# Feature Extraction using Variational Auto-encoder for Radar-based Posture Detection Systems

Eugene Casmin[†*], Rodolfo Oliveira[†*],
[†]Departamento de Engenharia Electrotécnica, Faculdade de Ciências e Tecnologia, FCT,
Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
[*]IT, Instituto de Telecomunicações, Portugal

2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring) | 979-8-3503-8741-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/VTC2024-SPRING62846.2024.10683040

*Abstract*—**This paper explores the impact of different classification and feature engineering techniques in radar-based posture detection systems. The research focuses on an experimental setup involving a radar sensor used to classify human postures and the absence of human activity. The methodology includes the detection of the absence of human activity in the first stage, followed by feature engineering extraction through Variational Auto-Encoders (VAEs), and subsequent data reduction via either t-distributed Stochastic Neighbor Embedding (t-SNE) or Principal Component Analysis (PCA) if human activity is detected and posture classification is required. Five solutions differing in terms of methodology and features' dimensionality reduction strategies are presented and compared in terms of classification accuracy and computation time. The main findings reveal that all five solutions achieve classification accuracy exceeding $90\%$, with variations observed in computation times, showing the effectiveness of adopting VAEs to derive classification features from raw data. The research highlights a trade-off between data preservation and classification accuracy and/or computation time, demonstrating that feature extraction and data reduction techniques are a valuable enhancement to posture detection and classification.**

*Index Terms*—**Posture Classification, Radar Sensing, VAEs, t-SNE, PCA, Performance Evaluation.**

## I. Introduction

The detection and classification of human posture have become a prominent focus of research trends in recent times. Classification of human postures has proven to be an essential component in many technological advancements, to improve human-computer interaction and/or stimulate innovation of human-centric systems. Practical applications of posture classification in quotidian life heavily include sensitive sectors such as occupancy detection for home security, pedestrian detection in traffic, and palliative care. However, with the exponential growth in technologies and application areas for posture classification, there are still several emerging issues that remain persistent. The dynamic nature of human motion gives rise to one of the fundamental problems. In contrast to immobile objects, humans move in a variety of ways, which makes it difficult to precisely capture and classify different postures. Variability is introduced by the dynamic nature of the problem and includes shifts in body positions, posture transitions, and motion speed variations [1], [2]. This variability presents a substantial challenge to achieving robust posture classification, because models must consider the aforementioned dynamic factors while taking temporal and spatial dimensions into account.

In addition, difficulties stem from the context dependence of sensing data, which affects how reliable posture classification algorithms are [3]. The effectiveness of posture classification systems is frequently highly dependent on the state of the environment, differences in sensor modalities, and the dynamic character of human movement [2]–[4]. Variability in the data is caused by factors such as lighting, obstructions, and sensor modality selection, which weaken the robustness of classification algorithms. For example, a posture classification model trained in one setting might not function as well in another, which would reduce the model's re-usability. In real-world applications, where diverse and dynamic contexts are common, this presents a fundamental issue. It is critical to address the problems brought about by context-dependent sensing data to guarantee that solutions work in a variety of situations and to help create more adaptable and durable human-centric systems.

It is worth mentioning that there are issues with data integrity brought about by the selection of sensing modalities. In various settings, different sensors record information about human posture with differing degrees of accuracy and dependability. Trade-offs must be made when choosing a sensing modality, taking into account aspects like cost, deployment simplicity, and environmental robustness [5]. Every modality has its own set of difficulties. For example, vision-based systems may not function well in low light, and inertial sensors may be subject to drift in their readings. The objective is to choose the modality that not only maximises the application environment but also maintains cost-effectiveness. Designing efficient and dependable posture classification systems requires an understanding of the trade-offs related to each sensing modality in relation to the application environment.

Another layer of complexity is related to the trade-offs related to the computational size of the solution. Particularly in real-time applications or systems with constrained processing resources, computational efficiency is critical. Navigating the temporal and spatial dimensions is necessary to achieve high classification accuracy while minimizing computational requirements [6]. The solution needs to handle data efficiently over time to guarantee low latency and responsiveness. The computational load can be dependent on the size of the dataset and the intricacy of the spatial features. Making concessions to match the deployment environment's capabilities with the computational requirements is often vital to arrive at the best

979-8-3503-8741-4/24/$31.00 ©2024 IEEE

possible solution.

