR through the implementation of LSTM networks. The radar board perceives the surrounding environment and supplies velocity and acceleration data for each target across x, y, and z axes, encapsulated within a 6-dimensional feature fed to the LSTM-based inference network. Our inference implementation achieves an accuracy of 93% on eight different activity classes using a model of size 311 KB implemented on the Raspberry Pi 4 module. Furthermore, the radar system offers precise range data for all identified targets along the x, y, and z axes, facilitating accurate localization in tandem with the activity classification process.

The paper is structured as follows: Section II details mmWave radar-based feature generation. Section III offers insights into the collected dataset. Section IV outlines the classification model. Section V presents the results, and Section VI offers the paper's conclusion.

II. Radar Based Feature Generation

Figure 1 provides an overview of the tinyRadar system's operation. It illustrates the generation of Frequency-Modulated Continuous Wave (FMCW) signals, or chirps, via a synthesizer, followed by their transmission through three transmit antennas. The IWR6843 radar board comprises several key components, including the radar front-end, dedicated accelerators to perform Fast Fourier Transform (FFT), such as the Hardware Accelerator (HWA), a Digital Signal Processor (DSP C67x), and the Cortex®-R4F microcontroller unit. The radar system operates in Time Division Multiplexing Multiple Input Multiple Output (TDM-MIMO) mode, where each transmit antenna transmits sequentially, while all four receive antennas simultaneously capture signals, effectively creating a total of 12 antenna elements. These received signals are mixed with transmit replicas to produce Intermediate Frequency (IF) signals. The IF signals consist of summed sinusoidal signals whose frequency and phase contain valuable information about the target's range and velocity. Each IF signal is digitized using an onboard Analog-to-Digital Converter (ADC) and stored as a radar cube matrix. The initial processing phase involves applying FFT to the samples of each IF signal, determining the range of targets, known as Range-FFT, and this operation is executed within the HWA of the radar board, as described in [22].

Following this, Capon beamforming is utilized to create Range-Azimuth heatmaps. Subsequent processing includes a two-pass Constant False Alarm Rate-Cell Averaging (CFAR-CASO) operation conducted along the range and angle axes. This step localizes the detected targets. Capon Beamforming is performed once again for each point detected in the Range-Azimuth heatmap for elevation estimation, and the strongest signal is selected as the detected elevation angle. Doppler estimation is accomplished by applying FFT across the range bins to determine the velocity of the targets, considering each detected range, azimuth, and elevation point. All the operations, from Capon beamforming to point cloud generation, are implemented on the DSP C67x.

The resulting point cloud is clustered using an onboard

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm, and each cluster is tracked using Kalman filtering. The resulting point cloud information, consisting of range, azimuth, elevation, and velocity data, is transformed into nine detection points corresponding to each target by the Cortex®-R4F MCU. These nine features encompass range along the x-axis (r_x), range along the y-axis (r_y), range along the z-axis (r_z), velocity along the x-axis (v_x), velocity along the y-axis (v_y), velocity along the z-axis (v_z), acceleration along the x-axis (a_x), acceleration along the y-axis (a_y), and acceleration along the z-axis (a_z). These nine features are transmitted to the Raspberry Pi 4 module via Universal Asynchronous Receiver Transmitter (UART). The subsequent classification involves LSTM on six features, capturing velocity and acceleration, while localization utilizes range information.

III. Dataset Description

tinyRadar is positioned at a height of approximately 2.5 meters, tilted at an angle of 15 degrees for data collection. It has an azimuth Field of View (FOV) of ± 60 degrees and an elevation FOV of ± 20 degrees. The radar is configured to detect objects within a range of around 7 m, with a range resolution of about 7.5 cm, based on the chirp configuration specified in TABLE I.

TABLE I
CHIRP PARAMETERS

Parameter	Value
Starting frequency	60.75 GHz
Maximum range	7.28 m
Range resolution	7.58 cm
Maximum velocity	4.62 m/s
Velocity resolution	9.62 cm/s
Periodicity	55 ms
Ramp slope	54.71 MHz/μs
Sampling frequency	2.95 Msps
Number of samples per chirp	96
Number of chirps	96
Idle time	30 μs
ADC start time	25 μs

The dataset used for this study encompasses eight distinct activities: **F**alling, **G**etting up from bed, **G**oing to bed, **J**umping, **S**itting down, **S**tanding up, **W**alking, and **R**est. These activities were chosen for their representation of common daily movements essential to routines. The inclusion of falling is crucial, considering the heightened vulnerability of the elderly to such incidents. Data collection involved two participants, each performing these activities at their own pace. Ten minutes of data were recorded for each participant in each class, resulting in a total of 2.35 hours of total data. During data collection, participants executed the activities at various distances from the radar, from different angles, and at different speeds.

