A comparison of public datasets for acceleration-based fall detection

Raul Igual[1], Carlos Medrano and Inmaculada Plaza

R&D&I EduQTech Group, Escuela Universitaria Politecnica de Teruel, University of Zaragoza, Ciudad Escolar s/n, 44003 Teruel, Spain. E-mail: {rigual, ctmedra, inmap}@unizar.es

Abstract: Falls are one of the leading causes of mortality among the older population, being the rapid detection of a fall a key factor to mitigate its main adverse health consequences. In this context, several authors have conducted studies on acceleration-based fall detection using external accelerometers or smartphones. The published detection rates are diverse, sometimes close to a perfect detector. This divergence may be explained by the difficulties in comparing different fall detection studies in a fair play since each study uses its own dataset obtained under different conditions. In this regard, several datasets have been made publicly available recently. This paper presents a comparison, to the best of our knowledge for the first time, of these public fall detection datasets in order to determine whether they have an influence on the declared performances. Using two different detection algorithms, the study shows that the performances of the fall detection techniques are affected, to a greater or lesser extent, by the specific datasets used to validate them. We have also found large differences in the generalization capability of a fall detector depending on the dataset used for training. In fact, the performance decreases dramatically when the algorithms are tested on a dataset different from the one used for training. Other characteristics of the datasets like the number of training samples also have an influence on the performance while algorithms seem less sensitive to the sampling frequency or the acceleration range.

Keywords: Fall detection, accelerometers, public datasets, comparison, data analysis

1- Introduction

Fall incidents are a major public health problem among the older adults. Falls and the subsequent long lie period are associated with severe adverse health consequences [1-3]. The *Centers for Disease Control and Prevention* [4] quantify the direct medical cost of falls among older adults over US$30 billion per year in the United States. Every 17 seconds an older adult is treated in a hospital emergency department for injuries related to a fall [5]. In this context, there is a need for robust fall detectors that trigger an alert when a fall is detected [6-9].

Several techniques for fall detection have been investigated. Igual et al. [6] classified the existing fall detection studies into 2 categories: context-aware systems [10] and acceleration-based wearable devices [11]. One of the characteristics of acceleration-based studies is that they report high detection rates. For example, sensitivity and specificity are reported respectively as 97.5% and 100% by Kangas et al. [12], 94.6% and 100% by Bourke et al. [13],

[1] Corresponding author. Phone number: (+34) 978 64 53 62. E-mail: rigual@unizar.es

98.6% and 99.6% by Yuwono et al. [14]; and 100% and 100% by Abbate et al. [15]. Other fall detection studies provide similar performances [16,17]. It should be noted that the detection rates provided by all these studies are very high. However, many authors on this field have noticed strong difficulties when comparing different acceleration-based studies [6,18]. This is due to the fact that each study uses its own dataset composed of simulated falls and ADL. Therefore, it is not clear whether the declared results are influenced by the specific dataset used and it is not possible to perform a fair comparison since the datasets used to provide a measure of the detection performances are different in each study.

In this regard, several authors have identified the need for having public datasets [19,20]. Some efforts have been performed in this direction since several datasets were made publicly available in the recent years: *DLR* [21] published in 2011, *MobiFall* [22] available in 2013 and *tFall* [20] uploaded in 2014 (the study of Fudickar et al. [23] cites another public dataset but it seems that it cannot be downloaded currently). Although these three datasets can be freely accessed, there is no study focused on comparing them. Therefore, some important questions are still without response: Can the public datasets be used indistinctly?, Are there any differences among them?, Is the performance of the fall detection algorithms affected by the specific selected dataset?

In this regard, the general goal of this paper is to compare in a fair play the existing public datasets (figure 1). For that purpose, the following specific objectives are stated:

1) To check whether or not the performance of a given algorithm depends on the selected dataset.
2) To compare the generalization capability of the public datasets. Generalization capability refers to the ability of a system trained under some conditions to work under different conditions.
3) To determine whether some of the datasets' parameters affect the performance of the fall detection algorithms.

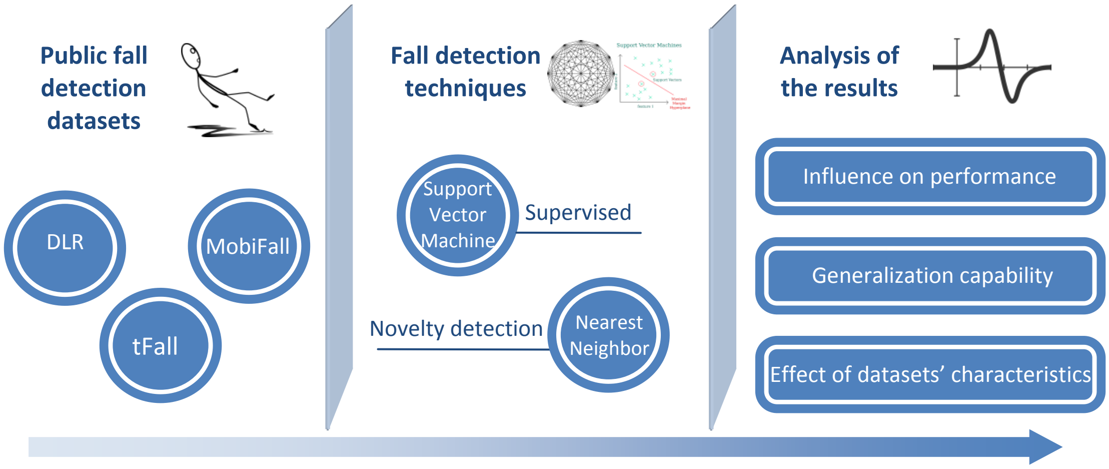

***Figure 1** – General schema of the study.*

2- Materials and methods

2.1 Datasets

As a result of an extensive literature search, we could identify three public datasets presenting acceleration samples of falls and ADL: *DLR* [24], *MobiFall* [25] and *tFall* [26]. These three datasets were collected by different research institutions, each conducting the experiments in a particular fashion. These datasets were selected since, to the best of our knowledge, they are the only ones that are publicly available to the scientific community.