The paradox of choice, which arises from the abundance of features available for characterizing postures, is another crucial factor in feature engineering for human posture data. Effective static and motion-wise posture classification depends on the selection of pertinent features, but the sheer volume of options can create a paradox whereby the more features taken into account, the higher the chance of over-fitting or noise introduction into the model. In posture classification research, there is always a challenge to strike a balance between the requirement for simplicity and interpretability and the richness of feature representation. In response to this, Principal Component Analysis is one popular method that has been implemented to reduce dimensionality in raw data. In [7], Principal Component analysis (PCA) was implemented as a discriminator between normal and abnormal gait by analyzing ground reaction force. A PCA-based classifier was employed to accompany ANNs in [8] to classify focal lesions in ultrasound liver images. Both these instances proved that PCA can be instrumental in condensing raw data pre-classification, particularly in cases where there exists a one-or-other phenomenon. Additionally, in [4], multiple features for radar-based posture detection were evaluated, and PCA was adopted for dimensionality reduction, showing that a smaller set of features combined with the PCA technique can obtain similar performance to the case when an extended set of raw features is considered.

On the point of context-dependency, one of the causes of bias arises from the use of vision-based sensor modalities i.e., a classification model trained with data collected in a well-lit environment may fail to perform as expected when applied in real-time in the same environment but under less than ideal lighting conditions. The framework presented in this study utilizes radar sensing, which is more robust to environmental conditions such as lighting, fog, and other obstacles that would be categorized as hostile for other sensor modalities [9]. This paper seeks to decrease the computational bottleneck caused by the overwhelming nature of raw data generated by a radar system for posture detection purposes. Data-related computational bottlenecks can especially impede real-time posture classification, which is one of the targets of seamless posture classification.

The contributions of this paper are as follows:

1) A radar-based static posture learning and classification framework is described to classify different postures, including the resulting Datasets acquired from the radar sensor.
2) A methodology for feature engineering based on Variational Auto-Encoders (VAEs) is proposed.
3) A performance comparison between three solutions based on VAE implementation (further denoted as $S_3$, $S_4$, and $S_5$) and non-VAE solutions ($S_1$ and $S_2$) based on raw data is performed to identify the gain of the VAE as a method to derive features from the raw data.
4) The performance of different dimensionality reduction techniques is investigated to characterize the classification accuracy of VAE. Additionally, the trade-off between the computation time and classification accuracy is compared for different classification techniques and different levels of dimensionality reduction, showing that the maximum accuracy value is obtained with VAE and its higher computational time can be reduced without significant reduction of accuracy by adopting the PCA dimensionality reduction method.

The rest of this paper is organized as follows: Section II introduces the system model and solutions that do not employ feature extraction. Section III presents the solutions based on feature extraction and dimensionality reduction. The assessment of the framework's performance and the final remarks are presented in Section IV and Section V, respectively.

## II. System Model

The logical flow of the experimental setup since the data is sampled from the radar until the end of the posture classification process is summarized in the flowchart in Figure 1. The overall process can be separated into two distinct steps. The first step relies on the detection of an Idle Scene that represents the case when no human is located in front of the radar. The second step is related to the classification of data according to the output of the classification process through the different solutions ($S_1, S_2, S_3, S_4$, and $S_5$) proposed in this paper. Regarding the data flow, the raw data is sampled from the radar and filtered before the initial Idle Scene classification. If a human is detected in front of the radar, the classification is then performed via one of the five proposed solutions.

### A. Experimental Setup

We use a Frequency Modulation Continuous Wave (FMCW) radar that transmits chirps from three transmitting antennas, generating periodic chirp sequences (Doppler Chirps). Sixteen consecutive Doppler Chirps constitute a radar frame. The sampling period of the frames is $250$ ms. The technical specifications of the radar operation are shown in Table I.

Raw data matrices are compiled from I/Q samples in each RX channel, and a column-wise FFT is applied to create Range Bins (RB). A subsequent Fast-Fourier Transform (FFT) on

| **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 / $\mu s$ |
| Chirp Duration, $T_C$ | 66.67 $\mu s$ |
| ADC Sample Rate | 2.727 MHz |
| Range resolution | 0.043 m |
| Maximum Unambiguous Range | 6m |
| Maximum Radial Velocity | 1m/s |
| Velocity Resolution | 0.13 m/s |

TABLE I
Radar parameters employed in the practical work.