Following data collection, the gathered data was transferred from the radar board to a local PC for subsequent offline processing, which included training and analysis. Among the nine available features, namely r_x, r_y, r_z were used for spatial localization, and v_x, v_y, v_z, a_x, a_y, and a_z were used for activity classification. After data cleaning procedures, 769 data samples were generated with activity length varying from 20 frames (20 x 55 ms = 1.1 s) to 96 frames (96 x 55 ms = 5.3 s).

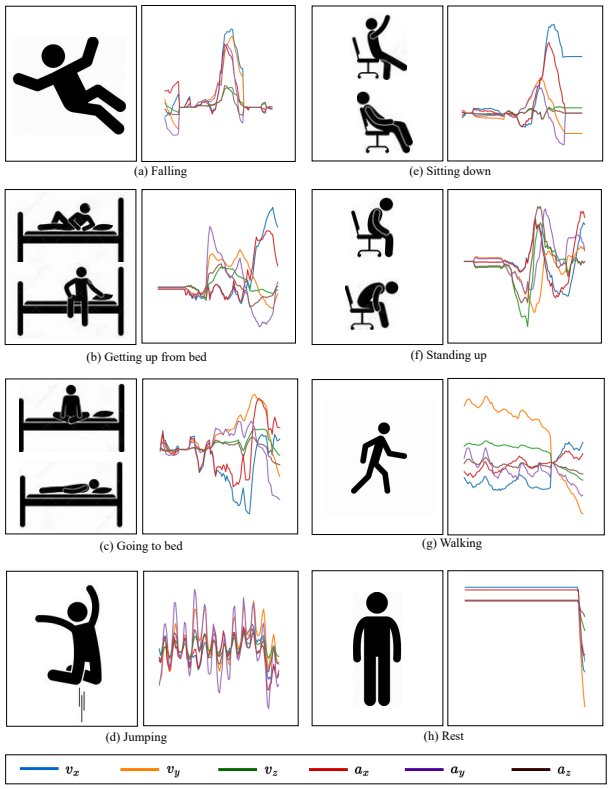

Fig. 2. Activity types and their feature maps: Illustration of activity performed by the user (left) and corresponding feature maps, each comprising of v_x, v_y, v_z, a_x, a_y, and a_z. The activities are as follows: (a) **F**alling (b) **G**etting up from bed (c) **G**oing to bed (d) **J**umping (e) **S**itting down (f) **S**tanding up (g) **W**alking (h) **R**est.

IV. LSTM-based Classification Engine

As can be seen from Figure 3, the classification engine consists of a single-layer Bidirectional LSTM (Bi-LSTM) with a total of 96 hidden units. The utilization of bidirectional processing allows the model to capture dependencies in both forward and backward directions of the input sequence, enhancing its understanding of the temporal aspects of the data. Following the Bi-LSTM layer, there is a fully connected layer that includes eight output neurons for making predictions.

In the variable-length training process of the network, a series of six consecutive frames is employed as input, with each frame consisting of six features. This input setup is structured as a tensor with dimensions of [1, 6, 6] at each time instant. For each training data sample, the model's loss is calculated using cross-entropy loss, measuring the dissimilarity between predicted class probabilities and the actual class. This loss is accumulated over each frame for the corresponding activity class. Subsequently, the Backpropagation Through Time (BPTT) algorithm is applied to compute gradients against the accumulated loss and propagate them

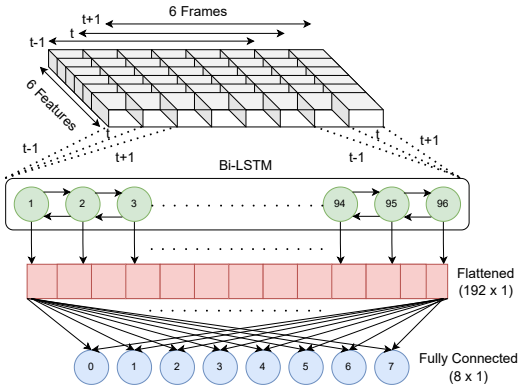

Fig. 3. LSTM Model Architecture

backwards through the network. The total data samples were divided into 80:20 training and testing splits, and the model was trained using PyTorch [23] in a stateful manner, wherein the hidden state and cell state from the previous time instant were passed on to the current time instant. Adam optimizer is utilized, with a learning rate of 0.01, to iteratively adjust the model's weights and minimize the loss for 50 epochs. During real-time inference, Raspberry Pi 4 reads UART data from the radar board and passes the six features into the trained PyTorch model for prediction. As seen from Figure 4, the predicted activity is triggered when the confidence in the activity gradually increases and exceeds 95%.

