IEEE Open Journal of the Computer Society

Received 28 March 2025; accepted 11 May 2025. Date of publication 20 May 2025;
date of current version 4 June 2025. The review of this article was arranged by Associate Editor Jiangtao Luo.

Digital Object Identifier 10.1109/OJCS.2025.3571587

Comparison of Deep Learning Techniques for RF-Based Human Posture Detection Systems

EUGENE CASMIN [1,2] (Graduate Student Member, IEEE), MIRIAM RODRIGUES[1,2], AMÉRICO ALVES [1,2], AND RODOLFO OLIVEIRA [1,2] (Senior Member, IEEE)

[1]Departamento de Engenharia Electrotécnica e de Computadores, Faculdade de Ciências e Tecnologia, FCT, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
[2]Instituto de Telecomunicações, 1049-001 Lisbon, Portugal

CORRESPONDING AUTHOR: EUGENE CASMIN (e-mail: e.owilla@campus.fct.unl.pt).

This work was supported byFCT - Fundação Para a Ciência e Tecnologia, I.P. under Project UID/50008 and Project 2022.08786.PTDC.

ABSTRACT This article focuses on techniques for a human posture classification framework that implements radio frequency (RF) active systems. In the first step, we describe the general approach considered for human posture classification. To this effect, we propose four different solutions: one based on traditional signal processing (SP) techniques, where the detection is centred around a correlation of prior classification masks; a second based on a hybrid SP and deep learning (DL) technique, where the DL model is trained with supervised data gathered at a single distance to the target; a third based on a hybrid SP and DL technique trained with data gathered at multiple distances to the target; and a fourth that uses variational auto-encoder (VAE) for feature generation. Their performance is then compared on the basis of classification accuracy and computation time. We show that although the SP-based solution presents high accuracy, the hybrid SP/DL solutions are advantageous in terms of classification accuracy and robustness at multiple distances, albeit requiring higher computation time. We further show the slight edge that VAE-based solutions have over plain DL solutions in terms of accuracy.

INDEX TERMS Human posture classification, signal processing (SP), machine learning.

I. INTRODUCTION

As technology advances, posture sensing has gained interest across various fields. Millimetre-wave radar technology has improved resolution and accuracy, enabling human movement detection. In particular, Frequency Modulated Continuous Wave (FMCW) radars are increasingly used for human posture recognition due to their low cost and discreet sensing capabilities [1], [2], [3], [4], [5], [6], [7], [8]. These active RF sensing systems are valuable for context awareness, with applications in smart homes, security, road safety, autonomous driving, healthcare, search and rescue, and sports. Their growing adoption is further driven by their flexibility in detecting distance, velocity, and object presence.

Machine learning (ML) has significantly transformed industries and remains highly relevant due to the exponential growth of data. RF sensing technologies, such as FMCW radar, collect high-accuracy data at high sampling rates but require learning tools to process and classify the information effectively. Integrating ML with RF sensing is promising, as RF sensing captures rich body motion and posture data, while ML enhances classification accuracy and efficiency.

A. STATE-OF-THE-ART

RF sensing for posture detection and classification has attracted increasing interest in recent years. The work in [1] introduces a feature-based gesture recognition system adopting an FMCW radar system. A Range Doppler Map (RDM) is obtained from raw signals of FMCW radar and a variety of features are generated from it.

A research study on gesture recognition using a 77 GHz FMCW radar system was presented in [2], where the system performed drivers' gesture recognition for an in-vehicle driver monitoring application by utilising micro-Doppler signatures. Due to the fact that these signatures were frequently distorted by the presence of multiple moving targets, and the goal was to recognise only the driver's hand gestures, range information

was included to filter out the influence of irrelevant moving targets. The authors proposed a method using five handcrafted features and a k-Nearest Neighbor (kNN) classifier, achieving an average prediction accuracy of 83.52%. However, for three of the gestures, the accuracy was 90%, while for the remaining two, which required more axial movement, yielded an accuracy of 63%. This is likely due to the radar's sensitivity to radial movements and lack of information on axial movements.

In [3], the authors presented a low-complexity multi-gesture classification solution for consumer and automotive electronics that employs a 77 GHz mm-wave low-power complementary metal oxide semi-conductor (CMOS) radar, as well as an innovative algorithm and software pipeline. The proposed solution enabled real-time gesture detection and used an artificial neural network (ANN) for classification, using magnitude and phase-based features, as well as statistical features. The work in [6] adopted FMCW radar to build a detection system that can distinguish between drones, humans, and cars. A range Fast-Fourier Transform (FFT) plot is created based on the radar signals, and the authors claim that this is the first project relying on range FFT features to perform target classification. Two lightweight classification models, namely logistic regression and naive bayes, are explored to assess their performance. The selected features fed to the ML algorithms were based on the peaks detected in those plots, some of which included width, area, and standard deviation of peak values. Overall, logistic regression and naive bayes algorithms yielded an accuracy value of 86.9% and 73.9%, respectively, and the researchers note that some future work could be done to improve target size and shape detection and that range-Doppler plots could be introduced for more reliable classification.

The work in [8] explored the importance of various features in classifying human mobility scenarios using an RF sensing system in the 76-81 GHz band. The authors compared the performance of different features computed from the RF time of flight and used four different methodologies to evaluate their importance. Overall, 126 raw features were parsed to each feature selection method, with kNN being used as a classifier. Despite all methodologies being capable of identifying features of lower importance, it was concluded that the recursive feature elimination method leads to better accuracy in classification when its 18 selected features are used for classification.

In [4], the authors proposed a method for recognising continuous human motion using radar technology, specifically a FMCW radar system. The proposed method, extracts relevant information from range-Doppler frames obtained from the radar's signals. Then, using a peak search method, each human motion is located and separated, and multidomain features are extracted. The time, range, Doppler, and Radar Cross Section features are fed to a kNN classifier, resulting in robust recognition accuracy of around 95% for single motions and 91.9% for continuous motions. The work in [9] presented an FMCW radar that uses micro-Doppler signatures for fall detection. The researchers extract micro-Doppler signatures from radar signals and use transfer learning (*AlexNet*), support vector machine (SVM), kNN, and Deep Learning (GoogleNet) algorithms to automatically extract features and perform classification, respectively. The three classifiers yielded an average test percentage accuracy of 78.25% for SVM, 77.15% for kNN, and 74.7% for GoogleNet. The use of radar-based sensors in short-range human monitoring applications, was also tackled in [5], where the authors proposed a stacked bidirectional long short-term memory (Bi-LSTM) model to address the temporal sequence nature of the data. The model achieved over 90% accuracy for 45 different activity sequences tested.

In [7], it is proposed a phase-extraction technique for an FMCW radar that allows the detection of target motions. The technique involves decomposing an FMCW chirp signal into multiple-frequency signals, 260, to be exact, to extract modulated phases due to target motions. This novel method can extract the phase without performing a FFT, which can overcome the range migration problem that causes errors in phase extraction using a conventional FFT-based method. It reduces noise and can improve detection accuracy for both human vital signals, which consist of small vibrations, and several body motions. Thus, phase detection can be precise regardless of the type of moving targets.

Variational auto-encoders (VAEs) can be used for human activity classification by learning compact, probabilistic representations of sensor data, which can help improve the accuracy and robustness of activity recognition models. VAE can be used for dimensionality reduction, as explored in the works in [10] and [11], that although not centred on human activity classification, adopt VAE for feature extaction and reduction.

