

# The Effect of Window Size and Shape in STFT for Pre-Processing FMCW Radar Data in Human Activity Recognition

Figgo Azzam De Fitrah<sup>a</sup>, Fiky Y. Suratman<sup>a,b,\*</sup>, Istiqomah<sup>a,b</sup>

<sup>a</sup>School of Electrical Engineering  
Telkom University

Jl. Telekomunikasi. 1, Terusan Buahbatu – Bojongsong  
Bandung, Indonesia

<sup>b</sup>Pusat Unggulan Iptek-Perguruan Tinggi (PUI-PT) Intelligent Sensing-IoT  
Telkom University

Jl. Telekomunikasi. 1, Terusan Buahbatu – Bojongsong  
Bandung, Indonesia

## Abstract

Many studies use radars for Human Activity Recognition (HAR), and numerous techniques for preprocessing FMCW radar data have been explored to improve HAR performances. Our approach employs 1-D radar to classify four human activities, i.e., walking, standing, crouching, and sitting. We use Fast Fourier Transform (FFT) and Short-Time Fourier Transform (STFT) with Kaiser window to generate range-time and Doppler-time data from in-phase and quadrature (I/Q) radar signals. Common belief considers that the choice of windowing parameters, i.e., window size and window shape represented by the beta parameter in Kaiser window, has significant impacts on the performances of deep learning LSTM models, such as measured by the F1-score. However, our study in this paper using t-tests statistical analysis shows otherwise. Our results consistently support the null hypothesis, which means that variations in window size and window shape do not significantly affect the F1-score. In essence, our findings underscore the robustness of our preprocessing methodology, emphasizing the stability and reliability of the selected configurations. This research provides valuable insights into the preprocessing techniques for radar data in the context of human activity recognition, enhancing the consistency and credibility of deep learning models in this domain.

**Keywords:** Human Activity Recognition, Radar, Preprocessing Techniques, Kaiser Windowing

## I. INTRODUCTION

Human Activity Recognition (HAR) is a system for identifying, monitoring, and analyzing the behavior of a person (or more) to infer the type of activity [1]. Estimating and classifying human activity has proven challenging since human activity is highly complex and dynamic. HAR can be used in various applications, such as smart environment, monitoring/surveillance for security, remote health/telemedicine, assisted living/aging-in-place, sports, and human-computer interaction [2].

Conventional approaches to HAR using wearable sensors, such as accelerometers, gyroscopes, and optical sensors/oximetry, have several drawbacks, such as discomfort due to being attached to the user's body, prone to damage due to contact, and it might be unusable for users suffering from burns or injuries. Camera-based HAR also has constraints since cameras are sensitive to lighting and privacy aspects (e.g., monitoring in the bathroom or bedroom). Studies on HAR using cameras can be found in [3]–[6], while a comprehensive survey on HAR with sensors other than

radar can be found in [7] and references therein.

This study uses a different approach with the aim of overcoming the above constraints. Here, we use radar instead of wearable sensors [2]. Radar can be used for HAR with the advantages of being safe, contactless, versatile, portable, and cheap [8]. Radar-based HAR still relies heavily on preprocessing to represent data in the form of spectrogram or micro-Doppler signature (Doppler-Time). Spectrograms can reflect unique patterns due to target movement [9]–[16]. Spectrograms based on Short Time Fourier Transform (STFT) are popular due to their low computational complexities compared to other types of time-frequency (TF) analysis [17].

In our previous studies, such as [18] and [19], Frequency Modulated Continuous Wave (FMCW) radar was used for HAR with preprocessing based on Fourier Transform. It was applied both to fast time (one frame of the received signal) to get beat frequencies which correspond with target ranges and to slow time to produce Doppler frequency for the same range [20]. The resulting 2D data is range-time and range-Doppler, which is then used as input for deep learning algorithms using Convolutional Neural Networks (CNN).

In this paper, we explore Linear Time Frequency (TF) Analysis with STFT applied to the slow time for each range in radar data so that the physical characteristics of the target, such as Doppler and micro-

\* Corresponding Author.

Email: fysuratman@telkomuniversity.ac.id

Received: November 15, 2023 ; Revised: February 4, 2024

Accepted: March 11, 2024 ; Published: August 31, 2024

Doppler due to the movement of the human as a function of time (Doppler-time) will be obtained in more detail. The set of frames generated from the preprocessing above is then formed into a sequence of 2D data matrices. The sequence will be used as input to the deep learning algorithm, namely Long Short-Term Memory (LSTM). Assuming the sequence is a time series, LSTM can be used to capture temporal information from sequential data. LSTM could learn temporal dependencies on sequential data, which is essential for classifying human activities.

It is widely known in signal processing, particularly in spectral analysis, that the window size and window shape of STFT will affect the accuracy of estimating signal frequency spectrum (spectrogram) [21]. The window size is related to the resolution of the spectrogram. The larger the window size, the better the resolution. However, using a larger window size increases computational time [22], [23]. Meanwhile, the window shape has an impact on the width of the main lobe and how large the amplitude of the side lobe is relative to the main lobe in the frequency domain. The trade-off is that the side lobes' magnitude in relation to the main lobe decreases as the main lobe grows wider, and vice versa. For example, in a rectangular window, the side lobes are wider than the main lobe, which results in improved frequency resolution but higher oscillations during frequency transitions.