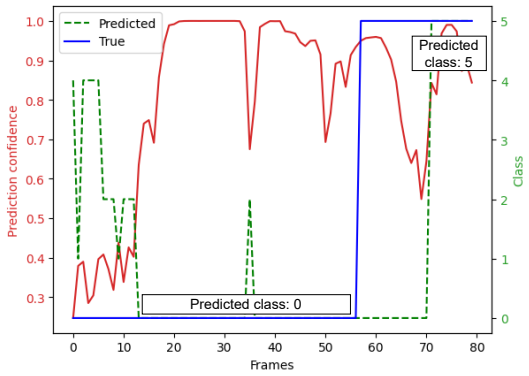

Fig. 4. In real-time, the LSTM classifier outputs the predicted class and confidence, gradually triggering a prediction when confidence surpasses a set threshold.

V. Results

The signal processing chain responsible for generating point cloud data is executed on the IWR6843 board. It takes approximately 36 ms to process each frame, with a CPU utilization of 72% on the IWR6843 board, as described in [22]. On the Raspberry Pi 4, the velocity and acceleration parameters are stored in a buffer of length six and then relayed to the inference engine for classification. The network exhibits the capability to classify each input data point in $\sim$ 10 ms, resulting in low-latency inference with a CPU utilization of 20% on the Raspberry Pi 4. A representation of the classifier's output for a test pattern can be observed in Figure 5, where the predicted output tracks the true label with confidence of >95%.

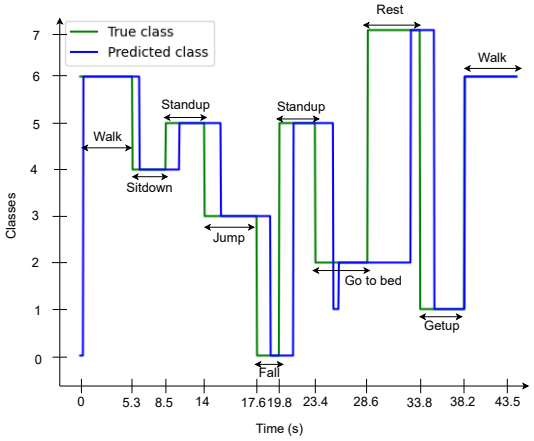

Fig. 5. LSTM-based classifier output for a test pattern. True class, and predicted class for each time instant are shown in green, and blue respectively. The points on the x-axis represent the start and end time of each activity in the test pattern.

During multiple people activity classification, the LSTM-based inference engine achieved a real-time classification accuracy of 93% on average. Each participant was localized with the recognized activity indicated beside it in the GUI for visualization, as seen from Figure 6.

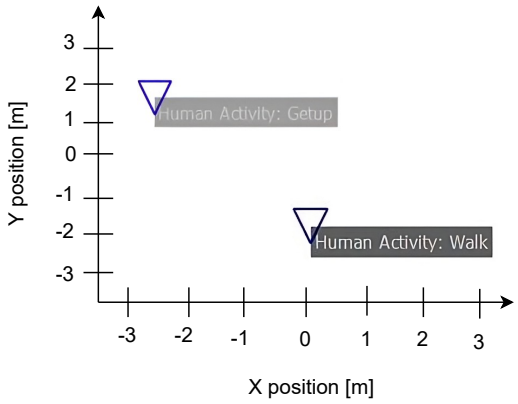

Fig. 6. 2D visualization of the target locations within a room w.r.t the radar. Simultaneously, on the right side, the detected activity identified by the LSTM-based inference engine is displayed.

VI. Conclusions

Our proposed system, tinyRadar, integrates radar data from the IWR6843 sensor with the processing capabilities of the Raspberry Pi 4, incorporating LSTM networks for HAR. This innovative design allows for efficient recognition of human activities, achieving an impressive real-time classification accuracy of 93% across eight different activity classes for multiple people. The system also provides accurate localization information for detected targets, further enhancing its utility. As a versatile system, tinyRadar opens up new possibilities for continuous, unobtrusive monitoring of human activities, contributing to the well-being and security of individuals in a variety of settings. In the future, our plan is to expand the dataset and transfer the LSTM inference engine onto the radar board, enabling end-to-end computing on a single platform.

REFERENCES

[1] J. Muangprathub, A. Sriwichian, A. Wanichsombat, S. Kajornkasirat, P. Nillaor, and V. Boonjing, "A novel elderly tracking system using machine learning to classify signals from mobile and wearable sensors," *International Journal of Environmental Research and Public Health*, vol. 18, no. 23, p. 12652, 2021.

