Detecting the Pre-impact of Falls in the Elderly, Along with the Use of an Airbag Belt for Protection Against Femoral Neck Fractures

Mohand O. Seddar[1,3](✉), Guillaume Rao[2](✉), Anthony Fleury[1](✉), and Maurice Kahn[3](✉)

[1] IMT Nord Europe, CERI SN, Lille, France
{anthony.fleury,mohand.seddar}@imt-nord-europe.fr
[2] Aix Marseille Univ, CNRS, ISM, Marseille, France
guillaume.rao@univ-amu.fr
[3] INDIENOV, Marseille, France
{mseddar,mkahn}@indienov.com

Abstract. Falls are a significant health risk for older adults, and fall-related injuries are a leading cause of morbidity and mortality in this population. Elderlies are particularly vulnerable to falls due to age-related declines in mobility, balance, and muscle strength, as well as chronic medical conditions with use of certain medications. These injuries can range from minor bruises and scrapes to more severe like fractures, head trauma, or internal bleeding. To prevent falls in older adults, some solutions propose to ensure a safe living environment, others to maintain physical activity, and others to manage chronic medical conditions. This article presents the implementation and test of a system preventing hip fractures resulting from falls using a fall detection and prediction system designed to protect and alert individuals during falls.

Keywords: Pre-impact fall detection · fall detection · machine learning · tinyML · medical devices · IMU · e-health · airbag system

1 Introduction

Falls among the elderly are currently a real human and financial scourge. 25 to 30% of people over 65 and 50% of those over 80 fall at least once a year. These are the first cause of emergency visits for people over 75 and the second cause of death by unintentional trauma. A significant proportion of falls result in femoral neck fractures, a real public health issue (Katsoulis, 2017 [1] Oberlin, 2016 [2]). Furthermore, research has demonstrated that the apprehension of falling is linked to unfavorable outcomes, including reduced engagement in everyday activities, decreased level of physical activity, increased likelihood of falling, symptoms of depression, and a lower overall quality of life [3]. According to INSERM (2017) [4], between 50,000 and 80,000 individuals in France, and approximately 1.5 million around the world (Silver Eco 2017 portal) experience fractures of the

© The Author(s) 2023
K. Jongbae et al. (Eds.): ICOST 2023, LNCS 14237, pp. 117–129, 2023.
https://doi.org/10.1007/978-3-031-43950-6_10

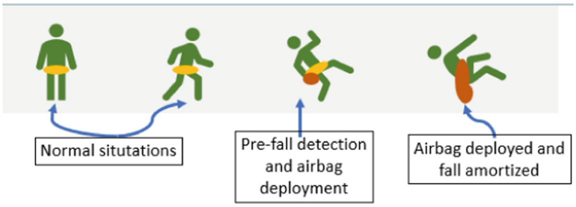

Normal situtations
Pre-fall detection and airbag deployment
Airbag deployed and fall amortized

Fig. 1. Belt Airbag illustration

neck of the femur. This type of fracture has severe implications as one-quarter of people pass away within one year, and 50% of those who survive experience a considerable loss of autonomy. This work is part of a project led by Indienov, a company that created a smart airbag belt capable of analyzing the movements of individuals and detecting anomalies (in this particular case falls and pre-falls), and instantly activate a perfectly positioned airbag prevent fractures of the femoral neck or greater trochanter (Fig. 1). The belt will also be able to detect critical situations such as lying for a prolonged time.

2 Literature Review

A large variety of fall detection systems are available and rely on diverse sensors (cameras, infrared motion sensors, LIDAR, and inertial motion units). These fall detection systems are generally classified into two categories: fixed systems (in the environment) and mobile/wearable systems (necklaces, watches, belts, phones, etc.) [5]. We focused on mobile systems using Inertial Motion Units.

2.1 State of the Art

Initial research concerning fall detection systems employed threshold-based algorithms, and demonstrated that by applying thresholds to vertical and horizontal velocities, it is possible to differentiate between daily activities and falls [6,7] with remarkable accuracy. In a study based on a sensor located on the chest [20], a fall detection algorithm is optimized using the Upper Fall Threshold (UFT) and Lower Fall Threshold (LFT) of the accelerometer combined with the UFT of the gyroscope for fall detection. The experimental results showed that the algorithm was able to detect falls compared to other daily movements with a sensitivity and specificity of 96.3% and 96.2%, respectively. The addition of gyroscope data significantly improves specificity over published results based solely on accelerometer data, as changes in angular velocity provide an additional indication of a fall event compared to other activities that may also experience high acceleration spikes. The use of LFTs and UFTs is equivalent to determining the window of fall and detecting it by means of its start and end.