2.1.1 DLR dataset

This dataset was made publicly available by the *Institute of Communications and Navigation* of the *German Aerospace Center* (*DLR*). The dataset was collected from 16 male and female subjects aged between 23 and 50 and annotated manually by an observer. In total it contains about 4.5 hours of labeled falls and activities (table 1). Each participant performed a different number of ADL and falls. To capture the motion data, the *Xsens MTx* inertial measurement unit with a single tracker placed on the belt was used. The data were sampled at 100 Hz and the measurement unit had an acceleration range of at least 7g.

2.1.2 MobiFall dataset

This dataset was developed by the *Biomedical Informatics & eHealth Laboratory* of the *Technological Educational Institute of Crete*. The *MobiFall* dataset contains data from 11 volunteers: 6 males and 5 females (age range: 22 to 36). Nine participants performed falls and ADLs, while two performed only the falls. On the one hand, each participant performed four types of falls which were repeated 3 times per subject. On the other hand, nine types of ADL were simulated (table 1). Specifically, a Samsung Galaxy S3 device with the *LSM330DLC* inertial module (3D accelerometer and gyroscope) was used to capture the motion data. The device was located in a trouser pocket freely chosen by the subject in any random orientation. The range of the accelerometer was 2g and the data were acquired at 100 Hz.

2.1.3 tFall dataset

This dataset was developed by the *EduQTech (Education, Quality and Technology)* group of the *University of Zaragoza*. Ten people were involved in the data collection process (7 males and 3 females, whose ages ranged from 20 to 42). The simulation set consisted of 8 different types of falls (table 1). Each fall was repeated 3 times per subject. The ADL collection process was carried out under real-life conditions. ADL were recorded in the subjects' real world environment while they performed their daily lives. Each subject was monitored during at least one week. Only ADL over a given threshold (1.5g) were recorded. At the end of the experience, an average number of about 800 records per subject (6 seconds length) were obtained. The data were acquired using Samsung Galaxy Mini phones at 50 Hz and with a range of 2g. In the fall study, participants carried a phone in both their two pockets.

		DLR	MobiFall	tFall
Experiments	No. subjects	16	11	10
	Device	Xsens MTx	Samsung Galaxy S3	Samsung Galaxy Mini
	Position	Belt	Pocket	Pocket
	Types of falls	Not specified	Forward-lying, front-knees-lying, sideward-lying and back-sitting-lying	Forward, forward straight, backward, lateral left and right, sitting on empty air, syncope and forward fall with obstacle
	Types of ADL	Sitting, standing, walking, running, jumping and lying	Standing, walking, jogging, jumping, stairs up, stairs down, sitting on a chair, step in a car and step out a car	Real-life activities
Samples	No. ADL	1077	831	7816
	No. falls	53	132	503
	Sampling frequency	100 Hz	100 Hz	50 Hz
	Acc. range	7g	2g	2g

Table 1 *– Features of the public fall detection datasets.*

2.2 Fall detection algorithms used to compare the datasets

It is clear that the comparison can depend on the algorithm. Therefore, we have selected two algorithms representing different approaches to fall detection.

The first one is the well-known Support Vector Machine (SVM) classifier [27]. By means of the kernel trick, it maps the inputs to another space in which an optimum hyperplane is found separating two classes, falls and ADL in our case. After training, the classification of a new input relies only on a small subset of the training inputs called the support vectors. Thus, SVM builds a sparse model. We have selected the popular kernel based on Radial Basis Functions (RBF). The inputs are time windows of acceleration shape, with the peak in the middle, and sampled at a given frequency. Then, given two acceleration patterns $\boldsymbol{a}(t)$ and $\boldsymbol{b}(t)$, the distance $d(\boldsymbol{a}, \boldsymbol{b})$ between them is obtained as:

$$d^2(\boldsymbol{a}, \boldsymbol{b}) = \int_{t_i}^{t_f} \|\boldsymbol{a}(t) - \boldsymbol{b}(t)\|^2 dt \qquad (1)$$

The kernel between two inputs is the RBF:

$$k(\boldsymbol{a}, \boldsymbol{b}) = e^{-\gamma \cdot d^2(\boldsymbol{a}, \boldsymbol{b})} \qquad (2)$$

After sampling with period T, d^2 is approximated as:

$$d^2(\boldsymbol{a}, \boldsymbol{b}) \cong T \cdot \sum_i \|\boldsymbol{a}(i) - \boldsymbol{b}(i)\|^2 \qquad (3)$$

On the other hand, we have also considered a novelty detector based on a nearest neighbor (NN) rule [28]. In this case, the system only models the normal activities, ADL. Falls are detected as movements that depart from the normal ones. NN is a pure data driven method. Given a set of exemplars $\boldsymbol{a}_n$, the training set, for a new acceleration shape $\boldsymbol{a}$, the distance to the nearest neighbor is:

$$d_{NN} = \min_{n} d(\boldsymbol{a}, \boldsymbol{a}_n) \qquad (4)$$

If d_{NN} is higher than a threshold, θ, the new input is considered a fall since it is very different from the normal movements stored in the exemplar set.

By varying θ, the Receiver Operating Characteristic (ROC) curve is obtained. In this curve, we selected the point that maximized the geometric mean of the sensitivity and specificity, which has been chosen as the figure of merit (see section 2.5). We performed the same operation on SVM, but this time varying the distance to the hyperplane to draw the ROC.

The training of the SVM was performed with the library *Scikit-Learn* [29]. For NN, we implemented our own code in Python.

2.3 Datasets' preprocessing

The datasets have been preprocessed in order to feed the fall detection algorithms with the data in the same format: 6 s time windows, labeled as ADL or falls, with the acceleration peak in the middle. The peak is always higher than 1.5 g. This is the format suitable for the algorithms explained in section 2.2, although actually, only the central portion of width 1 s was used.