On the contrary, the direct consequences of using different window sizes and window shapes of the STFT to HAR performances are rarely discussed in the literature. Thus, investigating the effect of window size and window shape of the STFT to HAR classification performances is well-motivated. Several windows functions are often used in STFT, such as rectangular, Hamming, Hanning, Blackman, and Kaiser [24]. However, in this study, the Kaiser window function is considered as a one-parameter family of window functions. The Kaiser window with a parameter  $\beta$  is used to represent window shapes that might be produced from various existing window functions. By adjusting the parameter and the window size of the STFT in the radar data preprocessing, we aim to determine how significant these variations affect the results when feeding the data into a deep learning algorithm, specifically an LSTM model for classifying human activities. Here, we use the F1-score performance measure to evaluate the variation of window size and shape of STFT on the deep learning algorithm. This study uses the t-test statistical method to decide whether the effect of window size and shape on the F1-score is significant or not.

We used uRAD FMCW Raspberry Pi v1.2 radar [25], integrated with a Raspberry Pi 4 Model B, to collect radar data. In general, our objective is to identify the key parameters that yield meaningful results in preprocessing FMCW radar data for subsequent utilization in LSTM models. The study is a part of our intensive investigations for searching the optimal preprocessing configurations of FMCW radar data to enhance its applicability in human activity recognition using deep learning algorithms.

## II. METHODS

The method employed in this study consists of three phases, i.e., data acquisition, preprocessing, and deep learning model training. The data acquisition involves the collection of radar data, followed by preprocessing to eliminate noise and clutter that are not related to human activity and to extract essential range-time and Doppler-time 2D data from the radar signals [2]. Lastly, training the deep learning model is delivered by using the pre-processed 2D data as the input. Overall, we used Python programming language [26], from radar data acquisition with the Raspberry Pi 4 Model B to the training of the deep learning model.

### A. Data Acquisition

We utilized the uRAD Raspberry Pi v1.2 radar, integrated into a Raspberry Pi 4 Model B, to collect raw data signals. Table 1 provides the radar specifications. The radar was configured in sawtooth mode, with an  $f_0$  24.045 GHz and a signal bandwidth of 200 MHz [25]. We use sawtooth mode to acquire this range data. The radar was programmed to generate 30 frames per second (fps) with 50 samples for each frame. The output data consists of complex signals represented as In-phase (I) and Quadrature (Q) components [27].

The radar was positioned 2 meters from a human target, as illustrated in Figure 1. Activities were conducted inside a detection area, and we used a chair to simulate sitting scenarios (not shown).

TABLE 1. SPECIFICATION OF URAD RASPBERRY PI 4 MODEL B

Specifications	
Frequency	24—24,25 GHz ISM
Field of view	$\pm 15$ deg
connector	Raspberry pins
Voltage	5V
Current	170 mA
Distance accuracy	$\pm 4$ cm or 0.3%
Velocity accuracy	0.05 m/s

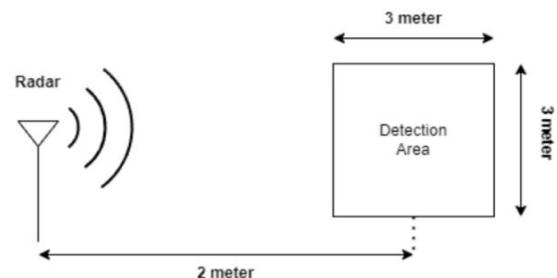


Figure 1. Radar Placement and Human Target, the Box Represents Area for the Target Movement (Detection Area).

TABLE 2. SCENARIO OF HUMAN ACTIVITIES

Scenario 1	Scenario 2	Scenario 3	Scenario 4
Walking	Crouching	Standing	Walking
Crouching	Walking	Walking	Standing
Standing	Sitting	Sitting	Crouching
Sitting	Standing	Crouching	Sitting

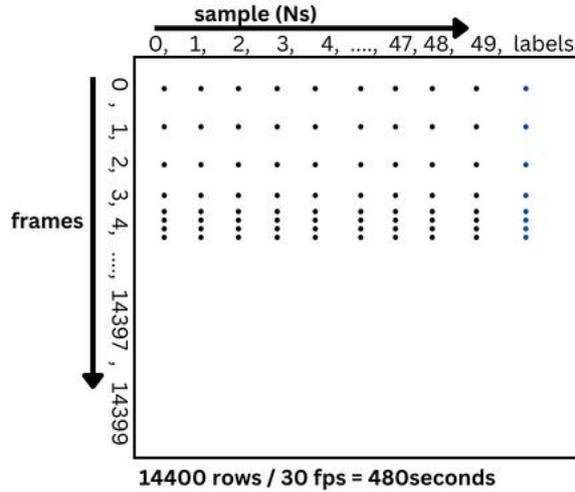


Figure 2. Visualization of Data Frame

The scenario for human activity includes walking, standing, sitting, and crouching, as detailed in Table 2. Three participants were involved, following a prescribed sequence of movements for the respective scenario within the 30-second intervals. Thus, each participant performed activities for 120 seconds. Subsequently, the collected data from each participant was combined, along with the empty datasets, to create a dataset where participants were outside the radar's detection range. This resulted in a dataset comprising 14400 rows and 51 columns (14400×51), with the last columns being the label of the data. Figure 2 provides a visualization of the data in CSV format.