Alternatively, Li et al. [8] used static postures and dynamic transitions between them to detect falls, defining a fall as a static "lying" posture with an unintentional transition to it. Numerous other studies have examined fall detection, including reviews, learning algorithms with automatic feature extraction, etc. [9–18], some of which have been found to be highly reliable [19].

Detecting the onset of a fall, or pre-fall detection, presents a challenging problem. Unlike fall detection, which relies on identifying the impact of a fall visible on acceleration signals, pre-fall detection requires detecting the early signs of an impending fall. As a result, accelerometer and gyroscope signals may not yet exhibit features indicative of a fall. The primary hurdle in solving this issue is to distinguish the crucial warning signs that trigger a fall while minimizing the number of inaccurate detections (false positive cases).

In the context of pre-fall detection, one of the earliest fall protection systems was proposed by G. Shi et al. [21] and later enhanced [22]. It relies on an inertial motion unit and airbag deployment, along with a Support Vector Machine (SVM) prediction model and an embedded Digital Signal Processing (DSP) unit.

In recent years, machine learning models have gained popularity due to their good performances in fall and pre-fall detection. In a comparative study conducted by Yu et al. [24], a hybrid ConvLSTM model was evaluated and compared to other existing models. The results showed that the ConvLSTM model outperformed other models for all three classes (no drop, drop before impact, and drop), with average sensitivities of 93.15%, 93.78%, and 96.00% for no drop, drop before impact, and drop, respectively. Furthermore, the model demonstrated higher specificities for all three classes (96.59%, 94.49%, and 98.69%) compared to the LSTM and CNN models. These results suggest that the proposed hybrid model outperforms the LSTM and CNN models, providing high detection accuracy (particularly for the pre-impact drop). The combination of convolution and recurrent models allows feature extraction and temporal linkage between the data, leading to satisfactory results compared to using these models separately.

Recent fall prediction methods have shown very promising results (as shown in Table 1). However, one important issue highlighted in the literature is the need to compare the performance of fall detection algorithms using real-world data,

Table 1. Fall prediction algorithms comparison

Fall prediction algorithms overview							
Réf	Sensor	Placement	Sampling [Hz]	Methods	FALL/ADL	Subjects	The best results
[23] L. Tong et al. (2013)	A	trunk	100	HMM	2/5	8	Sn : 100% Sp : 100% Tl : 200–400 [ms]
[25] Yang et al. (2013)	A, G	back-trunk	20	NN	N/A	5	Sn : 92.26 % Sp : 70.02 % Tl : 400 [ms]
[26] Saadeh et al. (2019)	A		256	LSVM, DT, KNN, NLSVM		MobiFall	Sn : 97.8% Sp : 99.1% Tl : 300–700[ms]
[24] Yu et al. (2020)	A	/	200	CNN, LSTM ConvLstm		SisFall	Sn : 93.78% Sp : 94.00% Tl : / [ms]
[27] Yu et al. (2021)	A, G, M	back-waist	100	Threshold, SVM, ConvLstm	21/15	32	Sn : 99.32% Sp : 99.01% Tl : 403 ± 163 [ms]

Table 2. Comparison of the Kfall model performances on simulated data and real data. Source : [27]

Algorithm Evaluation on real word data							
Réf	Algorithm	Data	FN (false negative)	FP (false positive)	Sensitivity (%)	Specificity (%)	Lead time ([ms])
[27] Yu et al. (2021)	ConvLSTM	KFall dataset	3/444	5/507	99.32	99.01	403 ± 163
		Farseeing dataset	1/15	4/15	93.33	73.33	411 ± 317

as significant discrepancies have been observed between the results obtained from simulated falls data of healthy individuals and real falls data from at-risk populations [11]. The Bourke and Chen algorithms perform the best in terms of sensitivity and specificity, but all algorithms exhibit lower performance on real falls compared to simulated falls. The latter being also true for the work from Yu et al. (as shown in Table 2) despite the very high performance obtained on simulation data (99.32%, 99.03% of sensitivity, and specificity, respectively). This difference in performance was mainly attributed to the fact that the movements in real falls do not exhibit the same signatures as simulated falls, especially in terms of the amplitude before impact [28].