The acceleration magnitude was calculated for the datasets. Then, we extracted all the 6 s time windows having a maximum of the acceleration magnitude in the middle. For *DLR*, which includes long timelines with several activities, the window was labeled with the activity tag associated to the peak. No ADL with a peak in the acceleration magnitude lower than 1.5g was considered for further processing.

As a result, we obtained 1077 ADL samples and 53 fall samples for the *DLR* dataset; 831 ADL samples and 132 fall samples for the *MobiFall* dataset and 7816 ADL samples and 503 fall samples for the *tFall* dataset. The most relevant features of each dataset are summed up in table 1. It is worth highlighting that both *MobiFall* and *tFall* were collected using smartphones while *DLR* was recorded with a sensor unit. Since the features of each dataset are different, to perform a fair comparison, a balanced comparison has been also included as explained in the next section.

2.4 Dataset comparison

In order to fulfill the objectives of the study (section 1), different experiments have been performed. In this section, we briefly describe them and the specific objective to which they relate, see table 2.

- *Goal 1*: To check whether or not the performance of a given algorithm depends on the selected dataset.
- *Experiment 1* (section 3.1): Measurement of the algorithms' performance when fed with the different datasets. For a comprehensive comparison, two different experiments have been performed (table 2, experiment 1):

  - *Raw datasets* (section 3.1.1): Firstly, the fall detection algorithms have been trained and validated using the raw datasets as they were recorded (original number of samples, frequency and range).
  - *Tailored datasets* (section 3.1.2): Secondly, for a fair comparison, the datasets have been tailored to compare them under the same conditions regarding the number of samples used for training, the sampling frequency and the acceleration range. All these parameters have been set to the same values (the most restrictive ones among all three datasets). The most restrictive numbers of ADL and fall samples used for training are determined by the *MobiFall* and *DLR* datasets, respectively. The *tFall* dataset was recorded at the lowest frequency (50 Hz). Thus, *DLR* and *MobiFall* were sub-sampled to have also the data at 50 Hz. Additionally, the minimum range (2g) is given by both *tFall* and *MobiFall*. Therefore, the *DLR* dataset, originally at 7g, was saturated to this value.

- *Goal 2*: To compare the generalization capability of the public datasets.
- *Experiment 2* (section 3.2): The algorithms have been tested with a particular dataset and trained with the other two, in two separate processes. In this way, we can examine the generalization capability of the datasets used for training. As in the previous case, two comparisons have been performed (table 2, experiment 2):
  - *Raw datasets* (section 3.2.1): The datasets used for training and validation were the original ones.
  - *Tailored datasets* (section 3.2.2): The number of samples used for training, the frequency and the range of the datasets have been set to the values of those parameters in the most restrictive datasets (similar to the tailored comparison in experiment 1).

- *Goal 3*: To determine whether the datasets' parameters affect the performance of the fall detection algorithms.
- *Experiments 3* (section 3.3): Several experiments have been conducted (table 2, experiment 3) to quantify the effect of varying the sampling frequency (section 3.3.1), the acceleration range (section 3.3.2) and the number of samples used for training (section 3.3.3):
  - *Sampling frequency* (section 3.3.1): The effect on the performance of using a dataset sampled at 100 Hz or 50 Hz has been measured. Both *DLR* and *MobiFall* datasets have been used in the experiment since they were recorded at 100 Hz. These datasets were sub-sampled to have also the data at 50 Hz. Then, the algorithms were trained and validated with both sets and the results compared.

    The effect on *tFall* dataset could not be measured since this dataset was originally recorded at 50 Hz, which is even below the minimum recommended by some authors [30].
  - *Acceleration range* (section 3.3.2): The effect on the performance of using a dataset acquired with an accelerometer with range of 7g has been compared with the same dataset saturated at 2g. The *DLR* dataset has been selected since its range covers the extension of the study. *tFall* and *MobiFall* datasets could not be evaluated since their records were originally acquired at 2g.

- *Number of training samples* (section 3.3.3): The effect on the performance of the algorithms when varying the number of ADL samples used for training (from 50 to 1450) has been quantified. A similar experiment has been performed by varying the number of falls used for training (between 50 and 300 samples). In this last case, only the SVM algorithm was used since the semisupervised NN method does not use falls for training. This experiment has been performed using *tFall* since it is the only dataset that has a number of records high enough to perform both comparisons.

Exp. No.*	**Experiment description**			**Train**	**Validation**
1	**Effect of the datasets on algorithms' performance (section 3.1)**	**Raw datasets (section 3.1.1)**		Raw DLR	Raw DLR
				Raw MobiFall	Raw MobiFall
				Raw tFall	Raw tFall
		Tailored datasets (section 3.1.2)		Tailored DLR	Tailored DLR
				Tailored MobiFall	Tailored MobiFall
				Tailored tFall	Tailored tFall
2	**Dataset generalization capability (section 3.2)**	**Raw datasets (section 3.2.1)**		Raw DLR	Raw tFall
				Raw MobiFall	
				Raw DLR	Raw MobiFall
				Raw tFall	
				Raw MobiFall	Raw DLR
				Raw tFall	
		Tailored datasets (section 3.2.1)		Tailored DLR	Tailored tFall
				Tailored MobiFall	
				Tailored DLR	Tailored MobiFall
				Tailored tFall	
				Tailored MobiFall	Tailored DLR
				Tailored tFall	
3	**Effect of dataset parameters on algorithms' performance (section 3.3)**	**Sampling frequency (section 3.3.1)**	DLR	DLR sampled at 100 Hz	DLR sampled at 100 Hz
				DLR sampled at 50 Hz	DLR sampled at 50 Hz
			MobiFall	MobiFall sampled at 100 Hz	MobiFall sampled at 100 Hz
				MobiFall sampled at 50 Hz	MobiFall sampled at 50 Hz
		Acceleration Range (section 3.3.2)		DLR with maximum acc 2g	DLR with maximum acc 2g
				DLR with maximum acc 7g	DLR with maximum acc 7g
		No. training samples (section 3.3.3)	Training ADL variation	tFall varying the no. of ADL	Remaining falls and ADL in tFall
			Training falls variation	tFall varying the no. of falls	Remaining falls and ADL in tFall

* Number of the experiment

Table 2 *– Dataset comparison*

2.5 Figure of merit for the comparison

As the figure of merit to measure the performances of the algorithms, we have used the geometric mean of the sensitivity (SE) and the specificity (SP), which is calculated with the formula 5. For a perfect detector the geometric mean has a value of 1.

$$\sqrt{SE \cdot SP} \tag{5}$$

This figure of merit is independent of the size of the datasets. This is a convenient property since the ADL and fall sets are clearly unbalanced. In all the experiments, 5 cross-validation has been used when obtaining the algorithm performances, therefore, getting the mean and the associated standard deviation.