B. CONTRIBUTIONS

Our work is centred around exploring an FMCW radar-based framework that can collect data frames from targets within its range and subsequently perform posture classification. We evaluate the real-time classification performance of various proposed methods addressing the same experimental scenario. Specifically, we analyze the performance of a signal processing (SP)-based approach and compare it with deep learning (DL)-based solutions using raw FFT data and features extracted via a VAE, in addition to the implicit dimensionality reduction. To the best of our knowledge, this is the first study to directly compare SP and DL-based solutions in terms of classification accuracy and computational performance for human posture classification in RF-based systems. The main contributions of this work are summarised as follows:

1) We conduct a systematic study comparing signal processing-based methods and deep learning-based classifiers for FMCW radar data, ensuring a fair evaluation under the same datasets and operational conditions.
2) Our proposed solution S_1 introduces an idle scene detection algorithm, an innovative approach to filtering irrelevant radar reflections, improving classification

accuracy and reducing the overall computation time by reducing the number of subsequent frames requiring marking further down the classification chain.

3) We evaluate the impact of raw data against synthetically derived features using VAEs, highlighting how feature learning influences model performance. This point also explores the effect of implicit dimensionality reduction on performance, further highlighting the novelty of our work.
4) We assess the trade-offs between classification accuracy and execution time, providing practical insights into real-world deployment feasibility.

This article distinguishes itself from other FMCW radar-based studies by providing a comprehensive benchmark of four different classification approaches. By evaluating accuracy and computation time under the same experimental conditions, our work establishes a comparative framework that is currently lacking in the literature. Additionally, we address the open question of whether raw data or generated features (from raw data) should be used in classification. As far as we know, no prior research has investigated this for human posture classification using FMCW radar. The results achieved in this work evidence that classification accuracy can be slightly improved when raw data is transformed into features before being fed into the classification model. Lastly, we explore the impact of using datasets collected at multiple distances from the FMCW device, a factor that has not been previously assessed in supervised learning models for this application. The results show that classifiers can benefit from training on such datasets.

The rest of this article is organised as follows: Section II introduces the system model preliminary work done on the data received from the radar sensor and the general steps required before running the human classification algorithms are presented in Section III. Section IV presents the four different solutions for human posture classification. The assessment of the solutions and the final remarks are presented in Sections V and VI, respectively. Further, the acronyms adopted in this work are presented in Table 1 to ensure clarity and consistency throughout the article.

II. SYSTEM MODEL

This section presents the framework setup, comprising the physical components of the framework and the rationale behind the decisions taken when setting up the RF-sensing-based data collection apparatus. We also describe the logical flow of the classification chain at a high level.

Fig. 1 illustrates the different scenarios adopted for classification in this study. Specifically, the two postures considered for classification rely solely on range information without involving motion. This approach allows for an adequate evaluation of the proposed techniques using low-cost FMCW devices, which do not provide Doppler radial speed information, but only the target's range information.

Fig. 2 shows the highest level of abstraction of the developed setup used to support this work. The setup comprises

TABLE 1. Table of Acronyms Adopted in This Work

Acronym	Definition
ANN	Artificial Neural Network
DL	Deep Learning
FFT	Fast Fourier Transform
FMCW	Frequency Modulated Continuous Wave
G_1	Dataset G_1; Single-distance
G_2	Dataset G_2; Multi-distance
IoT	Internet of Things
kNN	k-Nearest Neighbor
ML	Machine Learning
PCA	Principle Component Analysis
RDM	Range Doppler Map
RF	Radio Frequency
RX	Receiving/Reception antennae
S_1	Solution S_1; Mask Correlator
S_2	Solution S_2; Single-distance DL
S_3	Solution S_3; Multi-distance DL
S_4	Solution S_4; Single-distance DL + VAE
SMA	Simple Moving Average
SP	Signal Processing
SVM	Support Vector Machine
TX	Transmission antennae
VAE	Variational Auto-Encoder

(a) Idle. (b) Seated. (c) Standing.

FIGURE 1. Different postures classified in this work.

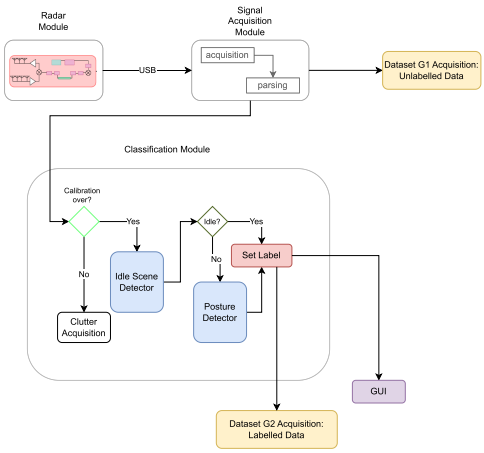


FIGURE 2. System model high-level view.

three important modules: the *radar*, the *signal acquisition* module, and the *classification* module. The radar and signal acquisition modules consist of the hardware used to obtain data frames. The radar itself has specific parameters that can be manipulated for different purposes. Table 2 presents the parameters set for this work. These parameters are inherently tied to the operational conditions in which the radar system operates, including factors such as the maximum range, the radar resolution (defined by the length of each FFT bin), and the number of samples collected over time. While these operational conditions dictate the configuration parameters, it

TABLE 2. Radar Parameters Employed in the Experimental Setup

Parameter	Value
Doppler Bins	16
Range Bins	256
Chirps per Frame	48
Start Frequency f_c	77 GHz
Bandwidth B	4 GHZ
Frame Period	250 ms
Chirp Slope	6 MHz/μs
Chirp Duration T_C	66.67 μs
ADC Sample Rate	2.727 MHz
Range resolution	0.043 m
Maximum Unambiguous Range	6 m
Maximum Radial Velocity	1 m/s
Velocity Resolution	0.13 m/s

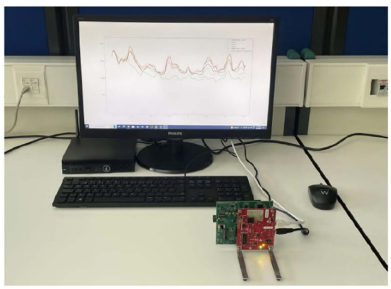


FIGURE 3. FMCW experimental setup.

is important to note that under different conditions, the output radar data may vary, but the techniques proposed in the article remain applicable. If the configuration is altered while maintaining the same classification scenarios, the conclusions of our experiments remain unaffected - for instance, a change in the sampling period does not impact the validity of the results. However, modifications to range-related configuration parameters could lead to a mismatch between the radar settings and the physical scenario being analyzed. In such cases, a decrease in performance may not stem from the proposed techniques themselves but rather from an inadequate radar configuration that fails to properly capture the relevant characteristics of the environment, such as an inappropriate range setting for the specific scenario used in the experiments.

The FMCW radar system setup, pictured in Fig. 3, operates by transmitting chirps from three TX antennas, resulting in periodic chirp sequences known as Doppler chirps. To collect a radar frame, 16 consecutive Doppler chirps are required. I/Q samples from each RX channel are compiled into a raw data matrix. A column-wise FFT is applied to each chirp, resulting in a matrix with rows representing different target ranges, referred to as range bins. A second FFT is applied to each row to obtain the RDM, allowing the calculation of the radial velocity of the targets. Each of the 16 columns of the RDM is denoted as a Doppler bin and corresponds to a specific Doppler shift. Notably, there are 256 FFT bins in each range profile, each corresponding to a different distance, with an approximate resolution of 0.043 m due to the maximum unambiguous range defined for this work. This study was conducted in a room measuring 4.02 m wide and 9.06 m long, using only the first 140 range bins for a range coverage of approximately 6.02 m. Radar samples were acquired, classified, and visualised in the context of this room.