## B. Preprocessing method

We applied some preprocessing steps to refine the acquired radar data, isolating the relevant information pertaining to HAR. The preprocessing includes several key steps, i.e., ADC scaling, DC component removal, Static Clutter removal, FFT and STFT. Figure 3 illustrates the sequence of the preprocessing steps before applying a deep learning algorithm, namely Bi-LSTM. We use the Numpy library to apply the preprocessing steps [28].

Our preprocessing pipeline begins with ADC scaling, a pivotal step aimed at normalizing raw signal data from the uRAD Raspberry Pi v1.2 radar to 12 bits. This normalization process is essential to align the data within a range of 0 – 4095, corresponding to the actual voltage of the signal, given that the ADC value of the Raspberry Pi 4 Model B operates at 3.3 V [29].

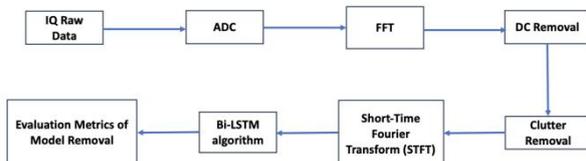


Figure 3. Preprocessing Steps

Equation (1) is applied to facilitate this data normalization.

$$x = \frac{x_{Ns}}{ADC_{intervals}} \times V_{ref}, \quad (1)$$

where  $x_{Ns} \in [0, 4095]$  is a sample,  $V_{ref}$  is a reference voltage 3.3 V,  $ADC_{intervals} = 4096$ .

After data scaling, we apply Fast-Fourier Transform (FFT) to the data  $x$  in each row to extract beat frequencies which correspond to range information (range bin) of the objects perceived by the radar [7], using Equation (2).

$$X_k = \sum_{n=0}^{N_s-1} x[n] e^{-j2\pi \frac{n}{N_s} [k]}, \quad (2)$$

$k = 0, 1, 2, \dots, K-1$  is frequency or range bin index, and  $K \geq N_s$  is the FFT size. By accumulating  $X_k$  as a baseband signal over a coherent processing interval  $L_c$  ( $L_c$  number of frames), we get the change in distance of an object in the respective interval. Note that we use Equation (3) to get the distance  $d$  (in meter) from the frequency index [25].

$$d(k) = 75 \times \frac{N_s}{BW} \times \frac{k}{K}, \quad (3)$$

where  $BW$  denotes bandwidth of signal. The output of the FFT is 2D dataset represent range-time data.

DC component removal is done over range bin ( $k = 0, 1, 2, \dots, K-1$ ) to eliminate the DC component, which is contributed by the GPIO pin of the Raspberry Pi 4 Model B [29]. Equation (4) serves as the mechanism for eliminating this DC value.

$$Y_{Ns} = X_{Ns} - \frac{1}{K} \sum_{k=0}^{K-1} X_k, \quad (4)$$

where  $X_{Ns} = [X_0, X_1, X_2, \dots, X_{K-1}]$  represents each row in the data frame of Figure 2 after the FFT in (2), and  $Y_{Ns} = [Y_0, Y_1, Y_2, \dots, Y_{K-1}]$ . Following DC component removal, static clutter suppression is delivered to remove clutter for each range bin over the intervals  $L_c$  from the surrounding environment, such as static objects like walls within the room, using Equation (5).

$$Z_{L_c} = Y_{L_c} - \frac{1}{L_c} \left[ \sum_{l=0}^{L_c-1} Y_l \right], \quad (5)$$

where  $Y_{L_c} = [Y_0, Y_1, Y_2, \dots, Y_{L_c-1}]$  and  $Z_{L_c} = [Z_0, Z_1, Z_2, \dots, Z_{L_c-1}]$ .

Furthermore, we employ STFT for each range bin along the time axis using Kaiser windowing functions. Equation (6) outlines the application of STFT with the chosen window  $w[n]$ .