For testing the statistical significance of the difference in performance between two different situations, we have estimated a p-value using a one-side t-test for independent samples (section 3.1) and dependent samples (section 3.2). For the particular case of determining the relation between the performance and the number of training samples (section 3.3.3), we have fitted different kinds of functions to the experimental data, obtaining an estimation of the parameters and their standard deviations. All these calculations have been done using the *Scipy* package for *Python* [31].

3- Results

3.1 Effect of different datasets on the performance

This section presents the results of comparing the performances of the algorithms when fed with the different datasets.

3.1.1 Raw datasets

Table 3 presents the geometric means associated with both NN and SVM algorithms when using the raw datasets. Table 4 shows the p-values when comparing the geometric means of the different datasets for the same algorithms.

At the view of these tables, it is possible to appreciate that the SVM fall detector provides similar results for the three datasets, the differences not being statistically significant. However, the NN fall detector presents better performance when tested with the *tFall* or *MobiFall* datasets, while the results provided when tested with the DLR decrease the performance by 6.3% and 6.8% respectively. As shown in table 4, this difference is statistically significant (p-value is lower than 0.01).

	NN		**SVM**	
	GM	**Std**	**GM**	**Std**
DLR	0.8925	0.0279	0.9777	0.0263
MobiFall	0.9576	0.0205	0.9841	0.0206
tFall	0.9528	0.0130	0.9715	0.0113

Table 3 *– Geometric means and associated standard deviations obtained when training and validating the algorithms using the raw datasets.*

	MobiFall-tFall	DLR-tFall	DLR-MobiFall
NN	0.3375	*0.0027*	*0.0018*
SVM	0.1384	0.3229	0.3425

Table 4 *– p-values for the comparisons of the performances of the raw datasets by pairs in both NN and SVM.*

3.1.2 Tailored datasets

When both NN and SMV algorithms are fed with the tailored datasets (equal number of training samples, frequency and range), their performances (table 5) are not as homogeneous as in the previous case. On the one hand, the NN algorithm presents a statistically significant variation in the performance when the DLR dataset is used (table 6). In this case, the performance of the algorithm decreases. Meanwhile, the results provide by *tFall* and *MobiFall* do not present remarkable differences.

On the other hand, the performance of the SVM algorithm presents statically significant differences when the *tFall* dataset is used (p-value lower than 0.05 as shown in table 6). The performance decreases by 2.91 % on average, while *MobiFall* and *DLR* do not present considerable variations.

	NN		SVM	
	GM	Std	GM	Std
DLR	0.8957	0.0362	0.9772	0.0270
MobiFall	0.9555	0.0198	0.9705	0.0213
tFall	0.9462	0.0155	0.9455	0.0172

Table 5 *– Geometric means and associated standard deviations obtained when training and validating the algorithms using the tailored datasets.*

	MobiFall-tFall	DLR-tFall	DLR-MobiFall
NN	0.2170	*0.0159*	*0.0084*
SVM	*0.0388*	*0.0315*	0.3359

Table 6 *– p-values for the comparisons of the performances of the tailored datasets in both NN and SVM.*

3.2 Comparison of the generalization capability

This section presents the results of comparing the generalization capability of the datasets. In this regard, the algorithms are validated with a specific dataset and trained with the other two in separate processes. Section 3.2.1 presents the algorithms' performance when raw datasets are used, while in section 3.2.2 the algorithms are trained and validated with tailored datasets.

3.2.1 Raw datasets

The performances of the algorithms using the raw datasets are shown in table 7, while the p-values associated with the comparisons are represented in table 8. In this case, the performances present great variations depending on the training and validation datasets.

When the NN or SVM fall detectors are trained with *tFall* and *DLR* datasets and validated with *MobiFall*, it is possible to appreciate that the *tFall*-trained algorithm presents better generalization capability since it clearly outperforms the DLR-based one. Similarly, when both *tFall* and *MobiFall* are used for training and *DLR* is used for validating, the *tFall*-trained algorithms present better performance than the *MobiFall* ones. The differences in both cases are statistically significant since the corresponding p –values remain low (less than 0.01).

When *tFall* is used for validation and the NN and SVM algorithms are trained with both *MobiFall* and *DLR*, the *MobiFall*-based detector generalizes better for the SVM algorithm and the reverse situation occurs when the NN performance is examined. Both results are statistically significant according to their corresponding p-values (table 8).

We can see that in all cases the performance provided by the *tFall*-trained algorithm clearly outperforms the rest of the results.

Additionally, the performances of the *MobiFall*-trained algorithms validated with the *tFall* dataset are lower than those obtained when validating them with the *DLR* dataset. The same happens with the *DLR*-trained algorithms: the validation with *tFall* always presents worse performance. This is a symptom that *tFall* is a harder dataset to generalize on.