When analysing the RDM instances received from the radar using the developed visualisation module, it was evident that among the 16 Doppler bins, Doppler bin 0, responsible for representing the distance information, was the most crucial for this study. Owing to the fact that the focus of this work was on identifying static postures rather than movements, the other bins, which gather radial velocity data towards the radar, were rendered redundant. Thus, for the remainder of this article, the term "frame" refers to the first 140 bins of Doppler bin 0.

The frames were hence sampled, adopting a period of 250 ms via a USB cable. After the sampling, a basic moving average was applied using a 5-frame window to filter noise, which revealed that using a rolling mean could be a valuable approach to reducing data fluctuations. Larger window sizes can provide smoother signals, but with a 250 ms frame collection rate, averaging 10 frames results in a 2.5 s delay in detecting posture changes. Thus, it was determined that a window size of k = 5 would be the most fitting for implementing a simple moving average (SMA).

Regarding the blocks in Fig. 2, after completion of the calibration phase during which reference clutter frames are acquired, the system begins classification, and the frames are directly sent to the Idle Scene detector. If a human is detected in the front of the radar, the frame is not labelled as *Idle*, and the system proceeds with the posture classification using one of the proposed solutions (among S_1 to S_4).

To evaluate the performance of the classification approaches presented in this work, it is required to know what labels are being assigned to the incoming radar frames as well as the computation time. Thus, dataset acquisition is also crucial during this stage, since labels, computation times, and frames can be saved into files for subsequent analysis. These files comprise dataset group G_2 . Consider a case where we would like to acquire frames for the *Standing* scenario at 1m using the classification solution S_3 . After the clutter frame is acquired, classification begins, and a subject enters the scene to perform the *Standing* pose. During acquisition, the subject stands 1 m away from the radar, in front of a chair that has been present in the scene since the beginning of the system's launch, also at 1 m. When the acquisition is complete, the dataset for that combination is saved in a file.

A total of 250 frames are recorded and stored in a file. To avoid capturing the moment where the subject is walking into the scene, the first 10 frames are discarded, which excludes approximately 2.5 s from the dataset. Thus, 240 frames are kept, which is equivalent to 1 minute. This file also contains the computation time associated with each frame as well as the label attributed to it.

III. METHODOLOGY

The classification process is illustrated in Fig. 4. The *initial calibration* procedure occurs once during the system's launch, where 40 radar frames are collected and averaged to create

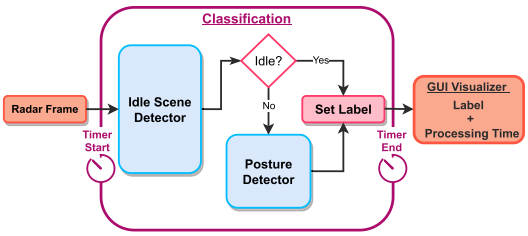


FIGURE 4. Block diagram summary for the classification process.

a designated $F_{clutter}$ frame. Then, the classification process begins, consisting of two parts: the *"Idle Scene Detector"* and the *"Posture Detector"*. As its name suggests, the $F_{clutter}$ frame is used during the latter in order to remove clutter from received frames.

The idle scene detector remains consistent across all solutions proposed for the posture detector and involves comparing incoming radar signals with the calibration frame. If a scenario does not correspond to the *Idle* scenario (pictured in Fig. 1(a)), the process moves on to determine human posture via the posture detector; otherwise, the scene is labeled as "Idle". The flow of operation behind classification is represented in Fig. 4, and the idle scene and posture detectors are presented in detail in Sections III-C and IV, respectively.

A. INITIAL CALIBRATION AND CLUTTER FRAME

This step precedes classification and is performed only once when the system is launched. The purpose of this procedure is to gather several frames in an empty scene, i.e., where no human subject is within the range of the radar. From this collection of frames, a "Clutter Frame" that represents the background scene. By removing it from the received radar frames, the subject in the scene is isolated and thus the posture should be easier to identify. Furthermore, detaching the subject from its surroundings should facilitate a context-independent classification. $F_{clutter}$ is represented as

$$F_{clutter} = \frac{1}{n_{calib}} \cdot \sum_{n=1}^{n_{calib}} S_{calib}^{(n)}, \tag{1}$$

where the number of collected frames during the calibration procedure is represented by $n_{calib} = 40$ and $S_{calib}^{(n)}$ is the n-th calibration frame.

B. DATA PREPARATION

This subsection covers data preparation, which comprises SMA operation, clutter removal, and normalization of the frames acquired from the radar.

When the posture detector receives a radar frame, the first operation that takes place is the SMA, where a window of 5 samples is considered ($K = 5$), which implies that the system stores the last 4 frames received in an array with the current frame and calculates the average of these 5 range arrays. This is described as:

$$S_{smooth} = \frac{S_t + S_{t-1} + S_{t-2} + \cdots + S_{t-K-1}}{K}, K = 5, \tag{2}$$

where S_t is the current raw data sample frame, and $K = 5$ is the window considered for the SMA. To complete the goal of removing the clutter from the S_{smooth} frame, a subtraction operation takes place, described as

$$S_{no\_clutter} = S_{smooth} - F_{clutter} \tag{3}$$

where S_{smooth} is the smooth radar frame obtained from the previous action and $F_{clutter}$ is the clutter frame. The clutter removal aims to achieve the objective of focusing on the subject in front of the radar rather than all targets within its range, which includes the background and other irrelevant elements. Lastly, $S_{no\_clutter}$ is normalised as follows

$$S_{normalised} = \frac{S_{no\_clutter}}{||S_{no\_clutter}||}, \tag{4}$$

where $S_{no\_clutter}$ is the frame obtained after the clutter has been removed. ||.|| represents the L2 norm of a vector given by

$$||V|| = \sqrt{\sum_{i=1}^{n} V_i^2}, \tag{5}$$

where V_i is the i-th element of the V array with n elements.

In this work, each of the classification solutions proposed in Section IV receives $S_{normalised}$, an FFT of 140 discrete points, as input, and each of them provides a way to label this data.

C. IDLE SCENE DETECTOR

As illustrated in Fig. 4, this is the first step of the classification chain, and it relies on the concept of a division mask to label the radar frames. With a statistical analysis of data, it is then possible to determine whether the frame belongs to Class 0, i.e., the class representing the absence of a human in the scene.

Upon receiving a radar frame, the incoming frame is divided by the clutter frame as follows

$$M_{div} = \frac{S}{F_{clutter}}, \tag{6}$$

where S is the incoming radar frame. If S is a sample taken in the Idle scenario, the result of this operation, M_{div}, is an array of elements with a value close to one, since both $F_{clutter}$ and S closely represent the same environment. On the other hand, if the frame does not correspond to that scenario, the result of this operation will be an array of elements whose values correspond to the relative difference between the acquired frame and the clutter frame. This comprises an effective way of detecting the Idle scenario, i.e., Class 0. However, it also means that the calibration phase must meticulously obtain frames that only correspond to the idle scenario in order for the division operation to make sense.

