Radar-Based Automotive In-Cabin Occupancy Detection in Driving Scenarios: An Edge-AI Powered Algorithm with Its Real-Time Implementation

Muhammet Emin Yanik[1], Anand Dabak[2], Slobodan Jovanovic[3], Anil Mani[4], Sandeep Rao[5]
Radar Systems and Algorithms R&D, Texas Instruments
[1234]Dallas, Texas, US, [5]Bangalore, India
[1]m-yanik@ti.com, [2]dabak@ti.com, [3]sjovanovic@ti.com, [4]a-mani@ti.com, [5]s-rao@ti.com

Abstract—**This paper proposes a novel millimeter-wave (mmWave) radar-based automotive in-cabin monitoring solution utilizing the combination of advanced signal processing and edge-AI techniques to accurately detect occupants in dynamic driving scenarios and ensure reliable decisions in unseen environments. By leveraging the high-resolution capabilities of mmWave radar, this study aims to provide a robust, cost-effective, and non-intrusive solution compared to traditional technologies like weight sensors. We present a detailed discussion on the end-to-end processing chain with its optimized implementation on Texas Instruments (TI) AWRL6844 mmWave sensors for real-time occupancy detection. Experimental results and benchmarks demonstrate the effectiveness of our approach with >98% occupancy detection accuracy in driving car scenarios using a single mmWave sensor and a real-time implementation that achieves a 5Hz frame rate. Additionally, we introduce the dataset, comprising numerous real-world driving scenarios, utilized for model training and validation. This paper also provides an overview of the custom-built toolset developed to support the entire classification workflow and address a scalable framework to solve further challenges for in-cabin sensing, such as occupant classification (child vs. adult, etc.).**

Keywords— Millimeter-wave radar, radar signal processing, tinyML, edge-AI, automotive in-cabin occupancy detection.

I. Introduction

Automotive in-cabin occupant detection systems, such as seat belt reminders and child presence detectors, have become essential safety features in modern vehicles [1][2]. The main requirements expected from these systems include reliable detection and localization of any life in the car, accurate identification of the detected occupant in any seat, and timely alerting mechanisms to notify the car's central processing units with low latency to take necessary safety-critical actions. These systems must operate under various conditions, such as different passenger sizes and postures, and environmental factors like temperature and lighting. Besides, due to cost concerns in the automotive industry, a single sensor performing all these mentioned features is desired to provide a cost-effective solution while maintaining comprehensive in-cabin safety monitoring. Recent advancements in frequency-modulated continuous wave (FMCW) millimeter-wave (mmWave) radar technology offer numerous advantages over conventional passive sensors to address such requirements and enhance automotive in-cabin occupancy detection tasks [3][4].

Child presence detection systems target scenarios where the vehicle is stationary, focusing on the safety of children left unattended in parked cars [5][6]. On the other hand, radar-based detection and localization in moving vehicles present unique challenges compared to stationary scenarios, particularly for seat belt reminder systems. In a moving car scenario, the radar system must contend with a dynamic environment, where relative motion between the radar and target objects inside the vehicle introduces varying signal reflections that may diverge from the signal model assumed at the algorithm design stage. The dynamic road and driving conditions that impact the stability and accuracy of reflected radar signals typically create unwanted clutter, complicating the identification of occupant presence. Therefore, this paper primarily focuses on seat belt reminder applications to ensure reliable occupancy detection and localization in moving car scenarios.

In radar-based applications, especially in the edge-AI domain, building solutions solely based on machine learning techniques often falls short of robustness due to the extensive data requirements and, more importantly, the complexity of models needed to generalize across diverse scenarios. To understand the underlying features in every corner case, a deeper network and a vast amount of data are needed to achieve a certain level of accuracy and robustness [7][8]. Since such high-accuracy deep learning models become memory- and compute-intensive, they are typically infeasible for tiny edge processors. Due to the cost-related concerns in the automotive industry, it is highly desirable to solve this problem using the resources embedded in a radar system-on-chip [9] without the need for a high-end external processor.

Conversely, model-based signal processing offers the advantage of leveraging domain-specific knowledge and deterministic physics-aware algorithms to enhance signal interpretation and noise reduction, hence the overall performance and robustness, which is crucial in safety-critical applications. However, such model-based methods [10] alone may not be sufficiently adaptive to cover the full spectrum of scenarios encountered in dynamic environments due to many unknowns in the underlying model assumptions.