		NN		SVM	
Validation	**Train**	**GM**	**Std**	**GM**	**Std**
DLR	tFall	0.8435	0.0129	0.8566	0.0142
	MobiFall	0.7791	0.0134	0.6557	0.0560
MobiFall	tFall	0.8135	0.0307	0.8902	0.0147
	DLR	0.7746	0.0259	0.4502	0.1870
tFall	MobiFall	0.6367	0.0126	0.6132	0.0212
	DLR	0.6774	0.0075	0.3968	0.0512

Table 7 *– Geometric means and their standard deviations when validating using a raw dataset different from the one used for training.*

Validation	Train	NN p-value	SVM p-value
DLR	tFall - MobiFall	*0.0006*	*0.0004*
MobiFall	DLR-tFall	*0.0002*	*0.0036*
tFall	MobiFall-DLR	*0.0045*	*0.0006*

Table 8 *– p-values for the comparisons of the performances of the raw datasets in both NN and SVM, when measuring the generalization capability.*

3.2.2 Tailored datasets

When tailored datasets are used to train the algorithm, the generalization capability of the different datasets (table 9) shows a trend similar to that of the previous section. When *DLR* is used for validation and *tFall* and *MobiFall* for training, the *tFall* dataset provides better generalization capability. Similarly, this dataset also generalize better than *DLR* when the algorithms are validated on *MobiFall*. The statistical analysis shows that the results are significant (table 10).

In fact, all comparisons performed are statistically significant (p-values much lower than 0.01). Therefore, the results present great variations depending on the datasets used for training and validation.

		NN		SVM	
Validation	Train	GM	Std	GM	Std
DLR	tFall	0.8373	0.0124	0.8598	0.0168
	MobiFall	0.7112	0.0411	0.6963	0.0770
MobiFall	tFall	0.8132	0.0329	0.8525	0.0222
	DLR	0.7779	0.0241	0.4753	0.1857
tFall	MobiFall	0.6373	0.0103	0.5917	0.0247
	DLR	0.6747	0.0110	0.3973	0.0779

Table 9 *– Geometric means and their standard deviations when validating using a tailored dataset different from the one used for training.*

Validation	Train	NN p-value	SVM p-value
DLR	tFall - MobiFall	*0.0018*	*0.0039*
MobiFall	DLR-tFall	*0.0007*	*0.0065*
tFall	MobiFall-DLR	*0.0002*	*0.0007*

Table 10 *– p-values for the comparisons of the performances of the tailored datasets in both NN and SVM, when measuring the generalization capability.*

3.3 Effect of datasets' parameters on the performance

This section presents the effect on the algorithms' performance of varying the sampling frequency, the acceleration range and the number of samples used for training.

3.3.1 Sampling frequency

The results of training and validating the algorithms using datasets with different sampling frequencies are shown in table 11, while the p-values of the comparisons are presented in table 12. No statistically significant differences are observed between the performance at 50 Hz or at 100 Hz for neither *DLR* nor *MobiFall* since their geometric means range in the same intervals (table 12). In this case, having data samples at 50 Hz does not have an influence on the performance.

	DLR				MobiFall			
	NN		SVM		NN		SVM	
	GM	Std	GM	Std	GM	Std	GM	Std
50 Hz	0.8925	0.0279	0.9777	0.0263	0.9576	0.0205	0.9841	0.0206
100 Hz	0.8910	0.0271	0.9652	0.0258	0.9580	0.0198	0.9854	0.0192

Table 11 *– Geometric means and their standard deviations when using the same datasets sampled at two different frequencies.*

	NN	SVM
DLR p-valor	0.2580	0.1121
MobiFall p-valor	0.2925	0.0910

Table 12 *– p-values for the comparison of the performances of both NN and SVM algorithms, when using the same datasets sampled at 100 Hz and at 50 Hz.*

3.3.2 Accelerometer range

Table 13 represents the performances of the algorithms when using different acceleration ranges (2g and 7g) in the *DLR* dataset. It can be observed that the NN algorithm provides better performance for the wider range. The difference, although moderate, is statistically significant (p-value lower than 0.05).

On the other hand, there is no statistically significant difference between the performance of the SVM algorithms with 2g and 7g ranges (table 14).

	NN		SVM	
	GM	**Std**	**GM**	**Std**
2g	0.8824	0.0365	0.9784	0.0233
7g	0.8925	0.0279	0.9777	0.0263

***Table 13** – Geometric means and their standard deviations when using the DLR dataset saturated at two different maximum acceleration ranges.*

	NN	**SVM**
p-valor	*0.0380*	0.3716

***Table 14** – p-values for the comparison of the performances of the DLR dataset with acceleration ranges of 2g and 7g in both NN and SVM.*

3.3.3 Number of samples

When increasing the number of samples used for training in the *tFall* dataset, the performance of the algorithms improves. Figures 2 and 3 represent the effect on the performance of increasing the number of ADL samples for both algorithms NN and SVM, respectively. The performance shows an initial increase but saturates at some point. Therefore, the results have been fitted to an exponential function (equation 6). From table 15, we can see that both *b* and *c* parameters of the fitted exponential function are clearly positive, which indicates the growing trend. Thus, it is possible to state that the performance of the algorithms increases up to a point when more ADL are used for training.

$$a + b \cdot \left(1 - e^{\frac{-x}{c}}\right) \qquad (6)$$

	a		b		c	
	Mean	Std	Mean	Std	Mean	Std
NN	0.9080	0.0041	0.0405	0.0039	156.5833	24.2746
SVM	0.9348	0.0034	0.0327	0.0033	110.4792	15.9995

***Table 15** – Values of the parameters of the exponential functions (formula 6, figures 2 and 3), which has been fitted to the performance values of both NN and SVM.*

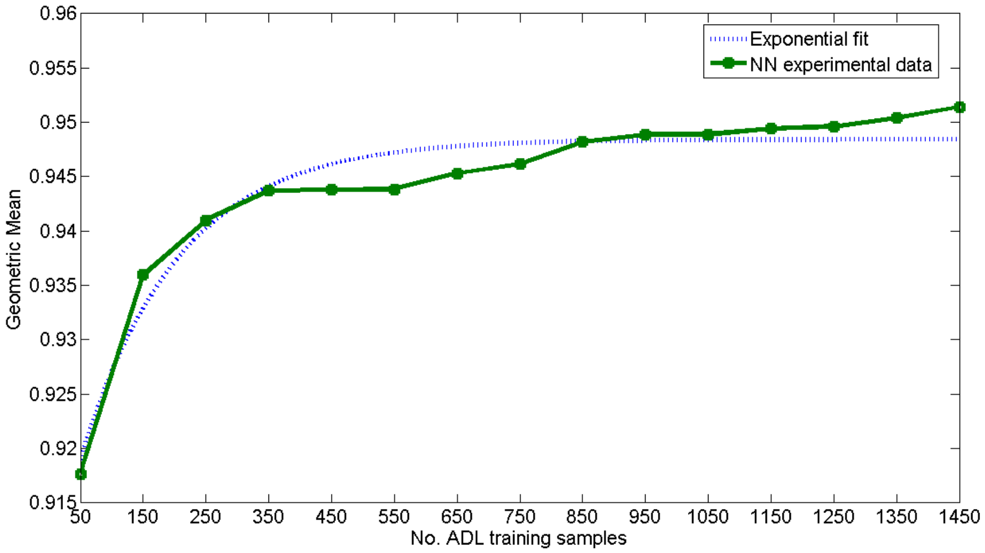

***Figure 2** – Performance of the NN algorithm (solid line) trained with different number of ADL samples. The dashed line represents the exponential function to which the performance values have been fitted.*

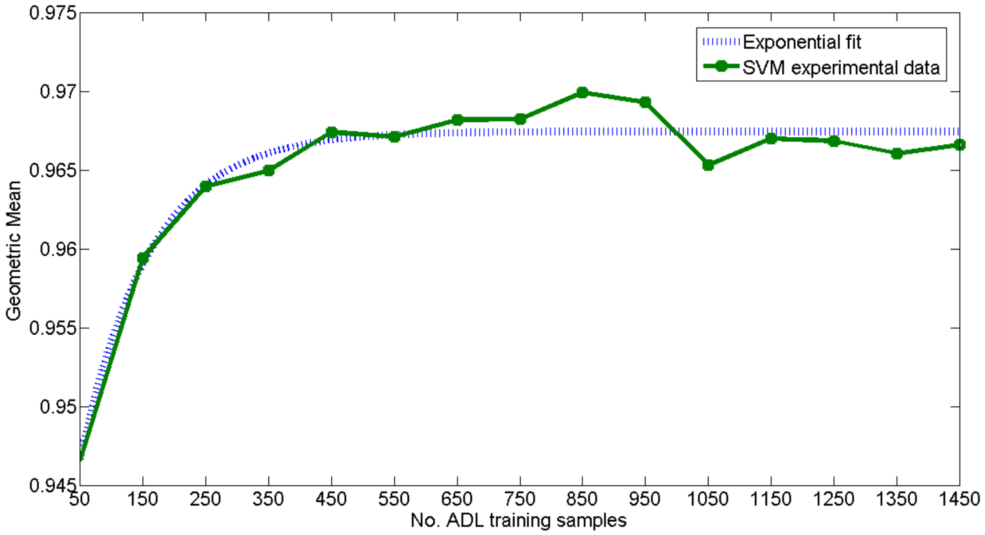

***Figure 3** – Performance of the SVM algorithm (solid line) trained with different number of ADL samples. The dashed line represents the exponential function to which the performance values have been fitted.*

Additionally, when the SVM algorithm is trained using a different number of fall samples in the *tFall* dataset, we can see in figure 4 that the performance follows a growing trend. In this case, the performance values do not show any sign of saturation, so they have been fitted to a linear function (dashed line of figure 4). The values of the parameters of this function are presented in table 16, clearly showing that the algorithm performance improves when the number of training samples increases, roughly 1 % per 100 samples.

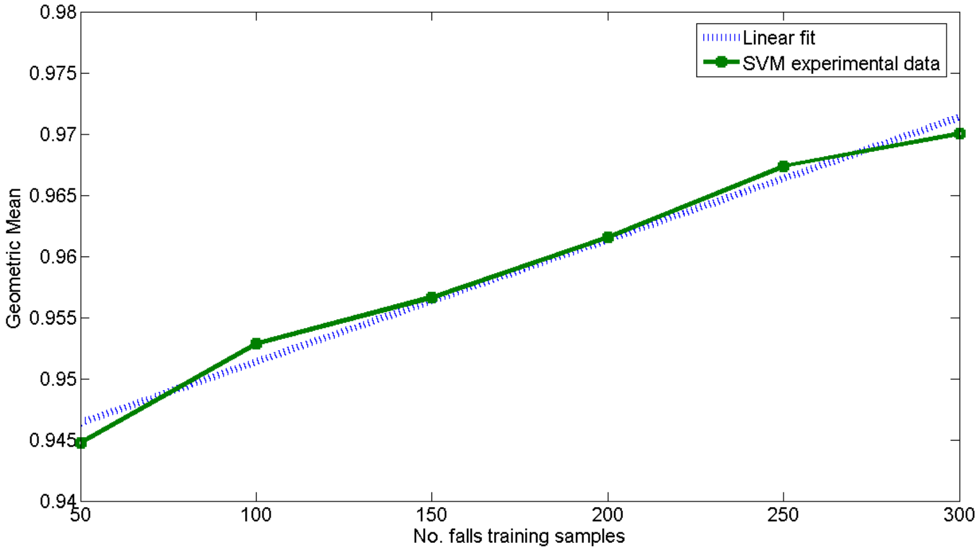

Figure 4 *– Performance of the SVM algorithm (solid line) trained with different number of fall samples. The dashed line represents the linear function to which the performance values have been adjusted.*

	a (slope)		b	
	Mean	Std	Mean	Std
SVM	9.99e-05	9.40e-06	0.9414	0.0018

Table 16 *– Values of the parameters of the linear function* $(ax + \mathrm{b})$*, represented in figure 4, which have been fitted to the performance values of the SVM algorithm.*

4- Discussion

In this study, we have compared different public datasets containing fall and ADL records, which was one of the remaining research efforts in the field of fall detection.

It has been shown in section 3.1 that the dataset used for checking an algorithm has some influence on the performance. This is clearer for NN, while for SVM this trend is mild, and with p-values never less than 0.03. NN is a pure data driven method, which does not rely on any internal parameters or assumption about distributions. Thus, it seems reasonable that the results depend strongly on the dataset.