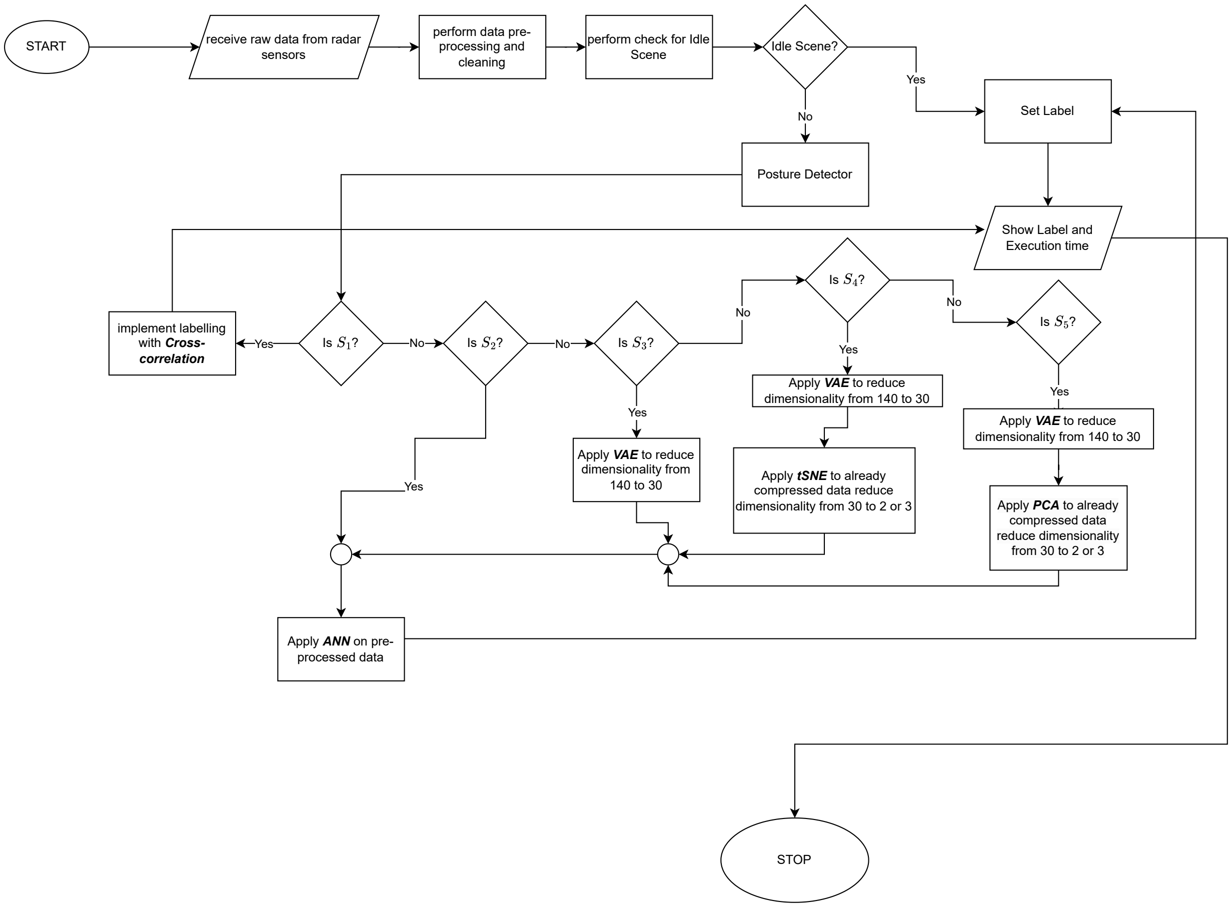
START
receive raw data from radar sensors
perform data pre-processing and cleaning
perform check for Idle Scene
Idle Scene?
Yes
No
Set Label
Posture Detector
Show Label and Execution time
implement labelling with *Cross-correlation*
Yes
Is $S_1$?
No
Is $S_2$?
No
Is $S_3$?
No
Is $S_4$?
No
Is $S_5$?
Yes
Yes
Yes
Yes
Apply *VAE* to reduce dimensionality from 140 to 30
Apply *VAE* to reduce dimensionality from 140 to 30
Apply *VAE* to reduce dimensionality from 140 to 30
Apply *tSNE* to already compressed data reduce dimensionality from 30 to 2 or 3
Apply *PCA* to already compressed data reduce dimensionality from 30 to 2 or 3
Apply *ANN* on pre-processed data
STOP