Therefore, in this paper, we propose a hybrid approach that harnesses the strengths of machine learning and model-based signal processing to achieve a >98% accuracy level and enable a computationally efficient, lightweight solution for edge-AI deployment. This combined strategy not only offers more accurate decisions with reduced complexity but also enhances generalization across varying operational contexts, ultimately ensuring a reliable radar-based occupancy detection system.

II. Proposed Solution

In the overall processing chain, we propose a two-stage algorithm. The first stage (i.e., low-level detection layer) involves processing analog-to-digital converter (ADC) data (i.e., the beat signal) received from the radio-frequency (RF) front-end and generating the point cloud, representing the spatial information of living objects from the radar signal reflections in a three-dimensional (3D) environment. In the second stage (i.e., high-level decision layer), the point cloud data undergoes various pre-processing and feature extraction blocks to enhance the quality and relevance of the information. Subsequently, the features extracted from the refined point cloud data are fed into a lightweight classifier model, which performs the classification task to decide whether the corresponding seat is occupied or not. This two-stage approach leverages the strengths of both advanced signal processing and machine learning, ensuring robust and scalable performance for in-cabin occupancy detection applications and enabling an efficient implementation.

The diagram in Fig. 1 illustrates the main processing blocks and algorithm flow of the low-level detection layer. The range processing step (e.g., windowing and fast Fourier transform (FFT)) is first applied to the ADC samples of each chirp from each virtual antenna to create the range spectrum of the target scene. The static clutter removal step is then applied to remove the purely static objects from the scene and leave only the signals backscattered from the dynamic objects. A Capon beamformer [11] based on the steering vectors over a coarse two-dimensional (2D) azimuth-elevation angular grid (with 8° inter-bin resolution, by default) is applied to generate the 3D range-azimuth-elevation spectrum matrix (i.e., heatmap). In covariance matrix estimation, instead of using all the received chirps per transceiver pair at each frame, only the selected bins in the Doppler spectrum of the corresponding frame are used (i.e., low-pass filtering) to enhance the detection performance for highly stationary occupants. In the Capon beamforming stage, coarse angle steps are used in the azimuth-elevation grid to reduce the overall processing complexity and memory resources (which will be enhanced with an additional zoom-in stage, as discussed in the following paragraph). Note that to achieve a very fine velocity resolution with a limited number of chirps per frame (due to power consumption, frame rate, and regulatory requirements), multiple frames are aggregated in memory (four frames, by default) to be fed into the signal processing chain.

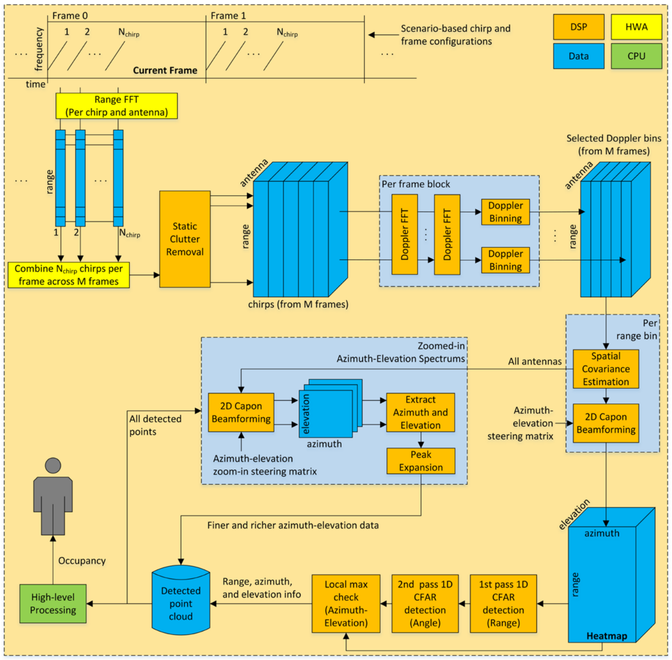
Frame 0
Frame 1
frequency
time
1
2
N_{chirp}
Current Frame
Scenario-based chirp and frame configurations
DSP
HWA
Data
CPU
Range FFT (Per chirp and antenna)
range
Combine N_{chirp} chirps per frame across M frames
Static Clutter Removal
antenna
chirps (from M frames)
Per frame block
Doppler FFT
Doppler Binning
Selected Doppler bins (from M frames)
Per range bin
Spatial Covariance Estimation
All antennas
Azimuth-elevation steering matrix
2D Capon Beamforming
Zoomed-in Azimuth-Elevation Spectrums
elevation
azimuth
Extract Azimuth and Elevation
Peak Expansion
Azimuth-elevation zoom-in steering matrix
All detected points
Finer and richer azimuth-elevation data
Heatmap
1st pass 1D CFAR detection (Range)
2nd pass 1D CFAR detection (Angle)
Local max check (Azimuth-Elevation)
Range, azimuth, and elevation info
Detected point cloud
High-level Processing
Occupancy