3 Methods and Material

3.1 Data Collection

As mentioned above, the problem of motion and fall data for at-risk individuals currently remains one of the biggest challenges. The first step consists of obtaining real, reliable motion data that comes from our target audience. Then, in addition to retrieving widely used movement data from the internet (especially in research, e.g. SisFall [30], MobiFall [29], kFall [27] etc.), a specific protocol based on [29,30] was developed to gather experimental data (Table 3 and 4).

A data collection campaign has been initiated in EHPAD (institutions for dependent elderly people) (Dataset_03, Dataset_04), in which multiple volunteers were equipped with dataloggers (embedded sensors with IMU + SD card - attached to a belt worn at the waist) to monitor their activities throughout the day (and in some cases, the entire week from morning to evening). The purpose of the database obtained from this data collection is to develop solutions and algorithms adapted to the target population, the elderly, where detecting pre-fall events is of utmost importance. As the participants' profiles are highly suitable

Table 3. Public Data Overview

Public Dataset						
Dataset	Labelled	Types ADLs/Falls	Number of participants	Age	Weight [KG]	gender (M/F)
MobiFall	Yes	9/4	24	22–47	/	17/7
SisFall	Yes	19/15	23 15	19–30 60–75	41–102	11/12 8/7
kFall	Yes	21/15	32	24 ± 3.7	69.3 ± 9.5	32/

Table 4. Acquired Data Overview

Recolted Data						
Dataset	Labelled	Types ADLs/Falls	Number of participants	Age	Weight [KG]	gender (M/F)
Dataset_01	Yes	11/8	1	47	/	H
Dataset_02	Yes	47/24	11	22–52	50–120	4/7
Dataset_03	Yes	13/0	∼ 60	>63	47–120	∼ 38/22
Dataset_04	No	/	> 100	>63	47–130	>40/>60

for this purpose, the collected data is expected to provide valuable insights and aid in validating the developed solutions and algorithms.

3.2 Configuration and Setup

After examining the SisFall and MobiFall datasets, we discovered that they were not optimal for predicting falls because they lacked temporal information on the pre-fall phase. Even though there has been a lot of research on fall detection, there hasn't been much done on prefall detection. The key contrast between prefall detection and fall detection is found here. Prefall detection focuses on detecting the beginning of a fall, which happens over a brief period of time and then passes. In other words, throughout the fall, we can only see the beginning of the fall during a specified time period. On the other side, fall detection focuses on identifying a fall when it really occurs, generally after the impact has taken place. At any time after the fall, it is relatively simple to identify because the impact of the fall clearly leaves an imprint in the signals. Therefore, we had to identify and label the pre-fall phase, which may have introduced bias into the labeling process. In contrast, the kfall dataset contained more detailed information on the onset and impact of falls, making it more useful for fall prediction. Additionally, the age and profiles of the participants in the various datasets were not very representative, so we generated new data for activities of daily living and falls that included a wide range of realistic scenarios.

For the labeled data, participants followed a predefined protocol that included activities from the SisFall dataset as well as additional daily activities such as walking, using stairs with a cane or walker, doing household chores, and dancing. We also included wheelchair activities. Unlabeled data were collected from participants engaged in their normal activities throughout the day, resulting in data that better reflects real-world scenarios. For data recording, we used dataloggers, which are developed by the same company, and we recorded the videos of these simulations. The IMU (MPU6050) that we used was equipped with a 3-axis accelerometer and gyroscope, with a resolution of ±8 [g] for the accelerometer and ±1000°/s for the gyroscope. The sampling frequency (200 [Hz]) is defined based on the data analysis and signal processing.

3.3 Data Processing and Feature Extraction

Data Cleaning and Labelling: To reduce accelerometer sensitivity noise, we used a 3rd order Butterworth lowpass filter with a cutoff frequency of 20 [Hz] as first step. We then divided the fall data into four stages (Fig. 2): non-risk, pre-fall, the impact, and the post-fall. This categorization was based on notes taken during fall simulations (such as the type of fall, subject, and duration) and recorded video data (including the start time of the fall, the end time of the pre-fall phase, the time of ground contact, and the time of impact).

Data Analysis and Feature Selection: An initial visual inspection was performed to analyze the collected data. This enabled the identification of movement-specific signatures and variable behavior, and the tracking of their fluctuations over time in relation to distinct activities and subject profiles (as shown in Fig. 3). A statistical examination of the information obtained from various activities was performed with the aim of identifying indicators that can

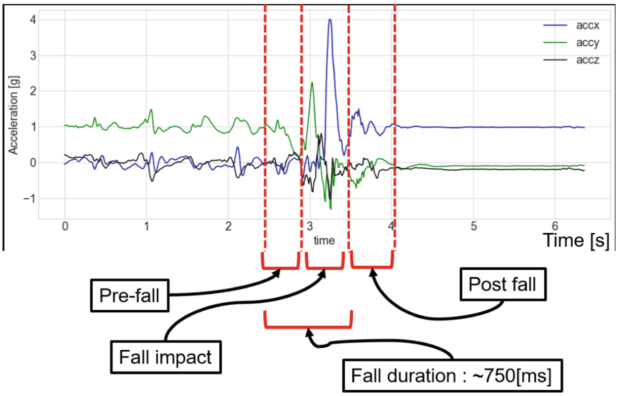


Fig. 2. Fall Data Segmentation.

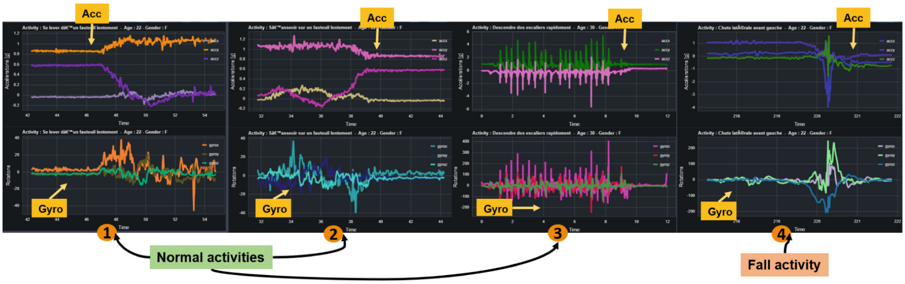
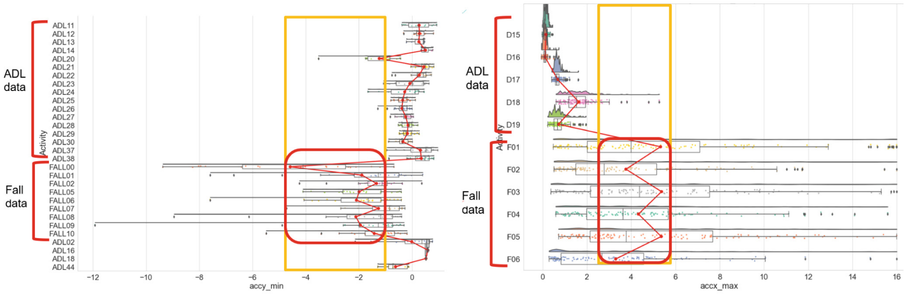


Fig. 3. Activities signal examples - Acc: accelerometer (acc_x, acc_y, acc_z), Gyro: gyroscope (gyro_x, gyro_y, gyro_z) - 3 Normal Activities (1, 2, 3) vs one Fall Activity (4)



(a) acc_y_min: the minimum values of acceleration over the y-axis in different activities

(b) acc_x_max: the maximum values of acceleration over the x-axis in different activities

Fig. 4. Analysis of the data distribution examples - compare the data distribution over different activities (ADLs vs FALLs)

Table 5. Selected Variables for Feature calculation

Selected Variables for Features calculation	
Variable	description
Acc_x	The acceleration data over the x-axis
Acc_y	The acceleration data over the y-axis
Acc_z	The acceleration data over the z-axis
Gyro_x	The angular velocity data over the x-axis
Gyro_y	The angular velocity data over the y-axis
Gyro_z	The angular velocity data over the z-axis
Acc_svm(signal vector magnitude)	The magnitude of the acceleration vector $\mathrm{Acc\_svm} = \sqrt{\mathrm{acc\_x}^2 + \mathrm{acc\_y}^2 + \mathrm{acc\_z}^2}. \quad (1)$
Gyro_svm	The square root of the sum of the angular velocities squared $\mathrm{Gyro\_svm} = \sqrt{\mathrm{gyro\_x}^2 + \mathrm{gyro\_y}^2 + \mathrm{gyro\_z}^2}. \quad (2)$
SMA(signal magnitude area)	The measure of the magnitude of a varying acceleration $\frac{1}{\mathrm{T}} \int_0^T (\lvert acc\_x(t) - \mu_{acc\_x} \rvert + \lvert acc\_y(t) - \mu_{acc\_y} \rvert + \lvert acc\_z(t) - \mu_{acc\_z} \rvert)\, dt. \quad (3)$
Tilt	The vertical tilt of the body $\mathrm{tilt} = \arctan(\frac{\mathrm{acc\_y}}{\sqrt{\mathrm{acc\_x}^2 + \mathrm{acc\_z}^2}}). \quad (4)$

distinguish between ADLs and falls (Fig. 4). Once this stage was completed, we selected the variables that would be used in computing the features (listed in Table 5). Subsequently, statistical computations were carried out on each variable, resulting in a collection of potential features for pre-fall detection (Table 6). The features were calculated in sliding windows of 0.5[s].

Algorithm and Features: For feature selection, we started an optimization process over all the features (cited on Table 6), based on recursive elimination of features by KNN (K Nearest Neighbors) and RF (Random Forest) algorithms, so we tested several combinations depending on the type of falls we were proposing and the performances we were getting. Ultimately, we chose an algorithm based on the RF (random forest) and the most important features, which we selected

Table 6. Selected Features for Calculation

Selected Features for calculation	
Feature	description
mean	the mean value of the current window
max	The max value of the current window
min	The min value of the current window
std(Standard Deviation)	The standard deviation value of the current window
mean_start	the mean of the m first values ([0; m]) of the current window
mean_median	the mean of the m values around the median ([median_index - $(\frac{m}{2})$; median_index + $(\frac{m}{2})$]) of the current window.
mean_q1	the mean of the m values around the 1st quartile of the current window.
mean_q3	the mean of the m values around the 3rd quartile of the current window.
diff_m1_m2	the difference between each two calculated means, cited above
skewness	measures the asymmetry of the distribution of the current window $\text{skewness} = \frac{\sum_1^m (x_i - \mu)^3}{m\sigma^3} \quad (5)$
kurtosis	measures the “tailedness” of the probability distribution of the current window $\text{kurtosis} = \frac{\sum_1^m (x_i - \mu)^4}{m\sigma^4} \quad (6)$

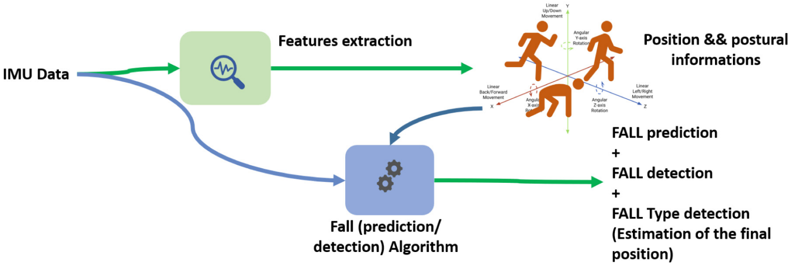

IMU Data
Features extraction
Linear Up/Down Movement
Y
Angular Y-axis Rotation
Linear Back/Forward Movement
X
Angular X-axis Rotation
Angular Z-axis Rotation
Linear Left/Right Movement
Z
Position && postural informations
FALL prediction
+
FALL detection
+
FALL Type detection (Estimation of the final position)
Fall (prediction/ detection) Algorithm

Fig. 5. Fall Detection/Prediction System.

for their relevance to decision making. Our solution detects pre-falls and falls (Fig. 5), then, sends an alert to relatives and caregivers (previously selected).

3.4 Model Deployment in Embedded Architecture

Due to their inherent resource constraints, embedded systems are unable to support sophisticated inference models. As a result, implementing AI models in these systems becomes difficult, especially in real-time scenarios when characteristics like response speed, space complexity, and computing complexity are critical. Focusing on low complexity choices is necessary since these limits impose extra factors to take into account when choosing the right model. Deep learning techniques, for instance, may not be practical due to their inference, time, and space complexity, necessitating careful attention throughout implementation. As a result, it may be necessary to develop customized methods to optimize model training, conversion, and implementation for better optimization and utilization of hardware resources such as DSP and FPU. Various methods (weight pruning, quantization, encoding, etc.) and solutions (TensorFlow Lite Micro, EdgeML, STM32Cube.AI, etc.) are available to address this challenge [31], but they require careful consideration and planning to achieve the desired results in the context of TinyML.

4 Experiments and Evaluation

4.1 Experiments

Initially, we tested our solution on daily life and fall acquired data to ensure its functionality (this was the Experiment_00). Our ultimate goal was to develop a functional and customized solution for the airbag belt. Therefore, we implemented and deployed the solution in the embedded system (belt) and conducted three types of experiments.

- **Experiment_00:** In this experiment, we have tested our solution on the acquired data (which contains ADLs and Falls).
- **Experiment_01:** The first test involved wearing the belt and performing guided activities (ADLs and falls).
- **Experiment_02:** The second test involved wearing the belt throughout the day and engaging in their daily activities.
- **Experiment_03:** Finally, we tested the belt on elderly people in EHPAD for a whole day and evaluated the performance recorded in the log files.

The first and second tests were conducted by young and healthy individuals.

4.2 Evaluation

We established a set of evaluation criteria for our solutions, which serve to measure their quality and degree of reliability. These criteria include:

. **Sensitivity:** measures the solution's ability to detect and predict falls accurately. Sensitivity $= \frac{\mathrm{TP}}{\mathrm{TP+FN}}$, TP : true positive, FN : false negative. (7)

. **Specificity:** measures its ability to correctly identify activities of daily living (ADLs). Specificity $= \frac{\mathrm{TN}}{\mathrm{TN+FP}}$, TN : true negative, FP : false positive. (8)

. **Time Lead:** which refers to the time between the prediction of a fall and the actual impact of the fall. Our objective is to detect pre-fall movements at least **220 [ms]** before impact (safe deployment of airbags).

Finally, we use performance as an overall measure of how well our solution performs based on the aforementioned criteria. This value is calculated using a weighted sum of sensitivity, specificity, and time lead. Based on the desired outcomes, it is possible to set the weights of the parameters, such as emphasizing sensitivity, specificity, or lead time.

$\mathrm{Performances} = (\mathrm{P_{sn}} * \mathrm{Sensivity}) + (\mathrm{P_{sp}} * \mathrm{Specifity}) + (\mathrm{P_{tm}} * \mathrm{time\_lead})$. (9)
$\mathrm{Performances} \in [0, 1]$, P: weights, $\mathrm{P_i} \in [0, 1]$, $\sum \mathrm{P_i} = 1$, time_lead normalized

Table 7. Experimental results

Experimental results					
/	PDS (pre-fall detection system)			FDS (fall detection system)	
/	Sensitivity(%)	Specificity(%)	Lead time[ms]	Sensitivity(%)	Specificity(%)
Experiment_00	99	98	∼ 250	99	99
Experiment_01	88	92	∼ 210	98	97
Experiment_02	/	87	/	/	97
Experiment_03	/	93	/	/	97

5 Experimental Result Analysis

The Dataset_02 was used for training, and subsequent optimization techniques were implemented to select a subset of the data that would yield lightweight and tailored models optimized for embedded systems. To assess the effectiveness of our solution, we employed different methods depending on the experiment. For Exp. 0 and Exp. 1, we evaluated the performance based on the number of simulations (each activity labeled as ADL or Fall), enabling us to determine false positive and false negative rates. For other experiments (exp. 2 and exp. 3), we based our evaluations on the number of activity hours and participant-reported information. If the participant doesn't fall but a fall was predicted or detected, it is considered a false positive. If the participant falls, but we didn't detect anything, it is considered a false negative. However, if the person falls and we predict it, it was considered as a true positive.

Generally, falls in real life are infrequent, so our solution specificity is highlighted in exp. 2 and exp. 3. When we encounter false positives or false negatives, we analyze the data and incorporate it into subsequent training sessions. The benefit of these experiments is that we can evaluate and validate our solution under real-world conditions with participants who are natural and free, unlike other evaluation tests. Therefore, our solution is exposed to situations (activities) not seen during training, representing a challenge for us. The evaluation results that we currently have are presented in (Table 7). It should be noted that the evaluation of our solution is still ongoing, with particular focus on (Exp. 03).

6 Discussion

Based on the initial results from experiments (exp. 00 and exp. 01), it appears that fall detection is facilitated by the fact that signals already contain the necessary information to accurately detect falls. This is supported by the high level of sensitivity and specificity observed. The fall prediction models showed a decrease in performance when the environment and subjects were changed (between exp. 00 and exp. 01), which is likely attributed to the prior compression of the models before deployment on embedded devices. This compression was intended to reduce the models space-time complexity and make them more lightweight. The two additional experiments exposed our solution to real-world conditions, which allowed participants greater freedom of movement, resulting in signals that differed, sometimes markedly, from what we had observed in our learning database. Moreover, as the belt-wearing was subtle, participants behaved more naturally, revealing movements we had not encountered during training.

7 Conclusion

This study involved the development of a fall detection system and a fall prediction system, which were adapted and deployed on an airbag belt to test their effectiveness in real-life conditions. While the theoretical results were comparable to other works, the goal was to create a functional system that performed well in both theory and practice. This is in contrast to some studies that relied on theoretical results or private motion data to which the researchers did not have access. However, the lack of movement data from elderly people and real fall data posed significant challenges. Therefore, the plan for the future is to launch a collection campaign in EHPAD to gather natural data over a long period of time, including real falls that can be added to the learning base.

References

1. Katsoulis, M., Benetou, V., Karapetyan, T., et al.: Excess mortality after hip fracture in elderly persons from Europe and the USA: the CHANCES project. J. Intern. Med. **281**(3), 300–310 (2017). PMID: 28093824. https://doi.org/10.1111/joim.12586
2. Oberlin, P.: Quel risque de décès un an après une fracture du col du fémur?, Etudes et Résultats, direction de la recherche, des études, de l'évaluation et des statistiques, Janvier 2016
3. Scheffer, A.C., Schuurmans, M.J., van Dijk, N., et al.: Fear of falling: measurement strategy, prevalence, risk factors and consequences among older persons. Age Ageing. **37**(1), 19–24 (2008). PMID: 18194967. https://doi.org/10.1093/ageing/afm169
4. https://www.inserm.fr/dossier/osteoporose/
5. Rajagopalan, R., Litvan, I., Jung, T.-P.: Fall prediction and prevention systems: recent trends, challenges, and future research directions. Sensors **17**, 2509 (2017). https://doi.org/10.3390/s17112509
6. Wu, G.: Distinguishing fall activities from normal activities by velocity characteristics. J Biomech. **33**(11), 1497–500 (2000). PMID: 10940409. https://doi.org/10.1016/s0021-9290(00)00117-2
7. Bourke, A.K., O'Donovan, K.J., OLaighin, G.M.: Distinguishing falls from normal ADL using vertical velocity profiles. In: 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 3176–3179 (2007). PMID: 18002670. https://doi.org/10.1109/IEMBS.2007.4353004
8. Li, Q., Stankovic, J.A., Hanson, M.A., et al.: Accurate, Fast fall detection using gyroscopes and accelerometer-derived posture information. In: 2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks, Berkeley, CA, USA, pp. 138-143 (2009). https://doi.org/10.1109/BSN.2009.46