Fig. 1. System model flowchart.

each row yields the Range-Doppler Map (RDM), enabling radial velocity calculations. The RDM comprises $16$ columns, each designated as a Doppler Bin (DB), representing specific Doppler shifts. The study, conducted in a $4.02$ m by $9.06$ m room, uses the first $140$ range bins, covering approximately $6.02$ m. This approach ensures comprehensive posture data collection, classification, and visualization within the specified room dimensions. The radar is positioned as illustrated in Figure 2 to collect data on three different scenarios: Sitting (Class 0) and Standing (Class 1), along with an Idle Scene (Class 2). Figure 2 shows the two human postures and the idle scenario. It is vital to note that transition postures are not taken into consideration in this experiment. All the posture samples recorded are for definite postures, either fully seated or fully standing, or fully idle.

### *B. Datasets*

The privately available data used in this study was gathered using the radar introduced in Section II-A. The collected dataset is diverse and representative, ensuring the robustness and generalizability of the posture classification model. This framework lays the foundation for a comprehensive understanding of human posture dynamics, with the experimental design emphasizing the system's ability to adapt to different contexts, distances and environmental variations.

The dataset used throughout this study comprised $5736$ samples. Each sample consists of 140 FFT bins representing the RBs. For deep learning purposes, the dataset is split in the following format:

- Training data - comprises the first $5000$ samples. The training data is further divided in the ratio 80:20 into training and validation sets.
- Test data - includes the remaining 736 samples for testing results, presented in Section IV.

### *C. Masks*

In the context of classification via correlation, a mask refers to a pattern previously defined to represent a specific class, which is further correlated with the sampled data to measure its similarity and classify the sampled data according to the class for each the correlation achieves the highest value. The masks are acquired in advance of the classification process, serving as efficient tools for rapid posture detection and classification. This could be seen as a preamble for Solution $S_1$, which implements a cross-correlation scheme between the masks and the real-time data. Classification is achieved by comparing the similarity scores resulting from matching a frame with the

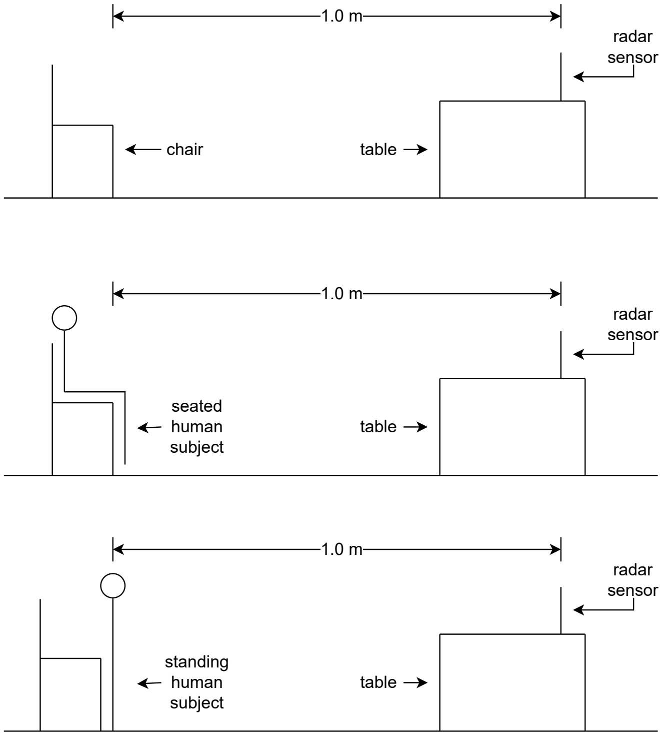

Fig. 2. Human postures and scenarios considered in the experiment.

Sitting and Standing posture masks. The masks defined in this work represent the postures *Sitting* (Class 0) and *Standing* (Class 1). The definition of the masks for a specific class encompasses the following steps:

1) Removal of clutter from the frames by subtracting an aggregation of all the Idle Scenes in the dataset. In theory, the goal is to get rid of background or non-posture information while still maintaining the integrity of the setup. The definition of a mask for human posture class $l$, denoted as $M_{sub_l}$, is as follows

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

where $F_{clutter} = \frac{1}{n_l} \cdot \sum_{n=1}^{n_l} S_n$ is the clutter frame that represents the average of $n_l$ consecutively sampled frames, each one denoted by $S_n$, sampled when no human is located in front of the radar. $S_{l,n}$ represents a frame sampled when a human is in front of the radar and performing a posture of class $l$.