Fig. 1. The low-level detection algorithm inputs the ADC data from the RF front-end and generates the point cloud for the high-level processing flow.

A two-pass CFAR algorithm is then applied to the created 3D range-azimuth-elevation heatmap for detection. Before the CFAR detection, the generated 3D range-azimuth-elevation heatmap is first reshaped into a 2D range-angle heatmap with one dimension on the range and another dimension on the angle. The reshaped 2D range-angle heatmap is then processed by the first-pass range CFARs per angle bin to create the tentative detection points. A second-pass CFAR across the angle domain of the reshaped 2D range-angle heatmap then confirms the detection points created by the first-pass range CFARs.

After the two-pass CFAR logic, the second-stage Capon beamforming algorithm based on the steering vectors over a finer 2D azimuth-elevation angular grid around the detected coarse azimuth-elevation local peaks is applied to generate the zoomed-in 2D azimuth-elevation angle spectrums (with a 5x zoom-in factor, by default). To generate a dense point cloud, the strongest peak of each zoomed-in spectrum and its neighbors that meet specific criteria are then extracted as the final detected points (with finer azimuth and elevation angles). Each measurement vector ultimately generated by the detection layer represents a reflection point with range, azimuth, and elevation.

In an in-cabin occupancy sensing application, training a single model across all seats offers significant advantages in terms of simplicity and robustness. From a simplicity standpoint, employing a unified model streamlines the development and maintenance process by eliminating the need for multiple seat-specific models. In terms of robustness, a single model benefits from a broader and more diverse dataset, improving its ability to generalize across various seat positions and real-world scenarios. Thus, this uniformity not only simplifies the implementation but also enhances the overall consistency and reliability of the system's performance. On the other hand, when utilizing a single model, it is essential to implement carefully handled pre-processing steps for each seat's data to eliminate spatial domain biases.

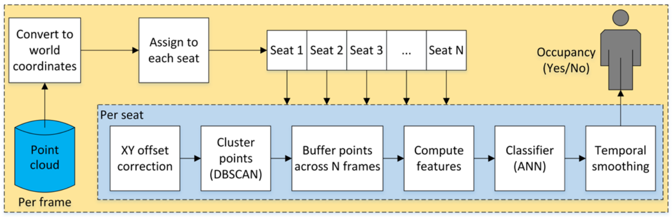

Fig. 2. The high-level decision algorithm inputs the point cloud and gives the final seat occupancy decision using the proposed feature extraction and classifier approaches.

As summarized in the algorithm flow in Fig. 2, each point generated by the detection layer per frame is first converted to the world coordinates, depending on the sensor setup, and then assigned to each vehicle seat. We then apply an offset correction (i.e., normalization) step in the XY-domain to bring the point cloud data from different seats into a common reference frame. As shown in Fig. 3, this simple pre-processing step is important for the unified single model to learn generalized patterns and make accurate predictions regardless of the seat position. In the proposed algorithm, instead of using the raw point cloud, we also employ DBSCAN clustering logic [12] to reduce outliers and achieve more robustness. As shown in Fig. 3, such a filtering process with the DBSCAN clustering results in a more reliable and cleaner dataset for model training and inference. After tuning the DBSCAN algorithm based on a typical in-cabin scenario, according to the statistics generated from the dataset, 95% of the points detected from real targets survived, while 55% of noisy points were eliminated.

In radar-based processing, observing the scene across time rather than relying on single-frame snapshots offers advantages for accuracy and robustness due to radar's known limitations, especially in the spatial domain. Hence, as another pre-processing step, we combine the refined point cloud per seat in a circular buffer across time. Such a temporal aggregation allows for capturing dynamic changes in the scene, enhancing the model's ability to discern between transient noise and persistent objects, hence leading to more stable and robust performance. Utilizing the point cloud buffered across multiple frames, we generate feature vectors based on some statistics and provide these vectors to the classifier model. In the feature extraction step, seven features in total are extracted: (1) the mean of the number of points, (2)-(4) the root-mean-square (RMS) of the points in the XYZ-domain, and (5)-(7) the standard deviation of the points in the XYZ-domain. After experimenting with some other statistical features on the XYZ-domain, including mean, peak-to-peak, kurtosis, etc., we conclude that using RMS and standard deviation yields the most accurate results.

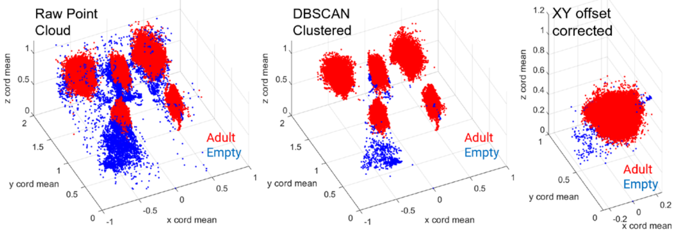

Fig. 3. The DBSCAN algorithm clusters points from real objects while eliminating noisy reflections, and the offset correction approach normalizes data in the XY-domain, enabling a single model for all seats.

Finally, a three-layer Artificial Neural Network (ANN, also known as Dense Neural Network) model, shown in Fig. 4, is run to decide occupancy utilizing the point-cloud-based features generated at each seat location. The model architecture comprises an initial layer with 20 nodes, followed by two hidden layers of 40 and 80 nodes. Each of these layers employs the ReLU activation function. The network's output layer is designed with a two-node softmax activation function tailored for binary classification. As shown in Fig. 4, at the final stage, we also apply a temporal smoothing logic for robustness and enhanced accuracy.

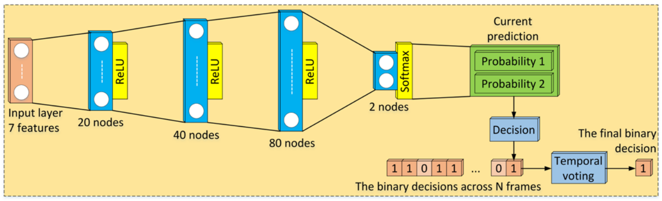

Fig. 4. The three-layer ANN model trained for binary classification.

Although we propose a binary classifier specifically trained to effectively distinguish between an empty and occupied seat to address the challenges associated with seat belt reminder systems, the versatility of our approach can be extended beyond this primary application. For example, the underlying methodology in the end-to-end framework can be adapted to support child presence detection systems (child vs. adult classification, etc.) by retraining a new classifier model in the corresponding scenarios. Furthermore, our technique can also be extended to multi-class classification for advanced occupancy detection problems, enabling the system to differentiate between various other occupant types, such as adults, children, babies, and pets.

III. Real-Time Implementation

We implement the occupancy detection algorithm proposed in Section II on the TI's latest high-performance and low-power mmWave sensor AWRL6844 [9], leveraging its advanced processing capabilities. The AWRL6844 mmWave sensor features three distinct processing cores: a hardware accelerator (HWA) running at 200MHz, a C66x digital signal processor (DSP) running at 450MHz, and an ARM R5F central processing unit (CPU) running at 200MHz. As illustrated in Fig. 5, the HWA is employed for range processing interleaved with the chirp acquisition period to create the 3D radar cube in the range-chirps-antenna domain. Once the radar cube is created and located in memory, the C66x DSP handles all the advanced signal processing blocks in the low-level detection layer in Fig. 1 to generate the point cloud. The point cloud is then passed to the ARM R5F core, which runs all the high-level decision logic in Fig. 2.