It could be thought that the datasets are equivalent since the performance of SVM is very similar using any of them. However the results of section 3.2.1 indicate that this is not true. *tFall* generalizes far better than the other two. The datasets are different in terms of accelerometer characteristics, number of records, kinds of movements represented and placement of the device. The only a priori advantage of *tFall* with respect to both, *DLR* and *MobiFall*, is the number of samples. However, in section 3.2.2 we obtain the same conclusion when the datasets are tailored. In this case, one of the main effects that could remain is the variety of movements. It is likely that *tFall* includes many different types of ADL and falls. Eight types of falls are simulated in *tFall*, four types in *MobiFall*, while in *DLR* they are not specified. Besides, *tFall* includes ADL from real-life, recorded while people wear a smartphone. In this situation, there are many more kinds of movements that cannot be thought in a laboratory environment, like using the phone to call, take off the trousers, etc. When *MobiFall* is used for validation, training with *tFall* has also some advantage compared with *DLR*, since *tFall* and *MobiFall* registered movements with the sensor in the pocket. However, this fact cannot explain the difference between *tFall* and *MobiFall* when testing on *DLR*.

It should also be noted that table 9 could also have been arranged by merging the two rows with the same training set and two different validation sets. Then, it would have become more apparent than *tFall* is the hardest dataset to generalize on. For instance, when training with *MobiFall* and testing with *DLR* the performance is 0.696, while it decreases to 0.592 when testing on *tFall*. This result is the contrary to what could be expected from sensor placement (the same in the pair *MobiFall-tFall*, different in the pair *MobiFall-DLR*). However, it could also be explained by the fact that *tFall* has a large variety of movements acquired in a real environment. Bagalà et al. [19] also pointed out a decrease in performance when testing algorithms in real circumstances.

It is also worth highlighting the decrease in performance when a system is trained with a dataset and tested with a different dataset, as can be observed by comparing any of the tables in section 3.1 with those of section 3.2. Public datasets are an important step towards allowing the comparison and reproduction of studies on fall detection. However our results rise the question of whether it would be recommendable to train a fall detector for real use using these datasets, since the performance gets worse when generalizing to new acceleration patterns. This could lead to a dramatic decrease in performance when using the detectors in a real-world context, resulting in the rejection of the technology by its potential users. The personalization and adaptation of the system are key aspect to overcome this problem [20].

Regarding the influence of the accelerometer range, we have seen no clear difference between 7g and 2g in *DLR*. This contrast with previous studies [30,32] that recommend ranges above 2g. This can be due to the algorithms used. For threshold-based algorithms, the value at some particular point (peak, valley) is crucial for the classification. In the current study, the results rely on an integral measure, which does not depend so much on the value at a particular time.

A similar result has been found for the influence of the frequency, since sampling at 100 Hz is not better than sampling at 50 Hz. In most previous studies the sampling frequency is higher than 50 Hz [8]. Again this can depend on the algorithm, since the NN and SVM algorithms that we have presented use the raw acceleration values without performing any kind of filtering operations, in contrast with many previous works [13,32].

In the dataset with a higher number of records, *tFall*, we have seen the influence of the number of training records. Regarding the number of ADL, $NADL$, a saturation effect is observed and a 95% of the maximum performance is reached ($\exp(-x/c) = 0.05$) when $NADL = 469$ for NN or $NADL = 331$ for SVM. These values give a clue of the reasonable number of ADL needed, provided a variety of movements is represented. $NADL$ is higher for NN, a result that is expected since NN is based on a set of exemplars. With respect to the number of falls, we have not seen any saturation effect even for $NFALL = 300$. Thus, it seems that the number of falls included in the dataset is still insufficient to train the detector, even though most published works include far less falls.

5- Conclusion

The main contribution of this paper is the comparison and analysis of several datasets used in fall detection research. We have used two different classification algorithms and tested the datasets either with raw values or with tailored values in order to bring them to a baseline of similar conditions (accelerometer range, sampling frequency and number of records). As an overall conclusion of the paper, we recommend to test the algorithms using several datasets, since the results obtained with them are dissimilar and they seem to represent different kinds of movements. At best, algorithms should be trained with a dataset and validated with another to minimize the influence of the dataset on the results. This study has shown that in such situations the performances decrease considerably, which is an important point since this scenario is more representative of the real-world operation of the detectors. Among all the datasets analyzed, *tFall* is the one that generalizes better, including more records and types of falls. Nevertheless, datasets should include much more fall samples while the number of ADL included are enough. Recording movements from real life seems to be more suitable than recording them in the laboratory. Moreover, it would be good to have a dataset with movements of older people, the main target of fall detection systems. To the best of our knowledge, there is no acceleration dataset containing real data from older people that is publicly available to the scientific community. Forming this dataset is hard for falls, since it would require many volunteers for long periods to increase the probability of getting a real fall, but it is more feasible for ADL. Recording real data from a variety of sources will allow obtaining more realistic fall detection performances.

Acknowledgements

We acknowledge the “European Social Fund”, the “Departamento de Tecnología y Universidad del Gobierno de Aragón", the “TEC2013-50049-EXP Project” of the “Ministry of Economy and Competitiveness” and the “ImpulsAPP Elderly: Application for mental and physical wellbeing of older people living in urban and rural areas” of the “Fundación Impulso” for their support.

Conflict of interest

None.

References

[1] Lord SR, Sherrington C, Menz HB. Falls in older people. Risk, factors and strategies for prevention. Cambridge: University Press; 2001.

[2] Friedman SM, Munoz B, West SK, Rubin GS, Fried LP. Falls and fear of falling: Which comes first? A longitudinal prediction model suggests strategies for primary and secondary prevention. J Am Geriatr Soc 2002; 50: 1329-1335.

[3] Hartholt KA, van Beeck EF, Polinder S, van der Velde N, van Lieshout EM, Panneman MJ, et al. Societal consequences of falls in the older population: injuries, healthcare costs and long term reduced quality of life. J Trauma 2011; 71(3): 748-753.