2) The masks are normalized as follows

$$M_{norm\_l} = \frac{M_{sub_l}}{||M_{sub_l}||}, \tag{2}$$

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

$$||V|| = \sqrt{\sum_{i=1}^{n} V_i^2}, \tag{3}$$

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

The resulting masks contain specific data patterns that are crucial for one-to-one frame filtering for labeling. To detect the case where no humans are located in front of the radar, denote as Idle Scene detection (Class 2), a sampled frame $S$ is processed as follows

$$M_{div} = \frac{S}{F_{clutter}}. \tag{4}$$

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. Due to noise in both $S$ and the averaged frame $F_{clutter}$, Class 2 is detected if the following statistical condition holds true

$$M_{div} - 2\sigma \leq M_{div} \leq M_{div} + 2\sigma,$$

where $\sigma$ represents the standard deviation of the $n_l$ frames used to compute $F_{clutter}$. Next, we present different solutions to classify the human posture classes, i.e., Class 0 and Class 1.

### *D. Solution $S_1$ - Mask Correlator*

Solution $S_1$ implements the masks defined in Section II-C using **cross-correlation** to classify the postures in each frame. The cross-correlation computes an array $X_{l,n}$, expressed as

$$X_{l,n} = \sum_{m=0}^{N-n-1} S_{norm_m} \cdot M_{norm\_l_{(m+n)}}, \tag{5}$$

where $l$ is the class label, $M_{norm\_l}$ is the posture mask associated with that class, and $S_{norm} = \frac{S - F_{clutter}}{||S - F_{clutter}||}$ is normalized frame of the sampled frame $S$ to classify. $N$ denotes the length of the mask, i.e., the number of the FFT bins.

The respective posture masks are shifted onto each of the 140-bin 5736 samples in the normalized dataset the cross-correlation in (5) is then computed. The frame is then classified according to the mask that achieves the highest $X_{l,n}$ value. We highlight that the time taken to create the masks used for classification is not considered in the classification process because the masks are previously defined and used each time a new sample is available. The computation time of this solution for each frame is represented by $\beta_1$, which includes the number of cross-correlation computations, i.e., one cross-cross correlation per class.

### *E. Solution $S_2$ - Deep Learning with Raw Data*

In this solution, we train an artificial neural network (ANN) with the 5000 samples of the dataset. This solution uses the raw data from the dataset without deriving classification features and it is proposed for performance comparison purposes.

The topology of the ANN is illustrated in Figure 3. The ANN input data is a normalized sample, $S_{norm}$ as used in (5), of length 140 bins. The ANN outputs identify the 2 human postures (Class 0 and Class 1). The network is composed of 3 different dense layers adopting a ReLu activation function, followed by a dropout layer parameterized with 20% of dropout.

Regarding the training, after splitting the dataset into *training* and *testing* sets as explained in Section II-B, the 5000 samples of the *training* set are further split into *training* and *validation* subsets in the ratio 80:20. Consequently, the training is done with 4000 training samples and 1000 validation samples, 1000 epochs, and a batch size of 128. The Adam optimizer was adopted in the training.

The computation time to classify a frame in is represented by $\beta_2$ and is based on the time needed to compute the outputs of the ANN for a raw frame of 140 bins, which is represented by $\delta_{pureANN}$ and

$$\beta_2 = \delta_{pureANN}. \tag{6}$$

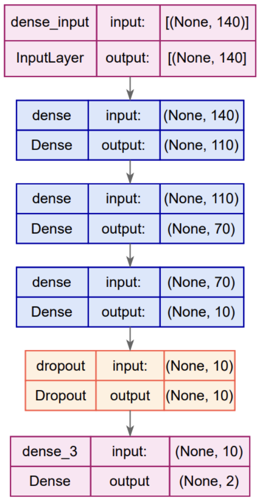
dense_input | input: | [(None, 140)]
InputLayer | output: | [(None, 140]
dense | input: | (None, 140)
Dense | output: | (None, 110)
dense | input: | (None, 110)
Dense | output: | (None, 70)
dense | input: | (None, 70)
Dense | output: | (None, 10)
dropout | input: | (None, 10)
Dropout | output | (None, 10)
dense_3 | input: | (None, 10)
Dense | output | (None, 2)

Fig. 3. ANN architecture adopted in solution $S_2$.