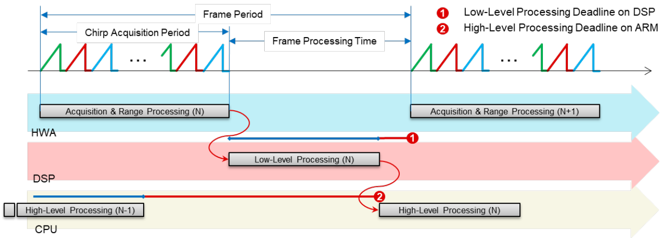
Frame Period
Chirp Acquisition Period
Frame Processing Time
1 Low-Level Processing Deadline on DSP
2 High-Level Processing Deadline on ARM
Acquisition & Range Processing (N)
Acquisition & Range Processing (N+1)
HWA
Low-Level Processing (N)
DSP
High-Level Processing (N-1)
High-Level Processing (N)
CPU

Fig. 5. The task model of the implementation across three cores.

Architecting the code to utilize these three independent cores optimally maximizes efficiency and performance, ensuring no cores remain idle. Such a distribution of tasks ensures that each core operates within its optimal performance domain, maximizing the overall throughput for efficient real-time implementation and achieving certain required frame rates. This implementation approach also demonstrates the potential of the AWRL6844 mmWave sensor for sophisticated algorithms needed in occupancy detection applications.

In the end-to-end processing chain, the low-level detection layer in Fig. 1 takes around 30-40 ms to process with the chirping parameters given in Section V, depending on the number of points generated for that particular frame (for around 100-300 points, respectively). The high-level decision layer in Fig. 2 takes around 10-20 ms to process, similarly, depending on the number of points. Considering the 156 ms chirp acquisition time detailed in Section V, the end-to-end chain is optimized to run efficiently on AWRL6844 at a 5Hz frame rate, thanks to the optimized task model across processing cores, as illustrated in Fig. 5.

The classifier model in Fig. 4 has a memory footprint of around 17.5KB for its parameters and requires an additional 1.5KB of heap memory for internal computations. The feature extraction block in Fig. 2 requires around 94KB of memory to buffer point clouds across frames for five zones, storing up to 3200 total points per zone with compressed XYZ coordinates (2 bytes per coordinate) and an additional 12KB of heap memory for internal computations. On the other hand, the low-level detection layer in Fig. 1 needs 512KB of memory to store the radar cube (4 bytes per sample) with the chirping parameters given in Section V and around 284KB of heap memory for internal computations. In summary, the total memory footprint of the end-to-end chain is optimized to fit into the DSP L2 RAM, R5F RAM, and L3 shared RAM of the AWRL6844 [9].

IV. A Custom-Built Toolset for Development

It is well-known that the edge-AI-powered techniques, as proposed in this paper, are all data-driven and need massive data from various real-world scenarios for training to address potential corner cases in complex environments. However, capturing and labeling radar data presents unique challenges compared to camera images or videos. Radar data is inherently more abstract and lacks visual cues, making it difficult to interpret and annotate. Hence, these challenges demand domain-specific expertise to identify and label objects accurately and necessitate specialized frameworks and tools to streamline and enhance the accuracy of data capturing and labeling for radar applications.

To address these challenges, this paper utilizes an end-to-end MATLAB-based custom-built toolset that streamlines data acquisition, annotation, feature extraction, and model training for radar-based automotive in-cabin occupancy classification tasks. The developed framework, summarized in Fig. 6, first facilitates the efficient collection of high-quality radar point cloud data from the AWRL6844 mmWave sensor and ensures seamless annotation, feature extraction, and model training to support the entire classification workflow. To simplify the processes of data collection and annotation, we implement a filename-based labeling approach, where the label information is embedded within the filenames of captured data. This metadata is then automatically extracted during preprocessing, enabling efficient and automated annotation without the need for manual intervention. For the model training, the MATLAB deep learning toolbox is utilized. The classifier model is implemented as bare-metal C code. However, a model to header file generation framework is also developed to facilitate the automated maintenance of the learnable parameters.

