



Article

# A Study of the Use of Gyroscope Measurements in Wearable Fall Detection Systems

Eduardo Casilari \*, Moisés Álvarez-Marco and Francisco García-Lagos<sup>†</sup>

Departamento de Tecnología Electrónica, ETSI Telecomunicación, Universidad de Málaga, 29071 Málaga, Spain; moisesnueva@gmail.com (M.Á.-M.); fgl@uma.es (F.G.-L.)

\* Correspondence: ecasilari@uma.es; Tel.: +34-952-132-755; Fax: +34-952-131-447

Received: 23 March 2020; Accepted: 15 April 2020; Published: 20 April 2020



**Abstract:** Due to the serious impact of falls on the quality of life of the elderly and on the economical sustainability of health systems, the study of new monitoring systems capable of automatically alerting about falls has gained much research interest during the last decade. In the field of Human Activity Recognition, Fall Detection Systems (FDSs) can be contemplated as pattern recognition architectures able to discriminate falls from ordinary Activities of Daily Living (ADLs). In this regard, the combined application of cellular communications and wearable devices that integrate inertial sensors offers a cost-efficient solution to track the user mobility almost ubiquitously. Inertial Measurement Units (IMUs) typically utilized for these architectures, embed an accelerometer and a gyroscope. This paper investigates if the use of the angular velocity (captured by the gyroscope) as an input feature of the movement classifier introduces any benefit with respect to the most common case in which the classification decision is uniquely based on the accelerometry signals. For this purpose, the work assesses the performance of a deep learning architecture (a convolutional neural network) which is optimized to differentiate falls from ADLs as a function of the raw data measured by the two inertial sensors (gyroscope and accelerometer). The system is evaluated against on a well-known public dataset with a high number of mobility traces (falls and ADL) measured from the movements of a wide group of experimental users.

**Keywords:** fall detection system; inertial sensors; wearable; accelerometer; gyroscope; convolutional neural networks

## 1. Introduction

The rise in life expectancy and the profound changes of the family structure have provoked a remarkable increase in the number of older people that live on their own. In this respect, falls are one of the most challenging risks to the autonomy and quality of life of the elderly. The World Health Organization has reported [1] that roughly a third of people aged over 65 fall at least once every year (with a prevalence of 50% for those over 80). According to certain clinical studies [2], up to 48% of fallers are incapable to get up on their own after a fall, thus, a third may remain on the floor longer than one hour. Unfortunately, a long period lying on the floor without assistance is closely linked to comorbidities (such as pressure sores, dehydration, pneumonia, or hypothermia) that cause long term hospital or care home admissions while remarkably increasing (up to 50% of the fallers) the probability of dying within six months [3,4]. In addition, from an economic point of view, the yearly medical costs attributable to older adult falls (only in the USA) have been estimated in \$50.0 billion [5]. With that in mind, the deployment of cost-effective Fall Detection Systems (FDS), has gained much attention in recent years, as a pivotal element to support the autonomy of older people and the affordability of health systems.

The general objective of an FDS is to track the movement of potential fallers and to automatically transmit an alarm (text message, phone call, etc.) to a remote monitoring point as soon as a fall is suspected to have taken place. Thus, FDSs are a particular typology of pattern recognition systems for human movements or HAR (Human Activity Recognition) intended to discriminate falls from ordinary movements or ADLs (Activities of Daily Living). FDS are designed to achieve a compromise between the need of minimizing the number of unnoticed falls and avoiding false alarms (namely ADLs misidentified as falls).

Depending on the nature of the employed sensors and the application environment, FDSs can be categorized into two generic groups [6–8]. The first group corresponds to Context-Aware Systems (CAS), which detect the occurrence of falls by analyzing the signals captured by a set of ambient and/or vision-based sensors, located in a preconfigured scenario around the patient (for example, a nursing home). The operation of CAS (or ambient-based) architectures (which normally entail high installation and maintenance costs) is restricted to a very particular zone and prone to errors caused by spurious elements [9] (falling objects or pets, changes in the lightning or in the position of the furniture, which create shadow or occlusion areas for the cameras, etc.). Furthermore, under this supervision, the user may feel his/her intimacy violated if permanent use of video-cameras or microphones is required.

On the contrary, the other general typology of FDSs (wearable systems) can undertake the monitoring of the subjects in an almost ubiquitous and continuous way thanks to transportable sensors that are attached to clothing, or embedded into a pendant or into any other personal device or accessory of the user. As long as the system is provided with a long-range connection (e.g., cellular telephony), the operability of a wearable FDS (and consequently the freedom of movements of the patient) is not constrained to a specific area. Moreover, the signals measured by the sensors unambiguously describe the mobility of the user to be supervised. Due to the increasing popularity and wide expansion of wearable devices among the general public, the research on wearable FDS has witnessed a boost during the last decade.

A wearable FDS typically consists of a set of inertial sensors and a central transportable node, which is in charge of classifying the user movements as a fall or as an ADL based on the measurements collected by the sensors. In most proposals, the sensors and the central node are integrated in the same device (usually a smartphone, which natively embeds an accelerometer and a gyroscope). Otherwise, the sensors are placed in independent external sensing nodes which can be attached to certain parts of the body more easily and which wirelessly connect to the central node (e.g., via Bluetooth), creating a Body Sensor Network.