[2] N. Farber, D. Shinkle, J. Lynott, W. Fox-Grage, and R. Harrell, "Aging in place: A state survey of livability policies and practices," 2011.

[3] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *Ieee Access*, vol. 6, pp. 14 410–14 430, 2018.

[4] A. Carfi, C. Motolese, B. Bruno, and F. Mastrogiovanni, "Online human gesture recognition using recurrent neural networks and wearable sensors," in *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2018, pp. 188–195.

[5] G. M. Balbim, I. G. Marques, D. X. Marquez, D. Patel, L. K. Sharp, S. Kitsiou, and S. M. Nyenhuis, "Using fitbit as an mhealth intervention tool to promote physical activity: potential challenges and solutions," *JMIR mHealth and uHealth*, vol. 9, no. 3, p. e25289, 2021.

[6] A. M. Kasparian and S. M. Badawy, "Utility of fitbit devices among children and adolescents with chronic health conditions: a scoping review," *Mhealth*, vol. 8, 2022.

[7] Q. Chen, Y. Li, Z. Cui, and Z. Cao, "A hand gesture recognition method for mmwave radar based on angle-range joint temporal feature," in *IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2022, pp. 2650–2653.

[8] M. Eggimann, J. Erb, P. Mayer, M. Magno, and L. Benini, "Low power embedded gesture recognition using novel short-range radar sensors," in *2019 IEEE SENSORS*. IEEE, 2019, pp. 1–4.

[9] S. S. Yadav, R. Agarwal, K. Bharath, S. Rao, and C. S. Thakur, "tinyradar: mmwave radar based human activity classification for edge computing," in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2022, pp. 2414–2417.

[10] S. S. Yadav, R. Agarwal, K. Bharath, S. Rao, and C. Singh Thakur, "tinyradar for fitness: A contactless framework for edge computing," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 17, no. 2, pp. 192–201, 2023.

[11] V. Clarkson and B. Ng, "A radar revolution," *IEEE Potentials*, vol. 38, no. 4, pp. 6–8, 2019.

[12] C. Yu, Z. Xu, K. Yan, Y.-R. Chien, S.-H. Fang, and H.-C. Wu, "Noninvasive human activity recognition using millimeter-wave radar," *IEEE Systems Journal*, vol. 16, no. 2, pp. 3036–3047, 2022.

[13] A. D. Singh, S. S. Sandha, L. Garcia, and M. Srivastava, "Radhar: Human activity recognition from point clouds generated through a millimeter-wave radar," in *Proceedings of the 3rd ACM Workshop on Millimeter-wave Networks and Sensing Systems*, 2019, pp. 51–56.

[14] Z. Meng, S. Fu, J. Yan, H. Liang, A. Zhou, S. Zhu, H. Ma, J. Liu, and N. Yang, "Gait recognition for co-existing multiple people using millimeter wave sensing," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 849–856.

[15] B. Sheng, Y. Bao, F. Xiao, and L. Gui, "Dyliteradhar: Dynamic lightweight slowfast network for human activity recognition using mmwave radar," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[16] M. Scherer, M. Magno, J. Erb, P. Mayer, M. Eggimann, and L. Benini, "Tinyradarnn: Combining spatial and temporal convolutional neural networks for embedded gesture recognition with short range radars," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10 336–10 346, 2021.

[17] Y. Zhang, L. Peng, G. Ma, M. Man, and S. Liu, "Dynamic gesture recognition model based on millimeter-wave radar with resnet-18 and lstm," *Frontiers in Neurorobotics*, vol. 16, p. 903197, 2022.

[18] A. Shrestha, H. Li, J. Le Kernec, and F. Fioranelli, "Continuous human activity classification from fmcw radar with bi-lstm networks," *IEEE Sensors Journal*, vol. 20, no. 22, pp. 13 607–13 619, 2020.

[19] Y. Sun, R. Hang, Z. Li, M. Jin, and K. Xu, "Privacy-preserving fall detection with deep learning on mmwave radar signal," in *2019 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 2019, pp. 1–4.

[20] T. Instruments, *IWR6843 Single-Chip 60-to-64GHz mmWave Sensor*, 2021. [Online]. Available: https://www.ti.com/lit/ds/symlink/iwr6843.pdf?ts=1697781295835&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FIWR6843

[21] R. P. Foundation, *Raspberry Pi 4 Datasheet*, 2020, https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi_DATA_2711_1p0_preliminary.pdf.

[22] T. Instruments, *3D People Counting Demo Software Implementation Guide*, 2021, https://e2e.ti.com/cfs-file/__key/communityserver-discussions-components-files/1023/3D_5F00_people_5F00_counting_5F00_demo_5F00_implementation_5F00_guide.pdf.

[23] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf