2025 IEEE International Conference on E-health Networking, Application & Services (Healthcom) | 979-8-3315-0989-7/25/$31.00 ©2025 IEEE | DOI: 10.1109/HealthCom60686.2025.11342661

Markov Transition Field for Fall Detection using Time-Series Data

Rebeca B. Kalbermatter*
CeDRI, SusTEC
Instituto Politécnico de Bragança and
Universidade de Trás-os-Montes e Alto Douro
Bragança, Portugal
*Corresponding author: kalbermatter@ipb.pt

Felipe G. Silva
CeDRI, SusTEC
Instituto Politécnico de Bragança
Bragança, Portugal
gimenez@ipb.pt

Ana Isabel Pereira
CeDRI, SusTEC
Instituto Politécnico de Bragança
Bragança, Portugal
apereira@ipb.pt

António Valente
INESC-TEC
Universidade de Trás-os-Montes e Alto Douro
Vila Real, Portugal
avalente@utad.pt

José Lima
CeDRI, SusTEC
Instituto Politécnico de Bragança
Bragança, Portugal
jllima@ipb.pt

Réda Yahiaoui
SINERGIES (UR 4662)
Université Marie-Louis Pasteur
Besançon, France
reda.yahiaoui@univ-fcomte.fr

Moustafa Fayad
SINERGIES (UR 4662)
Université Marie-Louis Pasteur
Besançon, France
moustafa.fayad@univ-fcomte.fr

***Abstract*—Fall detection systems have traditionally relied on sequential pattern recognition methods, using, for example, time series data obtained from inertial sensors, such as accelerometers. This paper proposes a methodology for fall detection based on converting time series from accelerometer sensors into visual representations using the Markov Transition Field (MTF) method. The UP-Fall dataset was used to test the performance of a Convolutional Neural Network (CNN) model trained on the MTF images generated. A systematic analysis of the image generation parameters was carried out, including the window size, the percentage of overlap, and the number of bins used in the discretizations. The experiments showed that the configuration with 55 bins, a window of 200 samples, and 40% overlap resulted in the best accuracy (97.13%), demonstrating that the conversion of sensory signals into MTF images is a promising alternative for fall detection, allowing computer vision models to capture relevant temporal patterns with high efficiency.**

***Index Terms*—fall detection, Markov Transition Field, Convolutional Neural Network, time series segmentation, sensor data**

I. Introduction

Population ageing is significant in Europe. According to Eurostat, on January 1, 2024, the European Union had 449.3 million inhabitants, 21.6% of whom were aged 65 and over [1]. In France, around 20.5% of the population is aged 65 and over, while in Portugal this proportion reaches 24.1% [1], [2]. This ageing trend results from declining birth rates and increasing life expectancy. According to the OECD, projections indicate that in Europe the proportion of people aged 65 and over will rise from 21% in 2023 to 29% by 2050 [3]. These statistics reflect major challenges for health and social systems, as the burden of expenses related to elderly people continues to grow.

Advanced age is the main risk factor for falls [4]. Increasing frailty (muscle strength, balance, vision, and cognitive disorders) significantly raises the likelihood of falling. In the EU-28, there were approximately 1.44 million annual hospitalizations for falls among the elderly [5]. For example, in France, the estimated annual number of hospitalizations for falls among those aged 65 and over exceeds 100,000, with more than 10,000 deaths. According to the EVITA system in Portugal, there were 40,842 emergency department visits for falls in 2023 among those aged 65 and over [6], [7].

To address the serious health and economic consequences of falls, the scientific community has been actively developing intelligent systems for fall detection and/or prediction. These systems mainly rely on recent advances in smart sensors and artificial intelligence. They are divided into two broad categories: (a) single-sensor systems and (b) multi-sensor systems [8]. Furthermore, fall detection sensors can be classified into two main types: (a) wearable sensors (accelerometers, gyroscopes, etc.) and (b) external sensors (microphones, cameras, etc.) [9]. The accelerometer is one of the most widely used wearable sensors, integrated into many fall detection devices. It measures body movement accelerations along three orthogonal axes. Although single-sensor systems offer high accuracy for fall detection, their low power consumption, particularly that of accelerometer-based devices, makes them especially suitable for wearable applications [8], [10].

Accelerometer data are collected as time series, enabling

the detection of sudden acceleration changes that characterize falls. Temporal features are essential to distinguish falls from normal activities, although some activities may have similar acceleration amplitudes but different temporal patterns. Moreover, the representation of input data is an important factor influencing deep learning performance [11]. In this paper, we explore the encoding of temporal dynamics into images using the Markov Transition Field method, as well as the use of Convolutional Neural Networks (CNNs) for classification and anomaly detection [12]. The MTF preserves temporal and dynamic information in a two-dimensional image format by encoding the quantized transition probabilities between the states of the time series. This allows CNN models to capture complex temporal dependencies that are often overlooked by traditional feature extraction methods.

The rest of the article is organized as follows: Section II describes the related works regarding time series to MTF image encoding. The Section III presents the methodology and outlines the experimental details. Section IV discusses the outcomes of the experiments, and Section V concludes and presents the proposed future works.

II. Related Work

Time series data is a series of data points aligned in temporal sequence, and has a fundamental role in feature-based motion identification, in general, and fall detection, in particular. By analyzing time series data measured by sensors, anomalies can be detected early and responded to promptly. However, data collected in real environments can contain noise, compromising the detection of anomalous patterns. An image-based deep learning model can more accurately learn various patterns from time series data, like using the Markov Transition Field (MTF) [13] to encode time-series signals as images.

Yang et al. [14] proposed the human activity recognition using deep convolutional neural networks to automatically learn hierarchical features from raw multichannel inertial sensor data. The CNN is learned in a joint optimization of feature extraction and classification in a supervised manner. Benchmarks show that on datasets like the Opportunity Activity Recognition Challenge (OARC) that this architecture is better than standard methods (e.g., SVM, kNN) by around 5–10%.

Hatami et al. [15] explored the time-series-to-image encoding, transforming sensor signals into recurrence plot images, which are then classified using deep CNNs. The method captures temporal patterns in 2D form, with superior performance to traditional classifiers. Though recurrence plots rather than MTF, the study provides crucial evidence that image-based embedding has the potential to significantly improve time-series classification.

Kim et al. [16] introduce a fault detection framework that applies Gaussian noise augmentation and MTF image conversion from time-series data of physical sensors. The data transformed via the MTF feed the PatchCore anomaly detection model. The approach achieves an accuracy of 0.9843 and an F1-score of 0.9898. The system is highly accurate with less data size and training time compared to raw time-series models.

Xu et al. [17] propose a human activity recognition that transforms multivariate inertial data into Gramian Angular Fields (GAF) [18] images and classifies them using CNNs. The method using GAF and Multi-dilated kernel residual (Mdk-ResNet) achieved an accuracy of 97%, reinforcing the effectiveness of image-based encoding in capturing dynamic sensor patterns.

These proposed works show the effectiveness of transforming time-series data in images to enable CNNs to learn more profound spatiotemporal features than traditional signal-based or sequence models. While these approaches have been documented with promising outcomes in more generic activity recognition contexts, their application to fall detection is scarce, particularly for MTF. The proposed approach here addresses this research gap through the systematic examination of the use of MTF image encoding on accelerometer signals for fall detection, through the research of the influence of significant transformation parameters (e.g., bin number, window length and overlap), and the fusion of the images into a CNN-based framework.

III. Methodology

This section will describe the proposed methodology for fall detection using time series and data conversion through Markov Transition Fields (MTF). The proposed methodology follows a pipeline made up of four main processes: (1) pre-processing the sensor data, (2) segmenting the time series, (3) transforming the segmented windows into images using the Markov Transition Field (MTF), and (4) classification using a Convolutional Neural Network (CNN). Fig. 1 represents the flow diagram of the proposed pipeline.

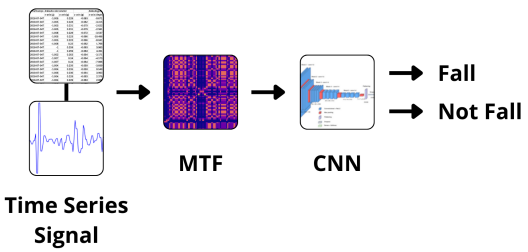


Fig. 1: The proposed pipeline.

A. UP-Fall Dataset and Data Preparation

The dataset UP-Fall [19] is a multimodal sensor dataset for fall detection. It includes 11 activities and 3 trials per activity, performed by 17 volunteers without impairment. The data is collected from three different wearable sensors and cameras. The complete dataset contains a total of 812 GB and includes temporal series data and images collected in the Faculty of Engineering, Universidad Panamericana, Mexico City, Mexico. The activities performed are related to five falls, with a duration of 10 seconds each action, and six daily

activities, with a duration of 60 seconds each. They used five wearable sensors of MbientLab MetaSensor, which collect raw data from the 3-axis accelerometer with a sampling rate of 100Hz in different parts of the body. For this proposed work, the focus will be only on the accelerometer data from the waist, composed of x, y and z-axis data of the accelerometer. The final dataset contains 294,679 rows, with the columns of timestamp, accelerometer axis (x, y and z), activity label and subject identification.

B. Time Series Segmentation

The multivariate time series used represents measurements from the accelerometer. Since the samples are the x, y and z-axis of the accelerometer, we need to perform a Signal Vector Magnitude (SVM) of the 3-axis value, represented by (1). This transformation of the three variates simplifies the data input for Markov transformation matrices by unifying the variation into a scalar vector.

$$\text{SVM} = \sqrt{x^2 + y^2 + z^2} \tag{1}$$

To enable the extraction of local patterns, the data is segmented into sliding windows with partial overlap. The configuration of these windows is controlled by two main parameters: (i) win_size, which defines the number of samples per segment, and (ii) $step\_percent$, which determines the displacement between consecutive windows. The overlap is calculated according to (2). Each extracted segment is treated as an independent instance to be classified.

$$overlap = win\_size - (win\_size \times \frac{step\_percent}{100}) \tag{2}$$

C. Image Generation through Markov Transition Field

The Markov Transition Field (MTF) method consists of performing image transformation based on the Markov transition probability of time-series data. The MTF process encodes temporal correlation by projecting state transition probabilities onto two two-dimensional matrices, which are subsequently represented as an image. The transformation process consists of the following steps:

1) Quantization: The data is discretized in a fixed number of levels (num_bins), using amplitude quantization. This discreatization will directly affect the next step.

2) State Transition Matrix (STM): The STM is determined by the frequency of the transitions between the bins. The matrix $P_{(i,j)}$ is constructed based on the number of transitions of state *i* to *j*, i.e. $N_{i \to j}$, represented by (3). The transition matrix will have the size of $num\_bins \times num\_bins$.

$$P_{(i,j)} = \frac{N_{i \to j}}{\sum_k N_{i \to k}} \tag{3}$$

3) MTF Encoding: The MTF maps the transition probabilities across time, represented by (4). The temporal dependencies are assigned for each pair of time indices (t_1 and t_2) a transition probability between their respective quantized states. Specifically, the matrix $MTF_{(t_1,t_2)}$ represents the probability of transitioning from state s_{t_1} at time t_1 to state s_{t_2} at time t_2. As a final step, the MTF will be an image without axes, with an image size of $win\_size \times win\_size$. An example of the MTF image is presented in Fig. 2. The total number of images that will be generated will depend directly on the combination of the win_size and the $step\_percent$, defined by (5), where N_{images} will be the total of the images generated and $N_{samples}$ is the total number of dataset samples, which is, in this case, 294,679 samples of the UP-Fall Dataset. These images will compose the dataset that will train the CNN. For clarity, each new dataset is saved using the convention "(num_bins)_(window_size)_(step_percent)".