## III. Feature Extraction and Dimensionality Reduction

This section presents the classification solutions that include the derivation of classification features through a variational auto-encoder (VAE).

### *A. Feature Engineering*

The methodology followed to extract classification features from data adopts a VAE that takes as input the raw data vector of a $S_{norm}$ sample, as adopted in solutions $S_1$ and $S_2$. Features are extracted by the VAE's encoder, which is essentially one of the neural networks composing the VAE. At the end of the encoder, 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.

Figure 4 shows the high-level topology of a VAE. The VAE maps an input vector, represented by $x$, onto a Gaussian distribution in the latent space $z$, expressed via the mean, $\mu_{z|x}$, and the standard deviation, $\Sigma_{z|x}$. The role of the encoder is represented by the function $q_\phi(z|x)$. The decoder then samples the latent representation of the input to attempt to produce a recreation of the original input, i.e., $\hat{x}$. The function $p_\theta(x|z)$ represents the target of the decoding operation, i.e., the goal is to maximize the probability of correctness when we attempt to recreate the original input $x$ given a random sample, $z$, from the latent space. The latent space is still of a lower dimension than the input vector.

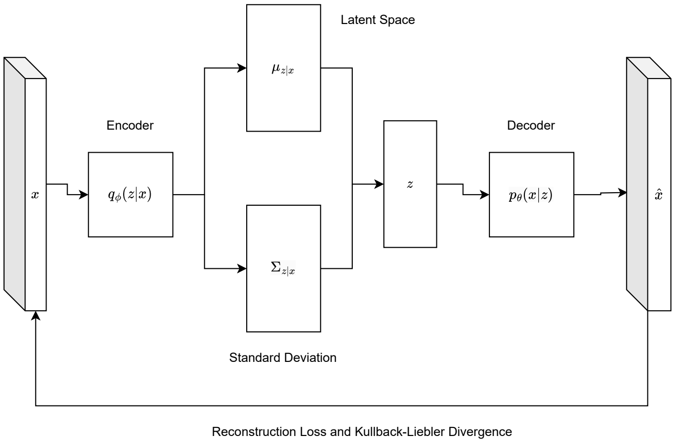

Fig. 4. Variational Auto-Encoder architecture.

In this work, the VAE encoder comprises a simple neural network with three *Dense* hidden layers, with the input layer receiving a 140-bin vector and compressing 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. The VAE model is trained considering a Mean Square Error loss.

### *B. Dimensionality Reduction through Complementary Techniques*

In addition to the dimensionality reduction achieved by the VAE encoder, in this work, we also adopt t-distributed Stochastic Neighbor Embedding (t-SNE) and PCA techniques in solutions $S_4$ and $S_5$, respectively. PCA and t-SNE are applied to the 30 latent variables representing the classification features after being computed by the VAE encoder. Both techniques are parametrized to reduce the 30 latent variables to 3 variables. We highlight that usually, t-SNE is more suited for data visualization purposes than for dimensionality reduction, particularly because when compared to PCA, t-SNE does not ensure the orthogonality of the projected features.

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

### C. *Classification Solutions considering VAE's Encoding*

*1) Solution $S_3$:* The solution $S_3$ is based on the extraction of the 30 classification features through the VAE encoder. The entire dataset, with 5736 samples, is encoded using the VAE model. The sample-wise computation time for VAE encoding, denoted as $\lambda$, is taken into account for execution time comparison.

Regarding the classification, it is performed by an ANN similar to the one adopted in solution $S_2$, and its topology is represented in Figure 5. 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 well as its dimension. The activation functions as well as the training optimizer and methodology are the same as adopted in $S_2$.

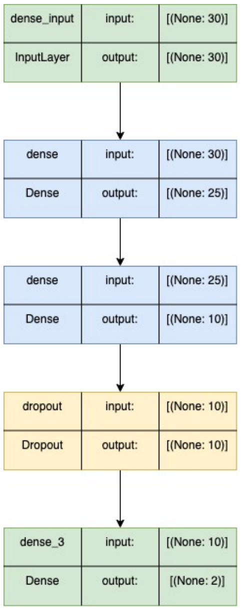

Fig. 5. ANN architecture adopted in solution $S_3$.