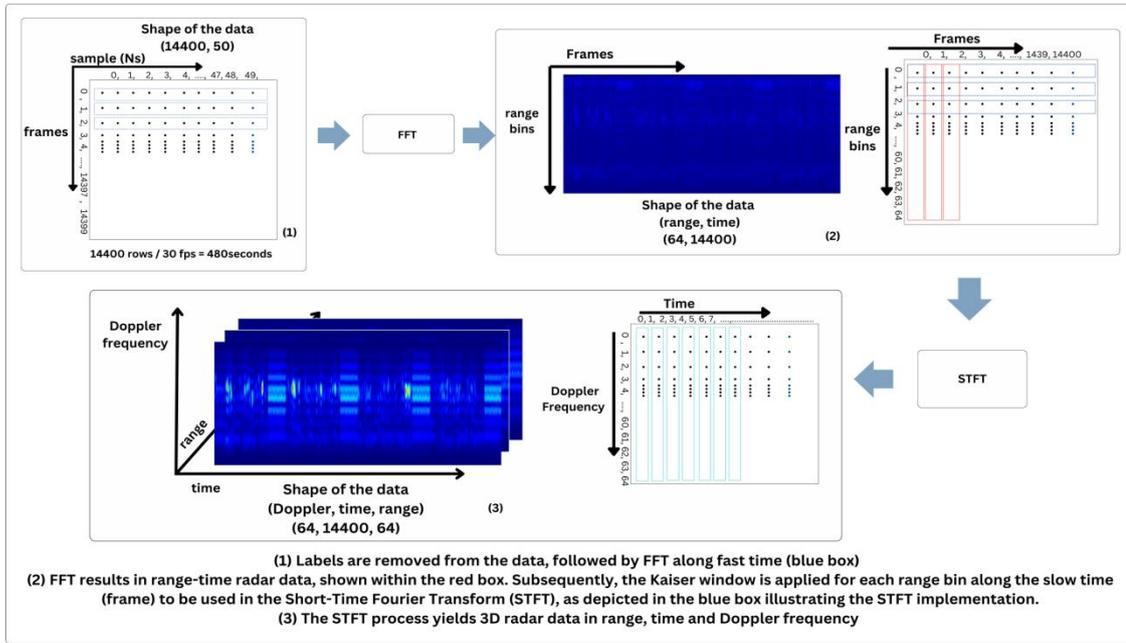


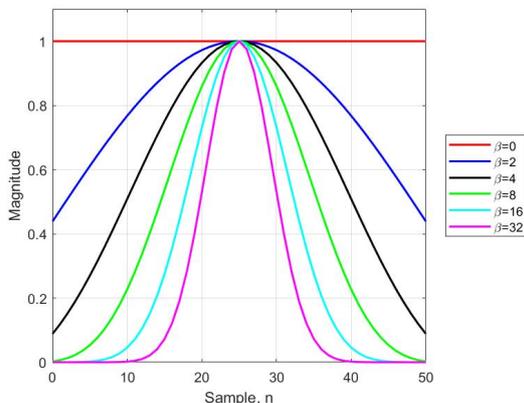
Figure 4. Preprocessing Using FFT and STFT

$$Z[l, m] = \sum_{n=0}^{N-1} Z[n + mH]w[n]e^{-j2\pi\frac{n}{N}l}, \quad (6)$$

where  $l = 0, 1, \dots, \frac{N}{2}$  is the frequency index, and  $N$  is the window size. Here,  $m = 0, 1, \dots, M$  represents time index, and  $M$  is the maximum time index such that the window's time range is fully contained in the signal's time range. Equation (6) introduces parameter  $H$ , referred to as stride, which determines the step size in which the window is to be shifted across the signal.

Figure 4 shows how FFT and STFT are being applied to produce 3D data in range, time, and Doppler frequency. Note that the window function that we use in this work is the Kaiser window, as written in Equation (7).

$$w[n] = \frac{I_0 \left[ \beta \sqrt{1 - \frac{4n^2}{(N-1)^2}} \right]}{I_0[\beta]}, \quad (7)$$


 Figure 5. The Shapes of Kaiser Window for Various Values of  $\beta$ 

where  $I_0$  denotes zeroth order Bessel function,  $\beta$  is a parameter to define the window shape [25], and  $N$  represents the window size. Figure 5 represents the shapes of Kaiser window for  $\beta = 2, 4, 8, 16$ , and  $32$ , with  $0 \leq n \leq N-1$  and  $N = 51$ . Figure 5 shows that the higher the value of  $\beta$  the tighter the window.

Up to this point, data acquisition and preprocessing steps have been fully explained. Next, we proceed with the deep learning method that we used for human activity classification.

### C. Deep learning method

In deep learning, LSTM stands as a pivotal algorithm within the Recurrent Neural Network (RNN) family [30]. RNNs, characterized by self-feedback loops that provide contextual information between input and output layers, underpin the foundation of LSTM. LSTM serves a crucial role in overcoming the vanishing gradient problem commonly associated with RNNs. However, it is important to acknowledge that LSTM has its own set of limitations. To address these shortcomings, we adopt Bidirectional LSTM (Bi-LSTM) [31], a variant designed to enhance the model's performance. The architectural layout of our chosen models is illustrated in Figure 6.

Before training the LSTM model, it is imperative to transform the preprocessed data. This transformation involves organizing the data into a windowed dataset and stride 15, with the total number of windows mirroring the window count employed in the STFT phase (e.g., 32 windowed STFT corresponds to a windowed dataset of 32 windows). Then, we apply a one-hot encoder for the data. Subsequently, the dataset is divided into training and testing sets, maintaining an 80:20 ratio which are 766 for train and 192 for test. Figure 7 shows the distribution of data in training sets, and Figure 8 shows the distribution of data in test sets. We employ validation for training from train sets with amount of 0.1 from training sets.

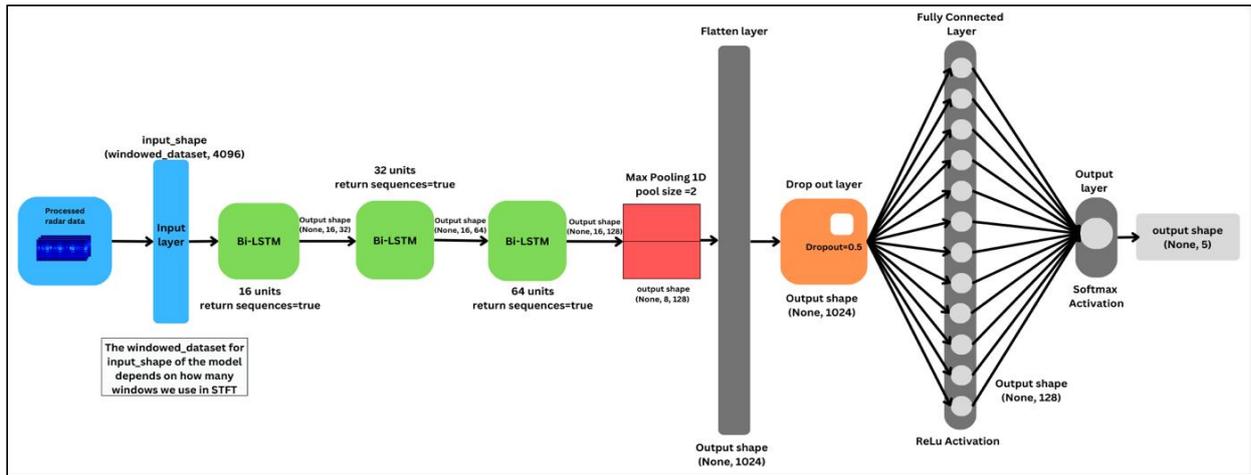


Figure 6. Architecture of Deep Learning

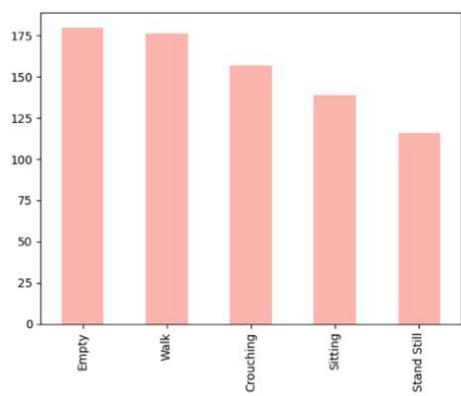


Figure 7. Train Distribution Data

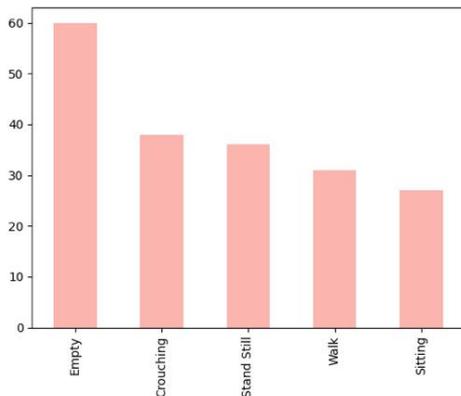


Figure 8. Test Distribution Data

To ensure the data's suitability for deep learning, we employ robust scaling via the scikit-learn library in Python [32]. This scaling technique effectively mitigates the impact of outliers in the data to make it ready for integration into the deep learning algorithm.

### III. RESULTS AND DISCUSSION

We have trained and tested our model with the highest validation accuracy, and the losses are 0.921 and 0.2435, respectively. Figure 9 shows model loss and accuracy for the training data and validation data as functions of time-epoch. The figures also indicate that overfitting has been avoided. Figure 10 shows how our model predicts for every activity in a confusion matrix,

where W is walking, E is empty, C is crouching, T is standing, and S is sitting. Our models show high performance for the detection of Empty classes, but the accuracy parameter is not the right answer to this problem because of imbalanced data [33]. In our evaluation to determine the effect of STFT window size and window shape on the model performance, we use the F1-score as the metric. This is because the F1-score could strike a balance between precision and recall, which is particularly valuable when dealing with imbalanced datasets. Equation (8) outlines the methodology for computing the F1-score [30].

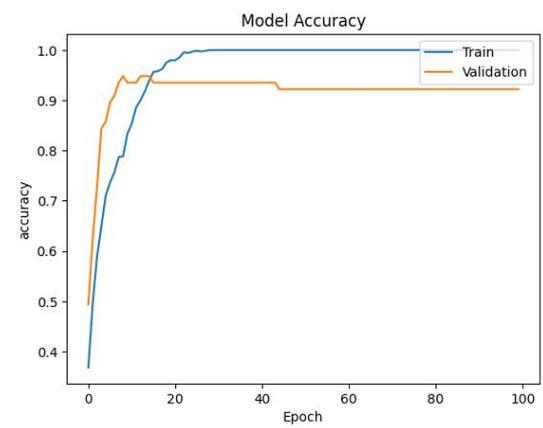
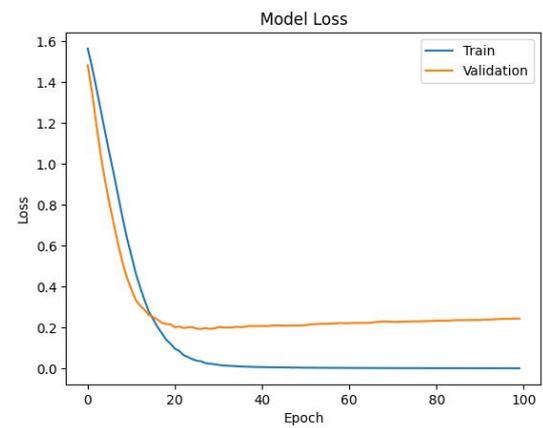