$$MTF_{(t_1,t_2)} = P(s_{t_1}, s_{t_2}) \tag{4}$$

$$N_{images} = \left| \frac{N_{samples} - win\_size}{step\_percent} \right| + 1 \tag{5}$$

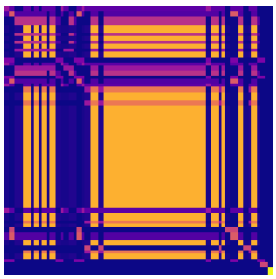


Fig. 2: Example of MTF image.

D. Convolutional Neural Network

For the task of classifying binary fall or no fall, a Convolutional Neural Network (CNN) was designed with a focus on the extraction of spatial patterns from the images generated from the MTF. The proposed architecture consists of two convolutional layers interleaved with max pooling operations, followed by fully connected layers, as described in Table I. The CNN was trained using the Adam optimizer, with a standard learning rate (0.001) and the binary cross-entropy loss function. The main metric used for evaluation was the accuracy, F1-score, and loss.

Since each dataset has a different image size, the CNN performed a detection of the image size from the first image of the path, adjusted the image input to this size, which varied according to the *win_size* parameter, and normalized it to the interval [0,1]. All images were converted to grayscale, resulting in tensors with a single channel in the format (64,64,1).

The training was conducted with a batch size of 32 and 50 epochs. The dataset was balanced using the undersampling

method and divided into 70% for the train and 30% for the test.

To ensure the statistical robustness of the results, the CNN training and evaluation were repeated 10 times for each dataset. In order to guarantee a more reliable comparison between different configurations and ensure the stability in the results, the mean and standard deviation of the metrics were calculated to minimize the effects of variability due to the random initialization of the weights and the splitting of the data.

TABLE I: CNN Structure.

Layer	Type	Parameters	Output Size
1	Conv2D	16 filters, kernel 3×3, ReLU activation	$62 \times 62 \times 16$
2	MaxPooling2D	pool size 2×2	$31 \times 31 \times 16$
3	Conv2D	32 filters, kernel 3×3, ReLU activation	$29 \times 29 \times 32$
4	MaxPooling2D	pool size 2×2	$14 \times 14 \times 32$
5	Flatten	-	6272
6	Dropout	30%	6272
7	Dense	64 units, ReLU activation	64
8	Dense (output)	1 unit, Sigmoid activation	1

E. Experimentation of the Hyperparameter

To explore the different combinations of the parameters (*win_size, step_percent* and *num_bins*), a search strategy was implemented. Based on the fact that the dataset contains fall experiments with 100 to 300 records, the window size was set to 100 to 300 records, in steps of 50. The number of bins was chosen according to the amount of information to be preserved, and was set to 5 to 100, increasing in steps of 5. And the window overlap was varied between 30% and 70%, in steps of 10%. For each parameter combination, a new image dataset was generated, and for each dataset, the training and testing process occurred 10 times, reshuffling the samples into training (70%) and testing (30%) for each run. The results are reported as averages, thus providing a more reliable and fair comparison between models. The target was to find the best values between the limits proposed, presented in Table II, that achieve the highest accuracy with the lowest standard deviation.

TABLE II: Experimental combinations

num_bins	win_size	step_percent	Nº of datasets
5, 10, 15, 20, 40, 45, 50, 55, 90, 95, 100	100, 150, 200, 250, 300	30, 40, 50, 60, 70	275

IV. Results

The experiments were conducted in multiple executions across different combinations of the experimental parameters. For each combination, a new dataset of MTF images was generated and used to train the CNN model. To account for variability in the training process, each configuration was executed ten times. The mean and standard deviation of the performance metrics over these ten runs were then computed to ensure the robustness and reliability of the results. In Table III, the best 10 results of the experiments are presented, where accuracy, F1-score, and loss are the mean of the runs for each combination.

TABLE III: The 10 best results of the experimentation.

Path*	Acc.	standard deviation	F1-score	Loss
50_300_50	0.9781	0.0124	0.9780	0.0863
100_300_50	0.9781	0.0152	0.9784	0.1147
50_250_70	0.9778	0.0104	0.9779	0.1123
90_300_40	0.9777	0.0123	0.9778	0.1663
45_300_50	0.9776	0.0065	0.9777	0.1385
95_300_50	0.9776	0.0127	0.9778	0.1296
90_300_30	0.9774	0.0062	0.9776	0.1460
40_300_30	0.9771	0.0060	0.9773	0.1501
90_300_50	0.9770	0.0076	0.9772	0.1269
55_300_40	0.9764	0.0083	0.9764	0.1554
...	...	...	...	...

*path = (num_bins)_(window_size)_(step_percent)

An important result is observed in Fig. 3, which represents the plots of the win_size over the time stamp. When using win_size of 250 or 300, it is possible to notice that two different activities can appear within the same window. This happens because of the way the dataset is organized, where in some cases the same activity is repeated multiple times in sequence. While this reflects the structure of the dataset, it does not necessarily represent real-world scenarios, where the person are unlikely to perform the same activity repeatedly.

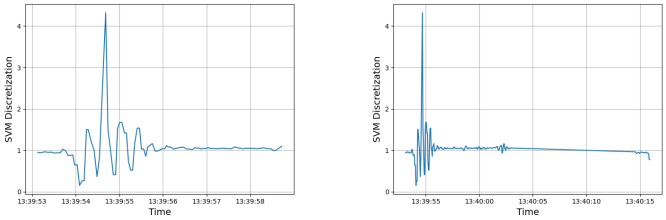
Therefore, these results show a limitation of using a larger window size with short-duration activities, such as falls. Since the duration of a fall is shorter than 250 or 300 samples, these larger windows may also include parts of preceding or subsequent activities.

As a consequence, the state matrix will be directly influenced to have an increase in the number of transitions. Informally, when using a larger window, multiple fall events may be represented in an MTF image. In this scenario, the CNN could learn to identify these unrealistic patterns, but the model will not generalize to real-world scenarios, where falls typically occur only once per episode.

To compare the best accuracy focusing on the window size, Table IV presents the combination with the highest accuracy and lowest standard deviation for each window.

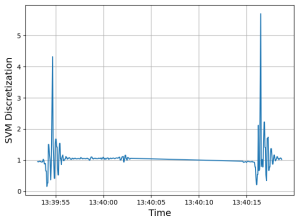
While the combination 50_300_50 achieved the highest mean accuracy (0.9781), this gain is very marginal compared to the combination 55_200_40 (0.9714, a difference of 0.64%), which has the lowest standard deviation. The increased deviation of the first path may undermine reliability, particularly in practical deployments where consistency is critical. Furthermore, when analyzing the F1-score and Loss, for the first two best accuracies, 55_200_40 stays with a lower deviation than the first one.

Also, the choice of the combination 55_200_40 results in an advantage regarding the computational requirements due to

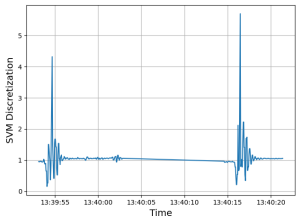


(a) 100 sample

(b) 200 sample



(c) 250 sample



(d) 300 sample

Fig. 3: Time stamp by SVM samples.