Owing to the fact that the calibration frame, $F_{clutter}$, is obtained during the system's operation, there is no offline operation required for the classification of the *Idle* Scenario, which also makes it possible for classification to happen in different environments or rooms since the clutter is gathered when the system is launched.

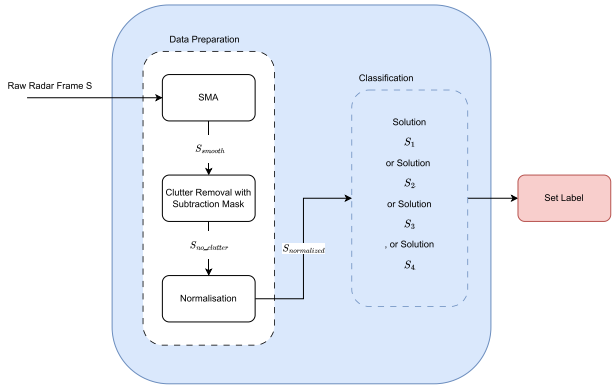


FIGURE 5. Diagram block of the key operations behind the posture detector.

FIGURE 6. Sitting and Standing Frames, before and after clutter removal. The clutter frame is presented in black.

Regarding the classification of the idle scene, we assume that noise typically follows a Gaussian distribution, where 95% of samples fall within two standard deviations of the mean value. In order to interpret the M_{div} mask obtained, two statistical metrics are adopted. The metrics are computed as follows

$$c_\mu = \mu_1\left[\mu_2\left(\frac{S^{(1)}_{calib}}{F_{clutter}}\right), \ldots, \mu_2\left(\frac{S^{(n_{calib})}_{calib}}{F_{clutter}}\right)\right], \quad (7)$$

$$c_\sigma = \mu_2\left[\mu_2\left(\frac{S^{(1)}_{calib}}{F_{clutter}}\right), \ldots, \mu_2\left(\frac{S^{(n_{calib})}_{calib}}{F_{clutter}}\right)\right], \quad (8)$$

where $S^{(k)}_{calib}$ is the k-th frame gathered during calibration, and $F_{clutter}$ is the clutter frame. μ_1 represents the statistical mean, and μ_2 is the second central moment. To label a frame as class 0, i.e., belonging to the idle scenario, we check if the frame to classify, S, meets the condition

$$c_\mu - 2c_\sigma \leq \mu_1(S) \leq c_\mu + 2c_\sigma.$$

If frame S does not meet this condition, the system will proceed to the posture detection stage to classify the posture in the frame as either *Standing* or *Sitting*, as described in Section IV.

IV. CLASSIFICATION METHODS

As depicted in Fig. 4, *posture detection* is the second step of the classification chain that labels radar frames as belonging to one of the two posture classes defined: *Sitting* (pictured in Fig. 1(b)) and *Standing* (pictured in Fig. 1(c)). Before this categorisation takes place, the raw frame received from the radar goes through three different operations: simple moving average, clutter removal, and normalisation described in (1) to 4. Fig. 5 illustrates these processes in a simple block diagram. After performing these operations, the resulting frame can be labelled as belonging to *Sitting* or *Standing* classes. This is mapped across the four different solutions: S_1, S_2, S_3, and S_4.

The considered approaches are developed for frame classification into *Sitting* and *Standing* classes. The first solution, S_1, utilises an algorithm to compare radar signals to saved masks for each posture, inspired by prior work, [12]. Solutions S_2 and S_3 employ DL for classification, differing only in the nature of their training datasets. Solution S_4 also employs DL, but instead of using raw data as the input, the DL model leverages synthetic features computed by a pre-trained VAE encoder.

A. SOLUTION S_1: MASK CORRELATOR

This solution is based on prior sampled masks that are used for correlation-based human activity recognition with RF sensors by capturing relevant spatial and temporal features in the signal data, allowing for more accurate correlation of patterns associated with different activities. More specificaly, solution S_1 utilises a subtraction mask to generate the posture masks to be compared with $S_{normalised}$. The aforementioned posture masks, which must be created offline, are obtained through the procedure introduced next.

The first step is to remove the clutter from each *Standing* and *Sitting* present in the previously recorded dataset $G1$. To gather the clutter associated with these, it is necessary to perform a similar operation as to the one depicted in (1), but this time for the *Idle* frames saved in dataset $G1$ as follows

$$F_{clutter} = \frac{1}{n_l} \cdot \sum_{n=1}^{n_l} S^{(n)}_{idle}, \quad (9)$$

where n_l is the number of frames for a given class l, and $S^{(n)}_{idle}$ is the n-th idle frame from the dataset.

The clutter removal relies on the subtraction of this clutter frame from each of the posture-related frames in the dataset. Consequently, a mask of each class after clutter removal is achieved by

$$M_{sub_l} = \frac{1}{n_l} \cdot \sum_{n=1}^{n_l} (S_{l,n} - F_{clutter}), \quad (10)$$

where $F_{clutter}$ is the clutter frame and n_l is the number of frames for a given class l. Thus, $S_{l,n}$ consists of the n-th frame of a class l.

Fig. 6 represents the clutter frame $F_{clutter}$ in black, along with the mean of *Standing* and *Sitting* Frames gathered from the dataset. It is also possible to see the result of the clutter removal for both cases. A final normalisation operation is

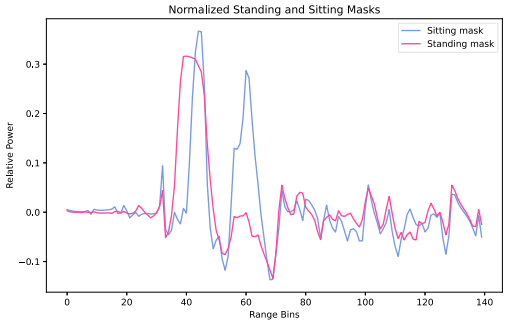


FIGURE 7. Normalised Sitting and Standing Masks, $M_{norm\_sub_1}$ and $M_{norm\_sub_2}$, in blue and pink, respectively.

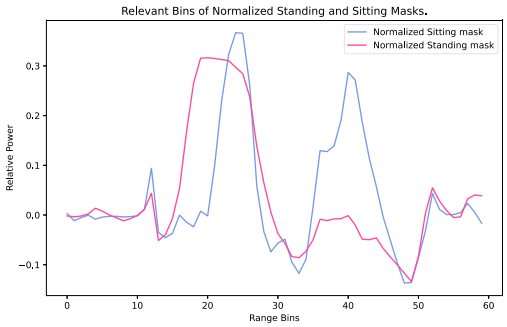


FIGURE 8. Final Masks obtained after selecting the most relevant range bins. $M_{posture\_1}$ represents the *Sitting* posture mask in blue, and $M_{posture\_1}$ the *Standing* posture mask in pink.

performed over the masks. This is done to counter the many bins with values close to or less than 0 after the subtraction operation in (10). The normalisation operation takes the form

$$M_{norm\_sub_l} = \frac{M_{sub_l}}{||M_{sub_l}||}, \tag{11}$$

where l is the class label and $||M_{sub_l}||$ represents the L2 norm of the M_{sub_l} array.

Fig. 7 illustrates the result of the normalisation process. As seen in these plots, essential information is simple to recognise, because bins containing important information have a greater value than the other bins, which round to 0. Thus, it is possible to handpick the bins that contain the most relevant information. Based on this, the masks were sliced from bins 20 to 80, resulting in the final masks depicted in Fig. 8. After this operation, the masks are finally ready to be stored and later loaded into the system to assist in classifying the postures in solution S_1, thus being denominated $M_{posture\_1}$ and $M_{posture\_2}$, corresponding to the *Sitting* and *Standing* scenarios, respectively.