Figure 9. Model Loss and Accuracy as Functions of Time Epoch

TABLE 3. F1-SCORE AS A FUNCTION OF WINDOW SIZE, WINDOW SHAPE, AND ACTIVITY

Performance metric (F1-score)												
Activity	Standing						Sitting					
Window Size	Window Shape ( $\beta$ parameter)											
	0	2	4	8	16	32	0	2	4	8	16	32
2	0,36	0,23	0,11	0,33	0,19	0,00	0,87	0,83	0,87	0,85	0,86	0,84
4	0,68	0,54	0,59	0,66	0,61	0,68	0,82	0,73	0,79	0,83	0,75	0,76
8	0,71	0,63	0,78	0,8	0,79	0,69	0,74	0,63	0,88	0,81	0,8	0,82
16	0,7	0,76	0,77	0,81	0,76	0,81	0,71	0,81	0,82	0,77	0,72	0,81
32	0,77	0,73	0,8	0,78	0,81	0,69	0,68	0,62	0,72	0,7	0,74	0,61
Activity	Walking						Crouching					
2	0,64	0,58	0,58	0,6	0,57	0,54	0,89	0,58	0,89	0,9	0,57	0,89
4	0,67	0,6	0,65	0,65	0,66	0,71	0,91	0,9	0,93	0,91	0,93	0,91
8	0,75	0,74	0,78	0,79	0,79	0,74	0,89	0,9	0,87	0,89	0,88	0,88
16	0,79	0,76	0,71	0,75	0,78	0,71	0,95	0,91	0,9	0,93	0,89	0,9
32	0,74	0,83	0,84	0,82	0,83	0,77	0,89	0,91	0,93	0,91	0,93	0,88
Activity	No Target											
2	0,99	0,99	0,99	0,99	0,99	0,99						
4	0,99	1,00	0,99	0,99	1,00	0,99						
8	1,00	1,00	1,00	1,00	1,00	1,00						
16	1,00	1,00	1,00	0,99	1,00	1,00						
32	0,97	0,98	0,97	0,99	0,99	0,99						

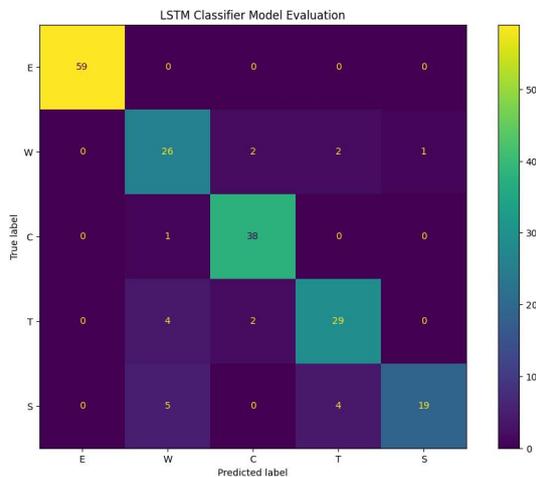


Figure 10. Confusion Matrix

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (8)$$

We measure the F1-score of the LSTM model using various STFT window sizes 2, 4, 8, 16, and 32 and Kaiser-window beta parameters 0, 2, 4, 8, 16, and 32. The results of the F1-score for four different activities (standing, sitting, walking, crouching, and no human target scenario) are shown in Table 3.

Table 3 shows that, in general, the variation of F1-scores as functions of window size and window shape is not significantly large. Except for the F1-score values in standing and walking activities with window sizes 2 and 4. The highest value of F1-scores with the slightest variation occurs in the scenario with no target. This is reasonable because the 2D radar dataset captured from no target scenario is significantly different from the dataset with target activities.

To be more conclusive and to pinpoint the parameters significantly impacting the F1-score (if any), we conduct paired two-sample t-tests [34]. These tests evaluate the null hypothesis ( $H_0$ ), which states that variations in window size or window shape do not result

in discernible changes in the F1-score. Simultaneously, we examine the alternative hypothesis ( $H_1$ ), suggesting potential F1-score alterations. Our choice of significance level is  $\alpha=0.05$  for the t-tests.

To evaluate the effect of window size on F1-score using paired two-sample t-tests, we use various window shapes  $\beta \in [0, 2, 4, 8, 16, 32]$  for each paired window sizes being compared. This is to make sure that the change in F1-score is only due to the difference in window size, not the beta value. In vice versa, we use various window sizes  $N \in [2, 4, 8, 16, 32]$  for each paired window shapes being compared, to evaluate the effect of window shape on F1-score. We include all five activities in the dataset (standing, sitting, crouching, walking and no target presents, as shown in Table 3). Note that we only compare commonly used window sizes of 8, 16, and 32 against each other, and window shapes of 0, 16, and 32 against each other, as shown in Table 4 and 5, respectively.

Table 4 presents the outcomes of t-tests for different window sizes employed in STFT. These are comparisons for 8 vs. 16, 8 vs. 32, and 16 vs. 32 window sizes, providing insights into the p-values and t-statistics obtained from our t-tests. From the table, all comparisons show t-statistic values that are smaller than the critical t-value (or a p-value that is greater than the significance level). This implies that the null hypothesis ( $H_0$ ) is accepted and so there is no significant difference in F1-score due to the different window sizes of 8, 16 and 32.