[4] Centers for Disease Control and Prevention (CDC). Cost of fall injuries in older persons in the United States. Available online: http://www.cdc.gov/homeandrecreationalsafety/Falls/data/cost-estimates.html (accessed on 10 January 2015)

[5] World Health Organization (WHO). Falls. Available online: http://www.who.int/mediacentre/factsheets/fs344/en/ (accessed on 28 December 2014)

[6] Igual R, Medrano C, Plaza I. Challenges, issues and trends in fall detection systems. BioMedical Engineering Online; 2013: 12:66.

[7] Habib MA, Mohktar MS, Kamaruzzaman SB, Lim KS, Pin TM, Ibrahim F. Smartphone-based solutions for fall detection and prevention: Challenges and Open Issues. Sensors 2014; 14(4): 7181-7208.

[8] Schwickert L, Becker C, Lindemann U, Maréchal C, Bourke A, Chiari L, et al. Fall detection with body-worn sensors: a systematic review. Z Gerontol Geriatr 2013; 46(8): 706-719.

[9] Scheffer AC, Schuurmans MJ, van Dijk N, van der Hooft T, de Rooij SE. Fear of falling: measurement strategy, prevalence, risk factors and consequences among older persons. Age Ageing 2008; 37: 19-24.

[10] Rougier C, Meunier J, St-Arnaud A, Rousseau J. Robust video surveillance for fall detection based on human shape deformation. IEEE Trans Circuits Syst for Video Technol 2011; 21: 611-622.

[11] Mann S, Wearable Computing. Available online: http://www.interaction-design.org/encyclopedia/wearable_computing.html (accessed on 27 December 2014)

[12] Kangas M, Vikmanb I, Wiklanderc J, Lindgrenc P, Nybergb L, Jämsäa T. Sensitivity and specificity of fall detection in people aged 40 years and over. Gait Posture 2009; 29: 571-574.

[13] Bourke AK, van de Ven P, Gamble M, O'Connor R, Murphy K, Bogan E, et al. Assessment of waist-worn tri-axial accelerometer based fall-detection algorithms using continuous unsupervised activities. In: Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Buenos Aires, Argentina; Institute of Electrical and Electronics Engineers 2010; 2782-2785.

[14] Yuwono M, Moulton BD, Su SW, Celler BG, Nguyen HT. Unsupervised machine-learning method for improving the performance of ambulatory fall-detection systems. Biomed Eng Online 2012; 11: 1-11.

[15] Abbate S, Avvenuti M, Bonatesta F, Cola G, Corsini P, Vecchio A. A smartphone-based fall detection system. Pervasive Mob Comput 2012; 8: 883-899.

[16] Liu SH, Cheng WC. Fall Detection with the Support Vector Machine during Scripted and Continuous Unscripted Activities. Sensors 2012; 12: 12301–12316.

[17] Koshmak GA, Linden M, Loutfi A. Evaluation of the android-based fall detection system with physiological data monitoring. In: Proceedings of the 35th Annual International Conference of the IEEE on Engineering in Medicine and Biology Society (EMBC), Osaka, Japan, 2013, 3–7 July.

[18] Albert MV, Kording K, Herrmann M, Jayaraman A. Fall classification by machine learning using mobile phones. PLoS One 2012; 7: e36556.

[19] Bagala F, Becker C, Cappello A, Chiari L, Aminian K, Hausdorff JM, et al. J. Evaluation of accelerometer-based fall detection algorithms on real-world falls. PLoS ONE 2012; 7: e37062.

[20] Medrano C, Igual R, Plaza I, Castro M. Detecting falls as novelties in acceleration patterns acquired with smartphones. PLoS ONE 2014; 9(4): e94811.

[21] Korbinian F, Vera MJ, Robertson P, Pfeifer T. Bayesian recognition of motion related activities with inertial sensors. In: 12th ACM International Conference on Ubiquitous Computing, Copenhagen, Denmark, 2010, 26 - 29 Sep.

[22] Vavoulas G, Pediaditis M, Spanakis EG, Tsiknakis M. The MobiFall dataset: An initial evaluation of fall detection algorithms using smartphones. In: IEEE 13th International Conference on Bioinformatics and Bioengineering (BIBE), Chania, 2013, 1-4.

[23] Fudickar S, Karth C, Mahr P, Schnor B. Fall-detection simulator for accelerometers with in-hardware preprocessing. In Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments, PETRA ’12, New York, USA, 2012; 41: 1–7.

[24] DLR dataset. Available online: www.dlr.de/kn/en/Portaldata/27/Resources/dokumente/04_abteilungen_fs/kooperative_systeme/high_precision_reference_data/Activity_DataSet.zip (accessed on 2 January 2015)

[25] MobiFall dataset. Available online: http://www.bmi.teicrete.gr/index.php/research/mobifall (accessed on 2 January 2015)

[26] tFall: EduQTech dataset. Available online: http://eduqtech.unizar.es/fall-adl-data/ (accessed on 17 January 2015)

[27] Bishop CM. Pattern Recognition and Machine Learning (Information Science and Statistics), New York: Springer-Verlag; 2006.

[28] Chandola V, Banerjee A, Kumar V. Anomaly Detection: A Survey. ACM Comp Surveys 2009; 41(3): Article No. 15.

[29] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. J Machine Learning Research 2011; 12: 2825-2830.

[30] Klenk J, Chiari L, Helbostad JL, Zijlstra W, Aminian K, Todd C, et al. Development of a standard fall data format for signals from body-worn sensors: The FARSEEING consensus. Z Gerontol Geriatr 2013; 46(8): 720-726.

[31] Jones E, Oliphant T, Peterson P. SciPy: Open source scientific tools for Python, Available online: http://www.scipy.org/ (accessed on 27 December 2014).

[32] Kangas M, Konttila A, Lindgren P, Winblad I, Jämsä T. Comparison of low-complexity fall detection algorithms for body attached accelerometers. Gait Posture 2008; 28: 285-291.