The masks are fundamental in classification since they carry the specific data patterns. In order to label a frame with the appropriate class, it is necessary to check which posture mask is the most similar to that frame. This is achieved by calculating a similarity score between the frame and each posture mask using cross-correlation. The respective class of the posture mask with the highest similarity score can then be

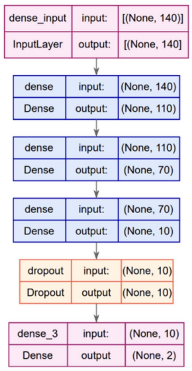


FIGURE 9. ANN architecture adopted in non-VAE DL-based solutions (S_2 and S_3).

assigned as the label for that frame. Cross-correlation computes an array $X_{l,n}$, expressed as

$$X_{l,n} = \sum_{m=0}^{N-n-1} S_{normalised}[m] \cdot M_{posture\_l}[m+n], \tag{12}$$

where l is the class label, $M_{posture\_l}$ is the posture mask associated with that class, and $S_{normalised}$ is the smoothed, clutter-free, normalised radar frame. The cross-correlation scores for each class are calculated and added to the $X_{l,n}$ array. Since there are two posture masks, corresponding to classes $l = 1$ and $l = 2$, $X_{l,n}$ consists of a two-dimensional array where $X_{1,n}$ and $X_{2,n}$ contain the correlation scores obtained for classes $l = 1$ and $l = 2$. To determine the likelihood of the frame belonging to Class 1 or Class 2 is obtained through

$$label = \text{argmax}_l(X_{l,n}), \tag{13}$$

where *argmax* finds the position of that score in the array, revealing the corresponding index l of the $X_{l,n}$ array, and thus also revealing the predicted class.

B. SOLUTIONS S_2 AND S_3: DEEP LEARNING WITH RAW DATA

Solutions S_2 and S_3 are very similar, as they both adopt the same ANN architecture based on DL rather than masks for classification. A simple ANN, whose topology is depicted in Fig. 9, receives a single real-time normalised frame ($S_{normalised}$), a result of the processes described in Section III-B.

In solution S_2, the ANN was trained with radar frames recorded at 1 m away from the radar, whereas solution S_3 uses multi-distance information for training (*Standing* and *Sitting* frames at distances of 0.7, 1, and 1.3 m). Another difference between the two solutions is the fact that S_3 does not use the K-fold cross-validation technique for training purposes.

The proposed ANN is made up of four successive fully connected (dense) layers. This architecture transforms the 140 input features (or range bins) into an output that represents the probability of a frame belonging to each of the two classes. The first three dense layers utilise the *ReLU* activation function, and all of them employ kernel regularizers that apply L1

regularisation penalties. The use of regularizers translates to inducing an error parameter so that the model converges to the optimal performance with minimal loss. That way, the model can learn representations that are simpler and more generic, allowing for the recognition of patterns in data, which combats overfitting. A 20% dropout rate between the output layer and the final hidden layer is adopted, aiding in the prevention of overfitting by minimising co-adaptation across neurons through some random input unit deactivations during training. The output layer utilises the softmax activation function, which outputs an array containing a probability value for each class. The model employs categorical cross-entropy as a loss function during training and modifies its weights to reduce that loss. The Adam optimiser is chosen for that purpose and is one of the most popular optimisers due to fast model convergence.

The process of constructing a neural network is divided into *training* and *offline testing*, which help optimise the model's accuracy and loss.

1) TRAINING

In real-time classification, the input of the ANN is a normalised frame that is the result of the *Data Preparation* block. Thus, the model should be trained to work with data that has undergone the same process as the real-time frames. This translates to applying the operations explained in Section III-B, to each radar frame that comprises the acquired dataset.

A total of 716 frames are available in a training dataset for each posture class. For S_3, a larger number is considered due to the fact that there are 956 frames per class and three distances being considered for each of them, resulting in 6 combinations (2 classes and 3 different distances). Thus, a total of 5736 frames can be used for training and testing of the ANN in solution S_3.

The second step, common to both S_2 and S_3, is to generate a dataset composed of labels and frames for both classes, which was achieved via simple coding. This final dataset is divided in the ratio 80:20 for training and testing, respectively. In order to prevent the model from picking up an unwanted pattern from the order in which the data was presented to it, the *standing* and *sitting* frames in this dataset are not grouped en masse but are rather randomly shuffled.

Finally, K-fold cross-validation is used in S_2 to assess the performance of the model. A 5-fold technique allows the training set to be divided into five subgroups, after being reshuffled. A distinct subset is chosen as the test set in each of the five iterations of the training process, with the remaining sets being referred to as training sets. In each iteration, the f1-score and recall metrics are saved. Upon completion of the process, the average of these metrics can be obtained. A total of 60 epochs per fold are performed, with a batch size of 32 samples.

The training phase differs from S_3 since it does not employ the K-fold methodology and utilises an increased batch size of 128 frames. The training procedure employs 1000 epochs instead of 60.

2) OFFLINE LEARNING AND TESTING

Three postures were subsequently selected for classification: *Sitting* (Class 0), *Standing* (Class 1), and *Idle* (Class 2). In each scenario, the room and radar positioning remain constant. The first scenario, "Idle," represents the experimental setup with no subject in front of the radar. The second scenario involves a person seated in front of the radar, while the third scenario features a standing subject in front of the radar with the chair positioned behind them. The process of saving information from the radar is denoted as *dataset acquisition* and is important in two stages of practical work. The first precedes classification, where radar frames for each scenario are recorded in separate files, comprising the first group of frames or first saved. The second stage is when calculating the accuracy provided by the solutions proposed in order to assess classification outcomes, where it is important to save each received frame, the label assigned to it, and the computation time required to set the label for that frame. This data is then utilised in training the DL models. For S_2, and S_4, the learning data was collected at a distance of 1 m, while for S_3, the learning data was collected at distances of 0.7 m, 1.0 m, and 1.3 m.

After completing the training process, it is possible to test the model on new radar frames. This can be done in two phases; the first is acquiring another small set of frames, which qualifies as unseen data, and testing the model on this data. This is called *offline evaluation*. For S_2, a total of 1912 frames were collected, yielding an accuracy value of 98.58% and a loss value of 18. This data was gathered for the *Sitting* and *Standing* scenarios at a distance of 1 m because it is expected for this ANN to classify posture with adequate accuracy for this distance, given that it was trained with data collected at the same distance of 1 m. For S_3, 736 were used to assess how the model behaved, and it was possible to gather the accuracy value of 100%, with a loss value of 6.56.

The next phase consisted of loading the model into the classification system and testing its performance in real-time. The performance results from this phase are reported in Section V.

C. SOLUTION S_4: VAE-BASED FEATURE GENERATION

This section presents the classification solutions that include the derivation of classification features with a VAE. VAEs were adopted for feature generation in real-time applications because they generate flexible latent representations without relying on the fixed statistical details of prior data, unlike methods such as principal component analysis, making them more suitable for dynamic environments where data statistics can shift over time.