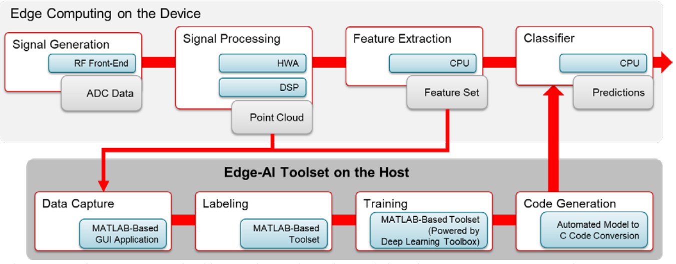

Fig. 6. The custom-built toolset developed in the MATLAB environment to support the end-to-end occupancy detection classifier workflow.

V. Dataset and Experimental Results

Diverse data collection from a wide range of scenarios is paramount in machine learning-based model training to ensure that the trained models can generalize effectively across various conditions, thereby improving their robustness and reliability. For this purpose, we created an extensive dataset on multiple car platforms by encompassing different driving environments, occupant postures, seat configurations, and external factors such as weather and road conditions. In this scope, we utilized the TI AWRL6844 radar evaluation module (EVM) [13] to capture data, ensuring a comprehensive dataset for developing and testing our solution.

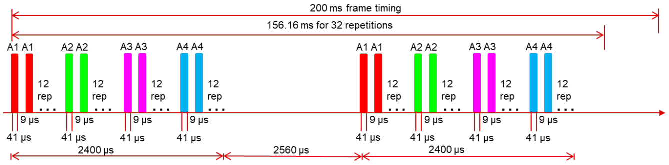

Fig. 7. The chirping scheme at a 5Hz frame rate to achieve optimal occupancy detection performance and meet FCC regulations.

The chirping scheme in FMCW radar processing is pivotal in enhancing overall performance while satisfying the various other design metrics. While the resolution and accuracy of the detections are directly influenced by the characteristics of the FMCW chirp design (e.g., bandwidth, chirping duration, etc.), compliance with the existing regulations, such as those imposed by the Federal Communications Commission (FCC) [14], is also essential for maintaining the integrity of radar systems in real-world applications. In the data capture campaigns, we carefully consider all these aspects of the chirping scheme to achieve optimal performance and ensure that it adheres to such regulatory standards.

The device is configured to transmit FMCW signals at a 57 GHz start frequency with a ramp slope of 97 MHz/μs. 128 ADC samples per chirp are configured with a sampling frequency of 4.4 Msps in real baseband. Therefore, a maximum range of 2.7 m is achieved with around 5 cm range resolution. As shown in Fig. 7, each transmit antenna is configured to transmit 12 consecutive chirps. These closely grouped chirps per transmit antenna are accumulated before the range processing to achieve good signal-to-noise ratio (SNR) performance while reducing the processing overhead in the low-level detection layer. After around 2.6 ms of idle time, this scheme is then repeated 32 times per frame. Therefore, at the end of each frame, a block of 32 chirps is created per transmit antenna. As mentioned in Section II, four frames are then concatenated to be used in the low-level detection layer processing in Fig. 1. The orthogonality between the transmit antennas is achieved by employing the time division multiplexing (TDM) technique.

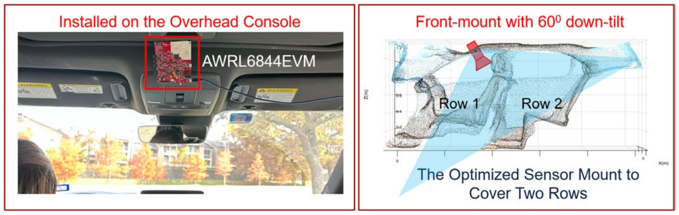
Installed on the Overhead Console
AWRL6844EVM
Front-mount with 60⁰ down-tilt
Row 1
Row 2
The Optimized Sensor Mount to Cover Two Rows

Fig. 8. The sensor mounting configuration to the overhead console.

We designed and executed over 210 scenarios across different dates, representing various real-world situations (all in dynamic driving conditions). The data collection spanned three different car models (Mercedes E-350, Tesla Model-Y, and Acura-MDX), providing a diverse range of in-cabin environments and seat configurations. To enhance the generalizability of our solution, we involved more than 10 participants of varying sizes and postures, simulating a realistic spectrum of occupant behaviors. As illustrated in Fig. 8, the radar sensor was mounted near the overhead console with a 60° down-tilt angle, optimizing its field of view for effective in-cabin monitoring, utilizing a single radar sensor. For data augmentation to address class imbalance, we use random noise injection into the input features.