TABLE IV: Best accuracy for each proposed window size in Table II.

path*	acc_mean**	F1_mean**	loss_mean**
50_300_50	0.9781 ± 0.0124	0.9780 ± 0.0126	0.0863 ± 0.0540
50_250_70	0.9778 ± 0.0104	0.9779 ± 0.0103	0.1123 ± 0.0818
55_200_40	**0.9714 ± 0.0079**	**0.9719 ± 0.0076**	**0.1846 ± 0.0600**
50_150_60	0.9628 ± 0.0114	0.9634 ± 0.0110	0.2807 ± 0.1210
40_100_70	0.9175 ± 0.0180	0.9182 ± 0.0177	0.4758 ± 0.0942

*path = (num_bins)_(window_size)_(step_percent)
**± standard_deviation

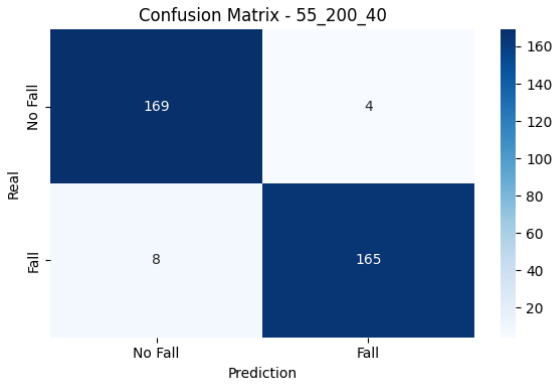
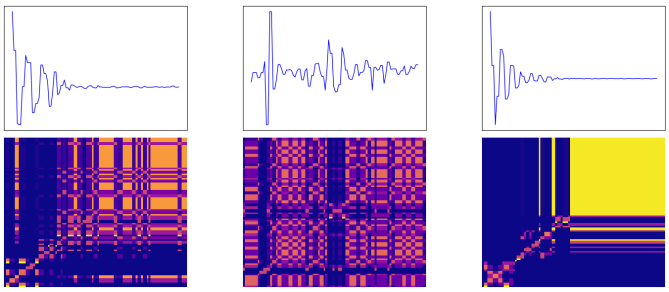


Fig. 4: Confusion Matrix of the best accuracy.



(a) (b) (c)

Fig. 5: Application of MTF image generation for the combination 28_100_70. (a) Falling sitting in an empty chair, (b) Standing, (c) Falling forward using hands.

the result of the input image size (200x200 pixels). Compared to larger window sizes (e.g, 250 or 300), the smaller input dimension significantly reduces computational requirements, which means that it enables faster training and inference, lower memory usage, and improved energy efficiency. This characteristic enables the deployment of the model in resource-constrained environments, such as edge devices or embedded systems.

As a final decision, the combination 55_200_40 was selected as the best combination of the parameters, achieving an accuracy of 97.13% (± 0.79%), an F1-score of 97.19%, and a loss of 0.1846. The CNN had a total of 406,337 parameters trainable with this input, and computed the ten runs in 3 minutes and 22 seconds. The CNN was trained on a MacBook Pro with an Apple M3 Pro processor and 18 GB of memory. The confusion matrix is presented in Fig. 4, and shows high sensitivity and specificity, with few false negatives. This behavior is crucial in fall detection applications, where errors of omission can compromise user safety. Fig. 5 shows examples of the images generated from the time series for MTF image coding of this best combination of parameters.

In comparison with other solutions on the same dataset, our proposed methodology achieved highly competitive results by combining time series data with image-based representations for deep learning in fall detection. In the comparative study of topologies for a deep neural network, Galvão et al. [20], using the accelerometer data and a CNN with one of the dimensions for its kernel processing, achieved an accuracy of 97.7% for fall detection. Using a Convolutional Long Short-Term Memory (LSTM), Islam et. al [21] achieved an accuracy of 89.081 ± 0.005. Qi et al. [22] explored the use of multimodal data in a CNN-based deep learning model, and achieved an accuracy for fall detection of $95.88\% \pm 0.138$. Table V summarizes the performance comparisons of our proposed architecture with recent studies on the UP-Fall detection dataset.