1) FEATURE GENERATION

In solutions S_1 to S_3, raw data is used to classify the postures. However, the quality of input data can be improved for machine learning classification if we can identify features generated from the raw data that better represent the underlying classes. In solution S_4, a VAE is trained to derive those features from raw data. The features generated from raw

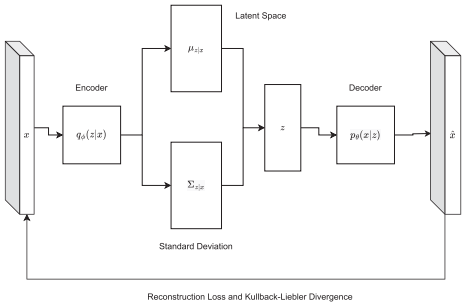


FIGURE 10. VAE architecture.

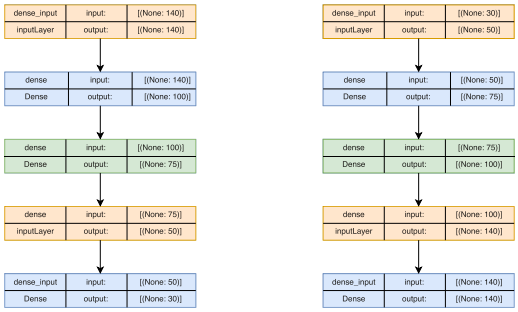


FIGURE 11. VAE *encoder* and *decoder* neural network structure.

data are obtained from the latent space of a variational auto-encoder, which represent synthetic features generated from raw data that are further adopted in the classification process.

The methodology followed to extract classification features from raw data, whose conceptual architecture is depicted in Fig. 10, takes as input the raw data vector of a S_{norm} sample, as adopted in solutions S_2 and S_3. Features are extracted by the VAE's encoder, which is essentially one of the neural networks composing the VAE. At the end of the encoding process, the encoded data is of a lower dimension than the input. The VAE also includes a decoder, which is a second neural network that uses the encoded data to approximate the original raw data. The training of a VAE involves looking at the reconstructed decoder output to assess the reconstruction loss relative to the initial input.

In this work, the VAE encoder comprises a simple neural network with three *Dense* hidden layers illustrated in Fig. 11. The input layer receives a 140-bin vector and represents it to a latent dimension of 30.[1] Consequently, the 30 latent variables represent the features extracted from the data represented with 140 bins, representing a dimensionality reduction of $140/30 = 4.67$ times. The decoder is composed of three hidden layers and a subsequent output layer. The VAE encoding loss was monitored during the training stage through the Kullback-Liebler divergence. A *relu* activation function was implemented for all layers of the encoder (depicted to the left in Fig. 11) as well as all layers of the decoder (depicted to the right in Fig. 11), except the last (outer reconstruction layer), where a *sigmoid* activation function was implemented. Note

[1]This value achieved an acceptable trade-off between the VAE's accuracy and the number of latent variables.

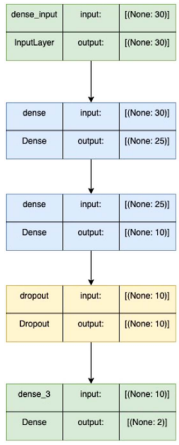


FIGURE 12. ANN architecture adopted in solution S_4.

that the encoder ANN structure in Fig. 11 is a mirror image of the decoder. The dataset used for training and testing consisted of 5736 records split into 5000 records for training and 736 for testing. This dataset was collected at a distance of 1 m for all postures. For the training, we also considered 1000 epochs and a batch size of 100. The VAE model was trained considering a Mean Square Error loss.

2) SOLUTION S_4: VAE WITH RAW DATA

The solution S_4 is based on the extraction of the 30 classification features through the VAE encoder. The entire dataset of frames collected at a distance of 1 m, with 5736 frames, is encoded using the VAE model. The sample-wise computation time for VAE encoding, denoted as λ, is taken into account for computation time comparison.

The classification is then performed by an ANN topology depicted in Fig. 12. When compared to the ANN used in S_2 the number of inputs is reduced from 140 to 30 and, consequently, the number of dense hidden layers is also reduced, as is its dimension. The activation functions as well as the training optimizer and methodology are the same as those adopted in S_2.

The computation time of the classification scheme is hence denoted as $\delta_{pureVAE}$ and is defined as

$$\beta_4 = \lambda + \delta_{pureVAE}, \tag{14}$$

where λ represents the average computation time of the VAE encoder to compute the 30 features and $\delta_{pureVAE}$ denotes the classification computation time, i.e., the time taken by the ANN to perform labelling when fed with VAE-encoded data.

V. PERFORMANCE EVALUATION

The proposed solutions were designed with different objectives in mind, balancing computational efficiency and classification performance. Solution S_1 was developed as a simple approach with low computational requirements, relying solely on a basic correlation between predefined masks and the gathered data. While this method ensures fast processing, it does not necessarily achieve the highest classification

TABLE 3. Table of Accuracy Metrics for non-VAE Solutions (S_1, S_2, and S_3)

	S_1			S_2			S_3		
Scenario Distance	**Idle**	**Sitting**	**Standing**	**Idle**	**Sitting**	**Standing**	**Idle**	**Sitting**	**Standing**
0.7 m	85.4%	94.6%	99.2%	91.3%	0.4%	95.8%	97.1%	96.7%	100.0%
1.0 m	93.3%	96.3%	100.0%	96.3%	96.7%	99.6%	95.0%	97.9%	100.0%
1.3 m	95.8%	45.8%	100.0%	96.7%	87.1%	39.6%	96.3%	92.9%	100.0%

performance. To address this limitation, solution S_2 introduces a basic DL model, leveraging learning techniques to improve the classification of three classes when the target remains at a fixed distance from the FMCW device. Building upon this, solution S_3 extends the learning stage of the DL model by incorporating multiple target distances, allowing for a broader evaluation of how classification performance adapts when targets appear at varying distances. Unlike solutions S_2 and S_3, which rely solely on raw data, solution S_4 extracts multiple features from the raw data to enhance data representation before classification. This feature generation step is designed to simplify the classification task, enabling the use of a less complex DL model compared to those employed in solutions S_2 and S_3, while still maintaining strong classification performance.

Two critical metrics were considered in the performance evaluation of the four solutions: classification accuracy and computation time. The computation time comprises explicitly the system's frame-labeling time, while accuracy refers to the correctness of the assigned label. Regarding the former metric, while various metrics can be used to assess computation time, our study specifically focused on frame-marking time, as it directly impacts the real-time performance of DL-based methods. Although training time is another potential metric, we chose not to include it for two main reasons: the first being that the training data differs between Solution S_2 and Solution S_3, with S_2 relying on data collected at 1 m and S_3 incorporating multiple distances (0.7 m, 1.0 m, and 1.3 m), making direct comparisons of training times impractical; and the second reason being that in real-world deployment, models are typically pre-trained, making training time less critical for real-time performance evaluation. The two metrics are measured across all classes and solutions and since classification is intended to work for multiple distances, it makes sense to test performance at different ranges as well.

Although accuracy and calculation time are important measures of algorithm performance, they do not necessarily provide a comprehensive assessment. Particularly in unbalanced datasets, additional pertinent measures like precision, recall, and F1-score can provide more in-depth understanding of the ratio of false positives to false negatives. However, as our main goal in this study was to examine the trade-off between efficiency and performance for the specified application, we prioritized computation time and accuracy.

As illustrated in Fig. 4, the classification process encompasses both the Idle Scene Detector and the posture detector. Because of this, the computation time required to label a frame is measured since a frame enters the Idle Scene Detector and only ends when a label is set.