TABLE I. Performance Results of the Proposed Algorithm

Scenario	Precision	Recall	Accuracy	F1-Score
(a)[1]	99.29%	99.29%	99.28%	99.28%
(b)[1]	97.95%	99.00%	98.74%	98.46%
(c)[1]	97.57%	99.04%	98.69%	98.28%
(d)[2]	92.80%	96.71%	96.06%	94.55%
(e)[2]	95.71%	97.99%	97.66%	96.79%
(f)[2]	95.64%	94.26%	95.09%	94.84%

[1] Performance results of the models trained, validated, and tested using the data from the same car model: (a) Mercedes (M) data only, (b) Tesla (T) data only, (c) Acura (A) data only.

[2] Performance results of the models trained and validated using two car models but tested with a separate car model: (d) Trained with (M) and (T), tested with (A), (e) Trained with (M) and (A), tested with (T), (f) Trained with (A) and (T), tested with (M).

In the first set of results shown in TABLE I. (a)-(c), each car's data is separated from the others to understand the accuracy of a per-car trained network. Each data set (from each car) is then split into 40% training, 10% validation, and 50% testing. Then, only the test accuracies are summarized. Even though half of the data set is reserved for testing (unseen by the model), we achieved a >98% accuracy for each car. In the second set of results shown in TABLE I. (d)-(f), we applied a robustness test to check how the model is generalizable in unseen environments. In these tests, we separated each car data set, but now we use two of them in training and validation, and the other one in the test. Similarly, we only summarized the test accuracies. As shown in these results, even in an unseen car environment, we could achieve around 96% mean accuracy across each combination.

In the results shown in Fig. 9, all the available data (from Mercedes, Tesla, and Acura) are merged and split into 40% training, 10% validation, and 50% testing. Then, the test accuracies across different window sizes are shown for 2-layer and 3-layer ANN models. This result demonstrates that utilizing a longer observation window enhances the performance. However, this improvement comes with a tradeoff regarding increased decision latency, which should be carefully managed to balance real-time responsiveness with detection accuracy. Besides, the result in Fig. 9 shows that the 3-layer ANN slightly outperformed the 2-layer counterpart (where the last hidden layer in Fig. 4 is removed).

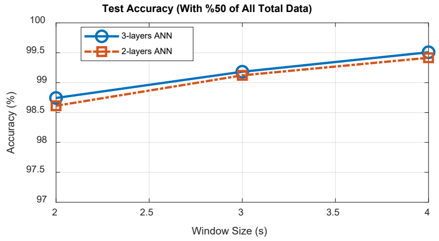

Fig. 9. The test accuracies of classifier models with different window sizes.

Finally, in Fig. 10, various example scenarios tested with the real-time demo running on the TI AWRL6844 EVM are depicted. As shown in these snapshots, different seat occupancy scenarios are detected successfully. It is important to note that all the test cases shown in Fig. 10 are created in driving scenarios. Besides, to emphasize that the car environment in this test is new to the model and has never been seen in the training process.

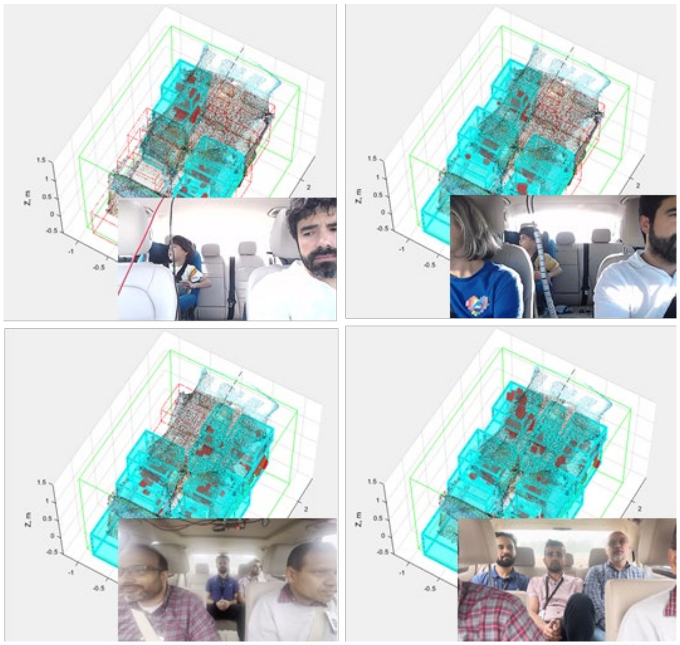

Fig. 10. The test results of the real-time chain (on the AWRL6844) in different driving scenarios (in new environments that are not used in model training).