9. Noury N., Fleury, A., Rumeau, P., et al.: Fall detection-principles and methods. In: Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 1663–1666 (2007). PMID: 18002293. https://doi.org/10.1109/IEMBS.2007.4352627
10. Noury, N., Rumeau, P., Bourke, A.K., et al.: A proposal for the classification and evaluation of fall detectors. In: IRBM, vol. 29, no 6, pp. 340–349 (2008). ISSN 1959–0318, https://doi.org/10.1016/j.irbm.2008.08.002
11. Bagalà, F., Becker, C., Cappello, A., et al.: Evaluation of accelerometer-based fall detection algorithms on real-world falls. PLoS ONE **7**(5), e37062 (2012). Epub 2012 May 16. PMID: 22615890; PMCID: PMC3353905. https://doi.org/10.1371/journal.pone.0037062
12. Casilari, E., Álvarez-Marco, M., García-Lagos, F.: A study of the use of gyroscope measurements in wearable fall detection systems. Symmetry **12**, 649 (2020). https://doi.org/10.3390/sym12040649
13. Xu, T., Se, H., Liu, J.: A fusion fall detection algorithm combining threshold-based method and convolutional neural network. Microprocess. Microsyst. **82**, 103828 (2021). ISSN 0141-9331, https://doi.org/10.1016/j.micpro.2021.103828
14. Zurbuchen, N., Wilde, A., Bruegger, P.: A machine learning multi-class approach for fall detection systems based on wearable sensors with a study on sampling rates selection. Sensors (Basel). **21**(3), 938 (2021). PMID: 33573347; PMCID: PMC7866865. https://doi.org/10.3390/s21030938
15. Liu, K.-C., et al.: Domain-adaptive fall detection using deep adversarial training. IEEE Trans. Neural Syst. Rehab. Eng. **29**, 1243–1251 (2021). https://doi.org/10.1109/TNSRE.2021.3089685
16. Collado-Villaverde, A., Cobos, M., Muñoz, P., Barrero, F.D.: A simulator to support machine learning-based wearable fall detection systems. Electronics **9**, 1831 (2020). https://doi.org/10.3390/electronics9111831
17. Liu, K.-C., Hung, K.-H., Hsieh, C.-Y., et al.: Deep-learning-based signal enhancement of low-resolution accelerometer for fall detection systems. IEEE Trans. Cogn. Dev. Syst. **14**(3), 1270–1281 (2022). https://doi.org/10.1109/TCDS.2021.3116228
18. Šeketa, G., Pavlaković, L., Džaja, D., Lacković, I., Magjarević, R.: Event-centered data segmentation in accelerometer-based fall detection algorithms. Sensors **21**, 4335 (2021). https://doi.org/10.3390/s21134335
19. Liu, K.-C., Hsieh, C.-Y., Hsu, S.J.-P., Chan, C.-T.: Impact of sampling rate on wearable-based fall detection systems based on machine learning models. IEEE Sens. J. **18**(23), 9882–9890 (2018). https://doi.org/10.1109/JSEN.2018.2872835
20. Huynh, Q.T., Nguyen, U.D., Irazabal, L.B., Ghassemian, N., Tran, B.Q.: Optimization of an accelerometer and gyroscope-based fall detection algorithm. J. Sens. **2015** (2015). https://doi.org/10.1155/2015/452078
21. Shi, G., et al.: Development of a human airbag system for fall protection using MEMS motion sensing technology. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, pp. 4405–4410 (2006). https://doi.org/10.1109/IROS.2006.282019
22. Shi, G., Chan, C.S., Li, W.J., et al.: Mobile human airbag system for fall protection using MEMS sensors and embedded SVM classifier. IEEE Sens. J. **9**(5), 495–503 (2009). https://doi.org/10.1109/JSEN.2008.2012212
23. Tong, L., Song, Q., Ge, Y., Liu, M.: HMM-based human fall detection and prediction method using tri-axial accelerometer. IEEE Sens. J. **13**(5), 1849–1856 (2013). https://doi.org/10.1109/JSEN.2013.2245231

24. Yu, X., Qiu, H., Xiong, S.: A novel hybrid deep neural network to predict pre-impact fall for older people based on wearable inertial sensors. Front. Bioeng. Biotechnol. **8**, 63 (2020). PMID: 32117941; PMCID: PMC7028683. https://doi.org/10.3389/fbioe.2020.00063
25. Yang, S.-H., Zhang, W., Wang, Y., Tomizuka, M.: Fall-prediction algorithm using a neural network for safety enhancement of elderly. In: 2013 CACS International Automatic Control Conference (CACS), Nantou, Taiwan, pp. 245-249 (2013). https://doi.org/10.1109/CACS.2013.6734140
26. Saadeh, W., Butt, S.A., Altaf, M.A.B.: A patient-specific single sensor IoT-based wearable fall prediction and detection system. IEEE Trans. Neural Syst. Rehabil. Eng. **27**(5), 995–1003 (2019). https://doi.org/10.1109/TNSRE.2019.2911602
27. Yu, X., Jang, J., Xiong, S.: A large-scale open motion dataset (KFall) and benchmark algorithms for detecting pre-impact fall of the elderly using wearable inertial sensors. Front Aging Neurosci. **13**, 692865 (2021). PMID: 34335231; PMCID: PMC8322729. https://doi.org/10.3389/fnagi.2021.692865
28. Klenk, J., Becker, C., Lieken, F., et al.: Comparison of acceleration signals of simulated and real-world backward falls. Med Eng. Phys. **33**(3), 368–373 (2011). Epub 2010 Nov 30. PMID: 21123104. https://doi.org/10.1016/j.medengphy.2010.11.003
29. Vavoulas, G., Pediaditis, M., Chatzaki, C., et al.: The MobiFall dataset: fall detection and classification with a smartphone. Int. J. Monit. Surveillance Technol. Res. **2**, 44–56 (2014)
30. Sucerquia, A., López, J.D., Vargas-Bonilla, J.F.: SisFall: A fall and movement dataset. Sensors (Basel). **17**(1), 198 (2017). PMID: 28117691; PMCID: PMC5298771. https://doi.org/10.3390/s17010198
31. Ray, P.P.: A review on TinyML: state-of-the-art and prospects. J. King Saud Univ.-Comput. Inf. Sci. **34**(4), 1595–1623 (2022)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.