Tables 3 and 4 present both the accuracy and average computation time obtained for each of the 27 combinations of solution, distance, and class (for S_1, S_2, and S_3). These results are obtained from each set of 240 frames analysed from dataset G_2.

We highlight that the optimum solution would have an accuracy of more than 90% across all classes and distances. However, it should also have a "suitable" computation time, long enough to appropriately label the frames but not too long to register a delay exceeding the system's 250 ms frame-reception rate. Table 5 presents a comparison of the average performance metrics for the non-VAE solutions metrics across all considered distances.

A. DISCUSSION OF RESULTS

Based on the results in Table 4, we see that S_1 achieves the shortest computation time. Furthermore, based on Table 3, the average accuracy for different distances and multiple scenarios is 90%, which provides a good starting point considering the short computation time achieved with this method. The faster computation time can be attributed to the method's reliance on correlation and division operations, which are mathematically and computationally simple. When examining the development of accuracy with increasing distance, we observe that the average accuracy across the three classes peaks at 1 m. This is because the posture masks developed are computed using frames collected at 1 m. The silver lining is that these masks can still classify frames at various distances with high accuracy, like in the instance of 0.7 m or at 1 m for the *Standing* pose. However, the pattern in the *Seated* mask may alter more dramatically as the distance increases, as the accuracy lowers to 45.8% at 1.3 m. This may be demonstrated by obtaining posture masks for distances of 0.7 and 1.3 m using the identical procedures discussed in Section IV-A, which detailed how the 1-metre posture masks were generated and utilised in this work. These masks are presented in Figs. 13 and 14.

Solution S_2 is based on an ANN trained with a dataset gathered at a distance of 1 m, containing a total of 1430 frames. Analysis of the subsequent accuracy results (from S_2) in Table 3, ranging from 0.4% to 99.6%, depicts less consistency compared to the results derived for S_1 and S_3. As

TABLE 4. Table of Computational Time Metrics for non-VAE Solutions (S_1, S_2, and S_3,), in ms

	S_1			S_2			S_3		
Scenario / Distance	Idle	Sitting	Standing	Idle	Sitting	Standing	Idle	Sitting	Standing
0.7 m	0.74	0.81	0.89	12.27	143.85	150.47	4.69	150.03	138.50
1.0 m	0.76	0.74	0.74	6.06	147.31	153.99	7.93	152.94	154.54
1.3 m	0.73	0.80	0.86	6.72	140.34	132.36	5.53	144.35	152.55

TABLE 5. Table of Total Average Accuracy and Computational Time (in ms) Metrics for non-VAE Solutions (S_1, S_2, and S_3), Across All Distances

	S_1	S_2	S_3
Accuracy (%)	90%	78%	97%
Computation Time (ms)	0.79	99.26	101.23

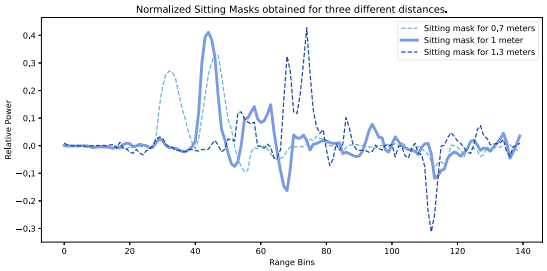


FIGURE 13. *Sitting* posture masks obtained for different distances.

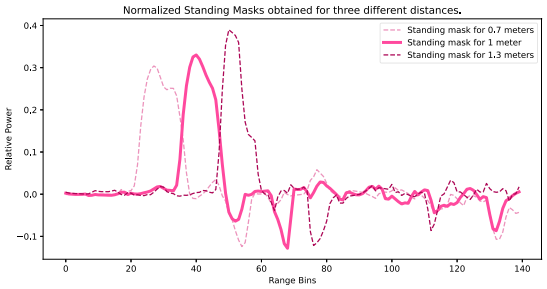


FIGURE 14. *Standing* posture masks obtained for different distances.

is expected, S_2 performs best at a distance of 1 m, with the *Sitting* pose at a distance of 0.7 m being almost impossible to classify, and the accuracy at a distance of 1.3 m is dispersed between two low values of 87.1% and 39.61% for the *Sitting* and *Standing* poses, respectively. The substantial increase in computation time did produce a corresponding improvement in accuracy. Consequently, the trade-off between accuracy and computation time is deemed unworthy, particularly due to compromised robustness.

Of all non-VAE solutions, S_3 achieves the highest average accuracy of 97.3%. Furthermore, classification accuracy is consistent across classes and distances, meaning accuracy values range from about 93% to 100%. This could be attributed to the robustness of the model and its consequent ability to generalise well, resulting from the use of a more diverse and larger dataset than the one used for S_2. In addition to considering several distances, each scenario and distance combination are recorded for a longer period of time, producing a greater

TABLE 6. Comparison of Total Average Classification Accuracy and Computation Time (in ms) Across All Solutions at a Distance of 1.0 m

	S_1	S_2	S_3	S_4
Accuracy (%)	96.53%	97.53%	97.63%	99.32%
Computation Time (ms)	0.7466	102.45	105.13	27.25

number of frames. Similarly to S_2, the average time consumed to attribute labels is higher than the one devised in solution S_1. However, considering that the accuracy is typically better and constant across the respective classes and distances, the trade-off between computation time and accuracy compensates.

Solution S_4, implementing VAE-based feature generation with the VAE trained and tested with raw data collected at 1 m, proves to have a slight edge over the non-VAE solutions in terms of classification accuracy, with a yield of 99.32%. This is shown in Table 6, where the results of all solutions at a distance of 1 m are compared. In terms of the computation time, S_4 remains well within the set 250 ms delay threshold, performing at 27.25 ms per sample with 30 latent variables. Solution S_4 implemented a VAE model for feature extraction before classification using an ANN. As explained before, the VAE model was trained and used to encode data collected at a distance of 1 m. The classification was also done for postures at a distance of 1 m.

B. COMPARISON OF PROPOSED SOLUTIONS

Taking into account that the radar system sampling period is 250 ms, the classification computation time must take less than 250 ms to categorise each received frame appropriately. Despite the fact that solutions S_2 and S_3 are slower than S_1, they still would not jeopardise the classification time window since they do not result in a delay that violates this preset condition. Thus, it is reasonable to claim that S_1, S_2 and S_3 are within the permissible boundaries of operation in terms of computation time.

Given that all solutions employ the singularly developed Idle Scene detector, it may appear strange that their classification computation times for Class 0 differ. This is due to the efficiency of the classification module. As shown in Table 5, the Idle Scene detector has an average accuracy of around 94%, indicating that certain frames, albeit of negligent quantity, will be labelled as *Standing* or *Sitting*, despite the actual class being "Idle". Because S_2 and S_3 are slower than S_1, classification will be slower when frames are fed to the posture detector while employing any of the former two solutions. As

a result, the average computation time achieved for solutions S_2 or S_3 is more than that obtained for S_1, despite the fact that the Idle Scene detector utilised is the same.

S_1 and S_2 consider the same distance (1 m) during the pre-classification phase: S_1 creates the posture masks using 1 m-frames, and S_2 uses a dataset gathered solely at 1 m to train the ANN developed. When comparing the accuracy values for the *Standing* and *Sitting* positions for this distance, S_1's average accuracy totals 98.15%, while S_2 's accuracy value is precisely the same. This implies that at a distance of 1 m, S_1 is more viable, not because it yields a higher accuracy, but because it is faster. However, bringing S_3, which was trained for multiple distances, into the picture, the average accuracy for 1 m is slightly higher at 98.95%. This brings us to the conclusion that at a distance of 1 m, if the goal is to achieve the highest accuracy, S_3 would be the most suitable.