Regarding the computation time of the classification scheme, it is denoted as $\delta_{pureVAE}$ and is defined as

$$\beta_3 = \lambda + \delta_{pureVAE}, \quad (7)$$

where $\lambda$ 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 required to compute the ANN output for the 30 input features.

*2) Solution $S_4$:* In this solution, the VAE encoder is used as in solution $S_3$. However, the 30 features obtained through the VAE encoder are reduced to 3 features using the t-SNE method.

The ANN used for classification is similar to the one adopted in $S_3$, but the number of inputs is reduced to 3 while maintaining the rest of the topology and the same training procedure. The computation time for each classification using this solution is denoted as $\beta_4$ and defined as

$$\beta_4 = \lambda + \theta_{TSNE} + \delta_T, \quad (8)$$

where $\theta_{TSNE}$ and $\delta_T$ represent the computation time of the t-SNE method and the ANN computation model, respectively.

*3) Solution $S_5$:* This solution is similar to $S_4$ but instead of using t-SNE, the 30 features generated by the VAE encoder are reduced to 3 adopting the PCA method. The ANN used for classification follows the same topology and training method used in solution $S_4$. Regarding the computation time of this solution, it is denoted as $\beta_5$ and defined as

$$\beta_5 = \lambda + \theta_{PCA} + \delta_T, \quad (9)$$

where $\theta_{PCA}$ represents the computation time of the PCA method.

## IV. Performance Evaluation

This section presents the performance results of the implementation of the proposed solutions. We explore the training and evaluation yields of each solution as well as provide relevant explanations for the results.

Regarding the methodology, the performance results were obtained considering the 736 samples of the test dataset, which were not considered for training purposes. The computation of solutions $S_1$ to $S_5$ was performed on a personal computer with a processor 2.7 GHz Quad-Core Intel Core i7 and 16 GB 2133 MHz LPDDR3 RAM.

In Figure 6, we plot the classification accuracy percentage. This metric describes the rate of successfully classifying the samples in class 0 or class 1 human postures. As observed in the results, the solutions $S_1$ and $S_2$ obtain a similar performance. These solutions use raw data, although based on completely different approaches (cross-correlation vs. deep learning). The adoption of the VAE encoder to extract the 30 features in solution $S_3$ is the technique that achieves the highest accuracy, demonstrating the advantage of adopting the VAE to derive the classification features. However, if the 30 features are reduced to 3 through the t-SNE method in solution $S_4$, the accuracy achieves the lowest value from all solutions. This is mainly because t-SNE is not an adequate method for dimensionality reduction and the advantage of using PCA for the same dimensionality reduction (Solution $S_5$) can lead to results close to the solution achieving the highest accuracy (solution $S_3$).

In Figure 7, we compare the average computation time required to classify each sample for the different solutions. The solution $S_1$ achieves the lowest computation time. However, it is not scalable because the computation time is a function of the number of classes in the classification domain, as the classification requires the computation of a cross-correlation for each class. Solution 2 requires the highest computation time, as it uses raw data, and its dimension, particularly the 140 bins, increases the computation time of the ANN adopted in the classification. The computation time is reduced from solution $S_2$ to $S_3$ because the adoption of the VAE encoder decreases the number of inputs of the classification ANN. Finally, the solutions $S_4$ and $S_5$ achieve a similar computation time, as the computation times of t-SNE and PCA methods are similar.

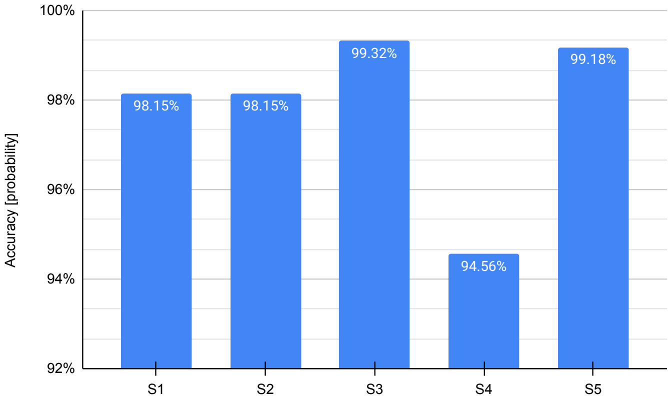

Fig. 6. Comparison of classification accuracy.