Similar results are produced by the t-tests to see the effect of window shape on F1-score, as shown in Table 5. For each window shape comparison, the t-statistic is always smaller than the t-critical value (p-value greater than the significance level). The results collectively

TABLE 4. RESULTS OF T-TESTS FOR DIFFERENT WINDOW SIZES, OBSERVATIONS = 30

Paired window sizes	Sig. level ( $\alpha$ )	p-value	t-stat (one-tail)	t-critical (one-tail)
8 vs 16	0.05	0.221	-0.778	1.699
8 vs 32	0.05	0.350	0.386	1.699
16 vs 32	0.05	0.175	0.946	1.699

TABLE 5. RESULTS OF T-TESTS FOR DIFFERENT WINDOW SHAPES (B PARAMETER), OBSERVATIONS = 25

Paired window sizes	Sig. level ( $\alpha$ )	p-value	t-stat (one-tail)	t-critical (one-tail)
0 vs 16	0.05	0,253	0,675	1,711
0 vs 32	0.05	0,124	1,184	1,711
16 vs 32	0.05	0,314	0,490	1,711

indicate that the t-tests support the null hypothesis ( $H_0$ ), confirming that variations in window shapes do not significantly impact the F1-score.

The application of t-test is aimed to rigorously evaluate the statistical significance of the window size and window shape in relation to our deep learning model. It is worth emphasizing that both the window size and window shape were subjects of interest in our analysis. The outcome of these statistical tests is pivotal in shaping our understanding of the relationship between these parameters and the F1-score. It is noteworthy that the results consistently supported the null hypothesis, with p-values exceeding the predefined significance level, denoted as  $\alpha$ . This suggests that the variations in window sizes and window shapes do not exert a statistically significant influence on the F1-score. This provides valuable insights into consistent performance of our deep learning models in the context of human activity recognition using radar data.

#### IV. CONCLUSION

In conclusion, our comprehensive evaluation of the window size and window shape employed within the Short-Time Fourier Transform (STFT) process has shed light on their influence on the training of deep learning models for radar data analysis. Specifically, we examined the impact of these parameters on the F1-score, a crucial metric for assessing model performance in human activity recognition. Our results reveal that neither the quantity of window size nor window shape exhibited a statistically significant effect on the F1-score. This conclusion is substantiated by the t-tests we conducted, where the calculated p-values consistently exceeded the predetermined significance level ( $\alpha$ ). This underscores the reliability of our deep learning models in their ability to effectively analyze radar data for human activity recognition.

#### DECLARATIONS

##### Conflict of Interest

The authors have declared that no competing interests exist.

##### CRediT Authorship Contribution

Figo Azzam De Fitriah: Methodology, Software, Validation, Investigation, Data Curation, Writing-Original Draft; Fiky Yosef Suratman: Conceptualization, Methodology, Formal Analysis, Resources, Writing-Review & Editing, Supervision and Funding Acquisition; Istiqomah: Methodology, Software, Data Curation, Project Administration.

##### Funding

Research reported in this publication was supported by Ministry of Education, Culture, Research and Technology Republic of Indonesia, under grant number 180/E5/PG.02.00.PL/2023.

#### Acknowledgment

The authors would like to thank Azhar Yunda Ramadhan and Muhammad Adi Nurhidayat from School of Electrical Engineering, Telkom University, for their assistances on producing datasets and fruitful discussions on deep learning models.

#### REFERENCES

- [1] S. W. Kang, M. H. Jang, and S. Lee, "Identification of human motion using radar sensor in an indoor environment," *Sensors*, vol. 21, no. 7, Apr. 2021, doi: 10.3390/s21072305.
- [2] S. Z. Gurbuz and M. G. Amin, "Radar-based human-motion recognition with deep learning: promising applications for indoor monitoring," *IEEE Signal Process Mag.*, vol. 36, no. 4, pp. 16–28, Jul. 2019, doi: 10.1109/MSP.2018.2890128.
- [3] R. Raja, N. A. Subramanian, and V. Vasudevan, "HARDeep: design and evaluation of a deep ensemble model for human activity recognition," *Int. J. Innov. Comput. Appl.*, vol. 14, no. 3, pp. 155–166, 2023, doi: 10.1504/IJICA.2023.131356.
- [4] Y. Ren, C. Zhu, and S. Xiao, "Deformable faster r-cnn with aggregating multi-layer features for partially occluded object detection in optical remote sensing images," *Remote Sens (Basel)*, vol. 10, no. 9, p. 1470, Sep. 2018, doi: 10.3390/rs10091470.
- [5] A. Markman, X. Shen, and B. Javidi, "Three-dimensional object visualization and detection in low light illumination using integral imaging," *Opt. Lett.*, vol. 42, no. 16, p. 3068, Aug. 2017, doi: 10.1364/OL.42.003068.
- [6] W. Bouachir, R. Gouiaa, B. Li, and R. Noumeir, "Intelligent video surveillance for real-time detection of suicide attempts," *Pattern Recognit. Lett.*, vol. 110, pp. 1–7, Jul. 2018, doi: 10.1016/j.patrec.2018.03.018.
- [7] M. H. Arshad, M. Bilal, and A. Gani, "Human activity recognition: review, taxonomy and open challenges," *Sensors*, vol. 22, no. 17, p. 6463, Aug. 2022, doi: 10.3390/s22176463.
- [8] Ed., M. G. Amin, *Radar for indoor monitoring: Detection, classification, and assessment*. Boca Raton, FL, USA: CRC Press Taylor & Francis Group, 2017. doi: 10.1201/9781315155340.
- [9] C. Karabacak, S. Z. Gurbuz, A. C. Gurbuz, M. B. Guldogan, G. Hendeby, and F. Gustafsson, "Knowledge exploitation for human micro-doppler classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 10, pp. 2125–2129, Oct. 2015, doi: 10.1109/LGRS.2015.2452311.
- [10] Y. Kim and T. Moon, "Human detection and activity classification based on micro-doppler signatures using deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 1, pp. 8–12, Jan. 2016, doi: 10.1109/LGRS.2015.2491329.
- [11] C. Ding *et al.*, "Continuous human motion recognition with a dynamic range-doppler trajectory method based on fmcw radar," *IEEE Trans. Geosci. Remote Sens. Lett.*, vol. 57, no. 9, pp. 6821–6831, Sep. 2019, doi: 10.1109/TGRS.2019.2908758.
- [12] X. Qiao, G. Li, T. Shan, and R. Tao, "Human activity classification based on moving orientation determining using multistatic micro-doppler radar signals," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2022, doi: 10.1109/TGRS.2021.3100482.
- [13] C. Li, V. M. Lubecke, O. Boric-Lubecke, and J. Lin, "A review on recent advances in doppler radar sensors for noncontact healthcare monitoring," *IEEE Trans. Microw. Theory Tech.*, vol. 61, no. 5, pp. 2046–2060, May 2013, doi: 10.1109/TMTT.2013.2256924.
- [14] W. Ding, X. Guo, and G. Wang, "Radar-based human activity recognition using hybrid neural network model with multidomain fusion," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 5, pp. 2889–2898, Oct. 2021, doi: 10.1109/TAES.2021.3068436.
- [15] X. Li, Y. He, and X. Jing, "A survey of deep learning-based human activity recognition in radar," *Remote Sensing*, vol. 11, no. 9, MDPI AG, May 01, 2019, doi: 10.3390/rs11091068.
- [16] S. Ahmed, J. Park, and S. H. Cho, "FMCW radar sensor based human activity recognition using deep learning," in *2022 International Conference on Electronics, Information, and Communication (ICEIC)*, Feb. 2022, pp. 1–5. doi: 10.1109/ICEIC54506.2022.9748776.