Overall, if the goal is to perform classification at multiple distances, the optimum non-VAE solution considered is S_3 due to the highest accuracy obtained through all classes and ranges.

Considering the VAE-based solutions S_4 which implements an ANN with purely VAE-encoded data collected at a distance of 1 m, yields a classification accuracy of 99.32%. Comparing this performance with the values obtained for classification at a distance of 1 m with S_1, S_2, and S_3, S_4 performs better due to the finer detail of the 30 latent variables generated from the raw data.

In terms comparison of computational time between S_4 and the rest, only S_1 which implements mask correlation for classification, exhibits a lower computation time. This can be attributed to the computationally lightweight nature of S_1 that only depends on mathematical operations and does not implement an ANN.

VI. CONCLUSION

In this work, a human posture recognition system has been developed using RF sensing technology with an FMCW radar device. A prototype was built to classify radar signals into *Idle*, *Sitting*, and *Standing* scenarios. To this effect, four solutions were proposed to classify the two postures: S_1 inspired by prior work, S_2 and S_3 implementing ANNs with different training approaches and tested for multiple distances, and S_4 implementing VAE-based feature generation and ANNs trained and tested with data at a single distance. The solutions' performance were consequently compared on the basis of classification accuracy and sample-wise computation time, with S_4 implementing a VAE trained and tested with data at a distance of 1 m yielding the highest average classification accuracy at 99.32% and S_1 yielding the lowest average sample-wise computation time at 0.79 ms. Across all solutions, the models performed best at a distance of 1 m, with S_3 trained for multiple distances naturally yielding the most robust results. Furthermore, findings from S_3 suggest that the use of generative AI techniques to synthesize data for a broader range of distances can potentially enhance classification performance, which is a promising direction for future research.

REFERENCES

[1] S. -J. Ryu, J. -S. Suh, S. -H. Baek, S. Hong, and J. -H. Kim, "Feature-based hand gesture recognition using an FMCW radar and its temporal feature analysis," *IEEE Sensors J.*, vol. 18, no. 18, pp. 7593–7602, Sep. 2018.

[2] Y. Sun, T. Fei, F. Schliep, and N. Pohl, "Gesture classification with handcrafted micro-doppler features using a FMCW radar," in *Proc. 2018 IEEE MTT-S Int. Conf. Microw. Intell. Mobility*, 2018, pp. 1–4.

[3] P. Goswami, S. Rao, S. Bharadwaj, and A. Nguyen, "Real-time multi-gesture recognition using 77 Ghz FMCW MIMO single chip radar," in *Proc. 2019 IEEE Int. Conf. Consum. Electron.*, 2019, pp. 1–4.

[4] C. Ding et al., "Continuous human motion recognition with a dynamic range-doppler trajectory method based on FMCW radar," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6821–6831, Sep. 2019.

[5] A. Shrestha, H. Li, J. L. Kernec, and F. Fioranelli, "Continuous human activity classification from FMCW radar with bi-LSTM networks," *IEEE Sensors J.*, vol. 20, no. 22, pp. 13607–13619, Nov. 2020.

[6] J. Bhatia et al., "Object classification technique for mmwave fmcw radars using range-FFT features," in *Proc. 2021 Int. Conf. Commun. Syst. Networks*, 2021, pp. 111–115.

[7] K. Han and S. Hong, "Phase-extraction method with multiple frequencies of FMCW radar for human body motion tracking," *IEEE Microw. Wireless Compon. Lett.*, vol. 30, no. 9, pp. 927–930, Sep. 2020.

[8] R. Cruz, A. Furtado, and R. Oliveira, "Assessment of feature selection for context awareness rf sensing systems," in *Proc. IEEE 95th Veh. Technol. Conf.: (VTC2022-Spring)*, 2022, pp. 1–6.

[9] S. A. Shah and F. Fioranelli, "Human activity recognition: Preliminary results for dataset portability using FMCW radar," in *Proc. 2019 Int. Radar Conf.*, 2019, pp. 1–4.

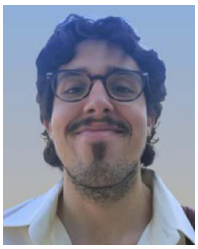
[10] R. Yao, C. Liu, L. Zhang, and P. Peng, "Unsupervised anomaly detection using variational auto-encoder based feature extraction," in *Proc. 2019 IEEE Int. Conf. Prognostics Health Manage.*, 2019, pp. 1–7.

[11] Y. Huang, C. -H. Chen, and C. -J. Huang, "Motor fault detection and feature extraction using RNN-based variational autoencoder," *IEEE Access*, vol. 7, pp. 139086–139096, 2019.

[12] H. P. da C. Fernandes, "Context-awareness through active RF sensing," Master's thesis, Universidade Nova de Lisboa, Lisbon, Portugal, Nov. 2022.

EUGENE CASMIN (Graduate Student Member, IEEE) received the Bachelor of Science and Master of Science degrees in computer engineering from Middle East Technical University, Northern Cyprus Campus (ÖDTU KKK), Ankara, Türkiye. He is currently working toward the Ph.D. degree in electrical and computer engineering with the Nova School of Science and Technology (NOVA FCT), Lisbon, Portugal. He is a Research Fellow with the Instituto de Telecomunicações (IT).

MIRIAM RODRIGUES received the M.Sc. degree in electrical and computers engineering from Universidade Nova de Lisboa, Lisbon, Portugal, in 2023, specializing in telecommunications, and robotics and manufacturing, where she focused on developing context-awareness systems using radio frequency signals in her M.Sc. dissertation. She was a Software Developer with Digital Payment Solutions Sector. Her research interests include advancing the understanding of real-time radar data behavior through real-time visualization and processing techniques and the development of a machine learning-based posture classification prototype.



AMÉRICO ALVES is currently working toward the M.Sc. degree in electrical and computers engineering with the NOVA School of Science and Technology, Universidade Nova de Lisboa, Lisbon, Portugal, specializing in telecommunications, control and decision and digital systems. From 2020 to 2021, he was a Researcher with SC – Spike Computing Project, Centre of Technology and Systems, UNINOVA, focusing on quantum computing. His current research focuses on designing and implementing innovative algorithms for human activity recognition using active radar signals, based on signal processing and deep learning techniques.

RODOLFO OLIVEIRA (Senior Member, IEEE) received the Licenciatura degree in electrical engineering from the Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa (UNL), Lisbon, Portugal, in 2000, the M.Sc. degree in electrical and computer engineering from the Instituto Superior Técnico, Technical University of Lisbon, Lisbon, in 2003, and the Ph.D. degree in electrical engineering from UNL, in 2009. From 2007 to 2008, he was a Visiting Researcher with the University of Thessaly, Volos, Greece. From 2011 to 2012 and in 2023, he was a Visiting Scholar with Carnegie Mellon University, Pittsburgh, PA, USA. He is currently with the Department of Electrical and Computer Engineering, UNL. He is also a Senior Researcher with the Instituto de Telecomunicações, where he researches wireless communications, computer networks, and computer science. He is on the Editorial Board of Ad Hoc Networks (Elsevier), ITU Journal on Future and Evolving Technologies (ITU J-FET), IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY, and IEEE COMMUNICATIONS LETTERS.