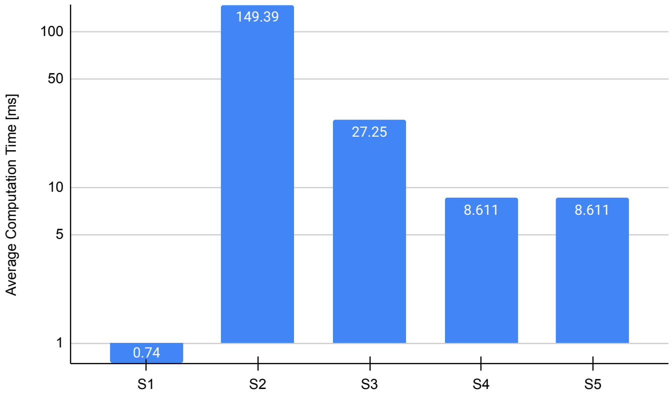

Fig. 7. Comparison of execution times $\beta_1, \beta_2, \beta_3, \beta_4$, and $\beta_5$ (in ms).

As conclusion, the VAE extract features that increase the classification accuracy performance but the adoption of additional dimensionality reduction methods such as PCA on top of the VAE encoder can lead to a slight decrease of accuracy but a higher gain in terms of computation time.

## V. Conclusions

This paper explores the performance comparison of five posture detection and classification solutions based on classification accuracy and computation time. Using the aforementioned solutions as a vessel, a light is further shone on the impact of feature extraction using VAEs. We have presented two solutions that do not implement feature engineering: $S_1$, which uses the cross-correlation filter, and $S_2$, which implements a pure ANN solution based on raw data. These two solutions have then been compared to $S_3$, $S_4$, and $S_5$ that do implement VAE-based feature extraction. Following the detailed discussions in Section IV, we can arrive at the definite conclusion that it is advantageous to include feature engineering in the posture detection and classification chain. The solutions $S_3$ and $S_5$ outperform $S_1$ and $S_2$ in terms of classification accuracy. This margin makes the case for the VAE-based solutions, despite $S_1$ spotting the lowest execution time. The deficit displayed in execution time is quite notable, but compared to the execution time of $S_1$, the VAE-based solutions are still spotting high accuracies while not being dependent on the number of classes defined in the classification domain.

## Acknowledgements

This work was supported by *Fundação para a Ciência e Tecnologia* under the projects CELL-LESS6G (2022.08786.PTDC) and UIDB/50008/2020.

## References

[1] Pascale Sévigny. Joint through-wall 3-d radar imaging and motion detection using a stop-and-go sar trajectory. In *2016 IEEE Radar Conference (RadarConf)*, pages 1–5, 2016.

[2] Vida Adeli, Ehsan Adeli, Ian Reid, Juan Carlos Niebles, and Hamid Rezatofighi. Socially and contextually aware human motion and pose forecasting. *IEEE Robotics and Automation Letters*, 5(4):6033–6040, 2020.

[3] Enric Corona, Albert Pumarola, Guillem Alenyà, and Francesc Moreno-Noguer. Context-aware human motion prediction. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6990–6999, 2020.

[4] Ricardo Cruz, Antonio Furtado, and Rodolfo Oliveira. Assessment of feature selection for context awareness rf sensing systems. *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, pages 1–6, 2022.

[5] Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu. Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. *ACM Computing Surveys*, 54(4), 2021.

[6] Neil C. Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F. Manso. The computational limits of deep learning. 2020.

[7] A. M. S. Muniz, E. F. Manfio, M. C. Andrade, and J. Nadal. Principal component analysis of vertical ground reaction force: A powerful method to discriminate normal and abnormal gait and assess treatment. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2683–2686, 2006.

[8] Deepalakshmi Balasubramanian, Poonguzhali Srinivasan, and Ravindran Gurupatham. Automatic classification of focal lesions in ultrasound liver images using principal component analysis and neural networks. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2134–2137, 2007.

[9] Wenjie Song, Yi Yang, Mengyin Fu, Fan Qiu, and Meiling Wang. Real-time obstacles detection and status classification for collision warning in a vehicle active safety system. *IEEE Transactions on Intelligent Transportation Systems*, 19(3):758–773, 2018.