VI. Conclusions

In this paper, we proposed a mmWave radar-based reliable automotive in-cabin occupancy detection solution that combines signal processing and machine learning techniques. Implementing a three-layer ANN model fed by hand-crafted features from advanced signal processing blocks offered reduced complexity and resulted in accurate and reliable occupancy detection across various real-world scenarios. We demonstrated the advantages of training a single, unified model for all seats with proper spatial-domain normalization steps, highlighting the benefits in terms of simplicity and robustness. The integration of temporal data aggregation to capture dynamic changes in the scene and DBSCAN clustering-based cleaning techniques to effectively mitigate the noise in the data further enhanced the overall system accuracy and robustness. This paper also presented the details of the proposed chain's real-time implementation on the TI AWRL6844 mmWave sensor, leveraging its three processing cores to achieve an efficient real-time performance. Additionally, we introduced our dataset, which comprises numerous real-world scenarios, and presented a custom-built classifier framework supporting the entire classification workflow. The proposed method in this paper also sets the stage for our future innovations in radar-based automotive in-cabin occupancy detection solutions.

Acknowledgement

The authors wish to thank all the colleagues at Texas Instruments for their support in the data capture campaign.

References

[1] X. Zeng et al., “In-vehicle sensing for smart cars,” in IEEE Open Journal of Vehicular Technology, vol. 3, pp. 221-242, 2022.

[2] A. Gharamohammadi et al., "In-vehicle monitoring by radar: A review," in IEEE Sensors Journal, vol. 23, no. 21, pp. 25650-25672, Nov.1, 2023.

[3] N. Munte et al., “Vehicle occupancy detector based on FMCW mm-wave radar at 77 GHz,” in IEEE Sensors Journal, vol. 22, no. 24, pp. 24504-24515, Dec.15, 2022.

[4] H. Song et al., “In-vehicle passenger detection using FMCW radar,” Proc. Int. Conf. on Information Networking (ICOIN), Jeju Island, South Korea, 2021, pp. 644-647.

[5] A. R. Diewald et al., “RF-based child occupation detection in the vehicle interior,” 17th Int. Radar Symp. (IRS), Krakow, Poland, 2016, pp. 1-4.

[6] N. M. Z. Hashim et al., “Child in car alarm system using various sensors,” ARPN J. Eng. Applied Sci., vol. 9, no. 9, pp. 1653–1658, 2014.

[7] H. Abedi et al., “Deep learning-based in-cabin monitoring and vehicle safety system using a 4-D imaging radar sensor,” in IEEE Sensors Journal, vol. 23, no. 11, pp. 11296-11307, Jun. 2023.

[8] S. Lim et al., “Deep neural network-based in-vehicle people localization using ultra-wideband radar,” IEEE Access, vol. 8, pp. 96606–96612, 2020.

[9] Texas Instruments, “AWRL6844, automotive single-chip high-performance, low-power, 57 GHz to 64 GHz mmWave radar sensor,” [Online]. Available: https://www.ti.com/product/AWRL6844.

[10] A. Lazaro et al., “Seat-occupancy detection system and breathing rate monitoring based on a low-cost mm-wave radar at 60 GHz,” in IEEE Access, vol. 9, pp. 115403-115414, 2021.

[11] J. Capon, “High-resolution frequency-wavenumber spectrum analysis,” Proc. IEEE, vol. 57, no. 8, pp. 1408–1418, Aug. 1969.

[12] M. Ester et al., “Density-based spatial clustering of applications with noise,” in Proc. Int. Conf. Knowl. Discov. Data Min., 1996.

[13] Texas Instruments, “AWRL6844EVM, AWRL6844 evaluation module board,” [Online]. Available: https://www.ti.com/tool/AWRL6844EVM.

[14] Federal Communications Commission (FCC) – Appendix D. [Online]. Available: https://docs.fcc.gov/public/attachments/FCC-23-35A1.pdf.