- [17] L. Cohen, "Time-frequency distributions-a review," *Proceedings of the IEEE*, vol. 77, no. 7, pp. 941–981, Jul. 1989, doi: 10.1109/5.30749.
- [18] R. Rafli, F. Y. Suratman, and Istiqomah, "FMCW radar signal processing for human activity recognition with convolutional neural network," 2023, pp. 429–445. doi: 10.1007/978-981-99-0248-4\_29.
- [19] D. Radhityo Sardjono, F. Y. Suratman, and Istiqomah, "Human motion change detection based on fmcw radar," in *2022 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*, Dec. 2022, pp. 1–6. doi: 10.1109/APWiMob56856.2022.10014226.
- [20] M. A. Richards, *Fundamentals of radar signal processing*. New York: McGraw-Hill Education, 2022.
- [21] J. Smith and J. O. Smith III, "PARSHL: an analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation," 2005. Accessed: Feb. 07, 2024. [Online]. Available: [https://ccrma.stanford.edu/~jos/parshl/Analysis\\_Window\\_Step\\_1.html](https://ccrma.stanford.edu/~jos/parshl/Analysis_Window_Step_1.html)
- [22] B. Kim, S.-H. Kong, and S. Kim, "Low computational enhancement of stft-based parameter estimation," *IEEE J. Sel. Top Signal Process*, vol. 9, no. 8, pp. 1610–1619, Dec. 2015, doi: 10.1109/JSTSP.2015.2465310.
- [23] "Numpy.kaiser - numpy v1.26." Accessed : Feb. 07, 2024. [Online]. Available: <https://numpy.org/doc/stable/reference/generated/numpy.kaiser.html>.
- [24] A.V. Oppenheim, *Discrete-time signal processing*. Pearson Education India, 1999.
- [25] "URAD user manual raspberry pi version software sdk v1.1., anteral, navvara, spain," 2021.
- [26] Python.org, "Python." 1990. Accessed: Sep. 05, 2023. [Online]. Available: <https://www.python.org/>
- [27] E. S.Z. Gurbuz, *Deep Neural Network Design for Radar Applications*. SciTech Publishing, 2020.
- [28] "Numpy 1.25.0," *Numpy.org*, 2023. Accessed: Feb. 07, 2024. [Online]. Available: <https://numpy.org/>.
- [29] "Raspberry pi 4 model b datasheet, raspberry pi (trading) ltd.," *United Kingdom (UK)*. Accessed: Feb. 07, 2024. [Online]. Available: <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf>.
- [30] A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*. Sebastopol, CA, USA: O'Reilly, 2019.
- [31] A. Shrestha, H. Li, J. Le Kernec, and F. Fioranelli, "Continuous human activity classification from fmcw radar with bi-lstm networks," *IEEE Sens. J.*, vol. 20, no. 22, pp. 13607–13619, Nov. 2020, doi: 10.1109/JSEN.2020.3006386.
- [32] scikit-learn, "Sklearn.preprocessing.robustscaler." Accessed: Sep. 05, 2023. [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html>
- [33] B. W. Yap, K. A. Rani, H. A. A. Rahman, S. Fong, Z. Khairudin, and N. N. Abdullah, "An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets," 2014, pp. 13–22. doi: 10.1007/978-981-4585-18-7\_2.
- [34] D. C. Montgomery and G. C. Runger, *Applied statistics and probability for engineers*, 7th ed. John Wiley and Sons, 2020.