V. Conclusion

This work presented a methodology for detecting falls using Markov Transition Field (MTF) representations of time series data and a Convolutional Neural Network (CNN) architecture. By systematically varying the MTF generation parameters, namely the number of bins (num_bins), the size of the sliding window (win_size) and the overlap (step_percent), it was possible to identify a combination that maximizes the binary classification performance of fall events. Among the configurations evaluated, 55_200_40 was selected as the optimal path due to its high accuracy (0.9713), the lowest standard deviation (0.0079), and consistent performance. Although 50_250_70 and 50_300_50 achieved a slightly higher accuracy (0.9781

TABLE V: Performance comparison of the results obtained in the recent studies on UP-Fall detection dataset.

Reference (year)	Data Source	Architecture	Accuracy	F1-score
[20] (2021)	Time-Series	CNN1D	97.700 ± 0.479	-
[21] (2023)	Time-series	ConvLSTM	89.081 ± 0.005	89.278 ± 0.005
[22] (2023)	Time-Series	GAF + 3-layer CNN	95.880 ± 0.138	95.570 ± 0.147
Ours (2025)	**Time-Series**	**MTF + CNN-based**	**97.138 ± 0.790**	**97.193 ± 0.760**

and 0.9778, respectively), the marginal gain did not outweigh the increased variability. The results suggest that visual coding of sensor data can effectively support fall detection tasks and highlight the importance of fine-tuning the discretization parameters in image time series transformations.

Different directions can be explored to improve the robustness of the approach and allow its applicability in real scenarios. With regard to transforming time series into images, it is proposed to compare the Markov Transition Field (MTF) with other relevant techniques, such as Gramian Angular Field (GAF), Recurrence Plot (RP) and topological persistence images. These alternatives can produce more discriminating representations, especially in adverse acquisition conditions.

Another relevant path is the use of alternative or hybrid models. Deeper CNN architectures, hybrid CNN+LSTM models or even transformers adapted for temporal data can capture more complex patterns.

In addition, test the same parameter's configuration with different datasets, to refine even more the combination, joining this to a detailed analysis of false positives and negatives, could guide fine adjustments to the pipeline or justify the adoption of a second verification stage, using fuzzy logic or additional classifiers.

ACKNOWLEDGMENT

This work was supported by National funds: UID/05757 - Research Centre in Digitalization and Intelligent Robotics (CeDRI); and SusTEC, LA/P/0007/2020 (DOI: 10.54499/LA/P/0007/2020) and the Junior Professor Chair (CPJ) of Franche Comte University, Galaxie number 4718 ANR-23-CPJ1-0010-01. Rebeca B. Kalbermatter is supported by the European Union under Horizon Europe (Project 101078933 - STEP - STEM and Equality, Diversity and Inclusion: an open dialogue for research enhancement in Portugal). Any related publications reflect only the views of the authors.

REFERENCES

[1] EUROSTAT, "Population structure and ageing," February 2025. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Population_structure_and_ageing (accessed 2025-07-05).

[2] INSEE, "Population par âge." https://www.insee.fr/fr/statistiques/4277619, February 2020. https://www.insee.fr/fr/statistiques/4277619 (accessed 2025-07-05).

[3] OECD and European Commission, *Health at a Glance: Europe 2024: State of Health in the EU Cycle*. Paris: OECD, Nov. 2024. https://doi.org/10.1787/b3704e14-en.

[4] A. F. Ambrose, G. Paul, and J. M. Hausdorff, "Risk factors for falls among older adults: a review of the literature," *Maturitas*, vol. 75, no. 1, pp. 51–61, 2013.

[5] S. Turner, R. Kisser, and W. Rogmans, "Falls among older adults in the eu-28: Key facts from the available statistics," *EuroSafe, Amsterdam*, pp. 1–5, 2015. https://eupha.org/repository/sections/ipsp/Factsheet_falls_in_older_adults_in_EU.pdf (accessed 2025-07-05).

[6] M. des Solidarités et des Familles, "Plan antichute des personnes âgées," August 2024. https://solidarites.gouv.fr/plan-antichute-des-personnes-agees (accessed 2025-07-05).

[7] T. Alves, S. Silva, P. Braz, C. Aniceto, R. Mexia, and C. M. Dias, "Quedas em pessoas idosas em portugal: uma abordagem epidemiológica a partir dos dados de 2023 do sistema evita," *Boletim Epidemiológico Observações*, vol. 13, no. 35, pp. 91–96, 2024.

[8] S. Nooruddin, M. M. Islam, F. A. Sharna, H. Alhetari, and M. N. Kabir, "Sensor-based fall detection systems: a review," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 5, pp. 2735–2751, 2022.

[9] M. Fayad, M.-Y. Hachani, K. Ghoumid, A. Mostefaoui, S. Chouali, F. Picaud, G. Herlem, I. Lajoie, and R. Yahiaoui, "Fall detection approaches for monitoring elderly healthcare using kinect technology: A survey," *Applied Sciences*, vol. 13, no. 18, p. 10352, 2023.

[10] A. Purwar and I. Chawla, "A systematic review on fall detection systems for elderly healthcare," *Multimedia Tools and Applications*, vol. 83, no. 14, pp. 43277–43302, 2024.

[11] B. Jokanovic, M. G. Amin, and F. Ahmad, "Effect of data representations on deep learning in fall detection," in *2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pp. 1–5, IEEE, 2016.

[12] Z. Wang and T. Oates, "Imaging time-series to improve classification and imputation," *arXiv preprint arXiv:1506.00327*, 2015.

[13] J. Yan, J. Kan, and H. Luo, "Rolling bearing fault diagnosis based on markov transition field and residual network," *Sensors*, vol. 22, no. 10, 2022.

[14] J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, p. 3995–4001, AAAI Press, 2015.

[15] N. Hatami, Y. Gavet, and J. Debayle, "Classification of time-series images using deep convolutional neural networks," in *Tenth International Conference on Machine Vision (ICMV 2017)* (A. Verikas, P. Radeva, D. Nikolaev, and J. Zhou, eds.), vol. 10696, p. 106960Y, International Society for Optics and Photonics, SPIE, 2018.

[16] G.-I. Kim, H. Yoo, H.-J. Cho, and K. Chung, "Defect detection model using time series data augmentation and transformation," *Computers, Materials & Continua*, vol. 78, no. 2, pp. 1713–1730, 2024.

[17] H. Xu, J. Li, H. Yuan, Q. Liu, S. Fan, T. Li, and X. Sun, "Human activity recognition based on gramian angular field and deep convolutional neural network," *IEEE Access*, vol. 8, pp. 199393–199405, 2020.

[18] Z. Wang and T. Oates, "Imaging time-series to improve classification and imputation," in *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, p. 3939–3945, AAAI Press, 2015.

[19] L. Martínez-Villaseñor, H. Ponce, J. Brieva, E. Moya-Albor, J. Núñez-Martínez, and C. Peñafort-Asturiano, "Up-fall detection dataset: A multimodal approach," *Sensors*, vol. 19, no. 9, 2019.

[20] Y. M. Galvão, J. Ferreira, V. A. Albuquerque, P. Barros, and B. J. Fernandes, "A multimodal approach using deep learning for fall detection," *Expert Systems with Applications*, vol. 168, p. 114226, 2021.

[21] M. M. Islam, S. Nooruddin, F. Karray, and G. Muhammad, "Multi-level feature fusion for multimodal human activity recognition in internet of healthcare things," *Information Fusion*, vol. 94, pp. 17–31, 2023.

[22] P. Qi, D. Chiaro, and F. Piccialli, "Fl-fd: Federated learning-based fall detection with multimodal data fusion," *Information Fusion*, vol. 99, p. 101890, 2023.