The central decision element in an FDS is the detection algorithm. Existing algorithms for FDS can be roughly grouped into two main categories [10]: threshold-based and machine learning strategies. Threshold-based solutions detect a fall accident by contrasting the signals collected by the inertial sensors (e.g., the acceleration magnitude) with one or several values of reference which are supposed to be surpassed whenever a fall occurs. However, due to the variety and complexity of the movements associated to falls, these deterministic “thresholding” policies are normally too rigid and produce poor results. In fact, the other type of detection strategies (based on machine learning and artificial intelligence techniques) generally outperform thresholding methods [11].

When the fall detection problem is approached with pattern recognition methods, the arbitrary selection of a decision threshold is avoided. Instead, the detection algorithms are parametrized to infer the unknown function that links a set of statistical features derived from the mobility measurements and the correct classification decision (i.e., fall or ADL) that corresponds to every movement. Under supervised learning schemes (which are typically employed in most fall detection architectures, although there are also examples of unsupervised and semi-supervised FDSs), this parameterization is achieved by means of a training procedure during which the algorithm is configured to map the inputs (computed from the collected measurements of the movement) and the desired output (the binary categorization of the movement) for a predetermined set of labeled samples. In this field, deep learning architectures can be characterized as a type of machine learning that utilizes multiple

and successive processing layers, aiming at obtaining a representation of the data with different levels of abstraction [12]. The most interesting advantage of deep learning techniques is that they can automatically and directly extract from the raw data (in an FDS, those directly collected by the sensors) the most adequate features to represent the input patterns that must be classified. Hence, in contrast with most machine learning algorithms, deep learning techniques do not require a previous feature-extractor, which is often manually-engineered and strongly dependent on the expertise of the designer.

Convolutional Neural Networks (CNNs) are one of the most promising and widespread deep learning methods. In order to discover the underlying structure and interrelation of the input data in big datasets, CNNs employ several layers with convolutional filters (or kernels) that sequentially operate to condense the input data into a series of feature maps.

CNNs were originally envisaged for image recognition and classification systems, but they have been profitably applied to other domains, including HAR (Human Activity Recognition) systems [13,14]. Therefore, CNNs can replace those conventional machine learning techniques that entail some sort of “handcrafted” selection of input features, which must be computed from the inertial information captured by the wearable sensors prior to be fed into the classifier.

On another note, most wearable FDSs proposed in literature exclusively make use of the accelerometry signals as the input data that characterize the movements of the users to be monitored. However, in many of these proposals, the detection architecture was provided with an Inertial Measurement Unit (IMU), which integrates not only an accelerometer, but also a gyroscope (and in many cases a magnetometer, also). Consequently, since it is not particularly complex to access the information about the evolution of the angular velocity of the user’s body, it is of great relevance to assess whether the complementary use of the measurements delivered by the gyroscope can enhance the efficiency of the fall detection system.

In this paper, we employ CNNs (also considered in the FDS by He et al. in [15]), to evaluate the advantages of introducing the data collected by the gyroscope in the input features of the classifier intended for a wearable FDS. As the benchmarking tool, we employ the SisFall repository [16] (available in [17]), one of the largest public datasets containing accelerometry and gyroscope signals captured during the execution of falls and ADLs.

This paper is organized as it follows: after the introduction Section 2 revises the state of the art. Section 3 presents the employed dataset while Section 4 discusses the election of the input features for the neural classifier. Section 5 describes the architecture of the CNN, which is tested and systematically evaluated in Section 6 under different configurations. Finally, Section 7 recapitulates the main conclusions of the paper.

## 2. Related Work

As aforementioned, most studies in related literature about wearable FDSs exclusively consider the accelerometry signals to feed the fall detection algorithm. However, the idea of complementing the information of the accelerometer with that provided by the gyroscope is not new.

For example, the study by Casilari and Garcia-Lagos in [18], which reviews in detail all those FDSs that are based on artificial neural networks, reveals that up to 11 out of the 59 revised proposals utilize the angular velocity captured by the gyroscope to obtain some of the input features of the neural architecture that classifies the movements. To extend that bibliographical analysis, Table 1 summarizes those works that describe a system intended for fall detection in which the triaxial components of the angular velocity (or some parameters derived from them) are employed as inputs of the classifying algorithm. The table indicates in separate columns the typology of the algorithm implemented by each detector, the achieved performance metrics (expressed in terms of sensitivity and specificity, or failing that, the obtained accuracy) as well as the size of the experimental population (number of volunteers) with which the system was tested. Likewise, those proposed FDSs that make use of both artificial neural networks and the measurements of the gyroscope are recapitulated in Table 2. For each FDS,

the last column of this table itemizes the nature of the features utilized by the neural detector. Thus, that column specifies if they are statistics derived from the sensor measurements (which implies a heuristic selection of features and the need of a certain preprocessing phase to generate the inputs) or if the neural architecture is directly fed by the “raw” measurements so that a deep neural classifier can autonomously identify the features that allow discriminating the occurrence of falls from the monitored movements of the user.

Apart from these studies, in other works, such as in [19–26], the data obtained from the three-axis accelerometer, gyroscope and magnetometer are combined to obtain information about the orientation and tilt of the body, which are, in turn, input into the classifier.

Gyroscopes have even also been considered in hybrid FDSs that blend CAS and wearable approaches. In [27], Nyan et al. describe an architecture that employs the signals of a gyroscope to monitor the sagittal plane of the body, while the user is simultaneously monitored by a video-camera. Following a similar approach, Kepski et al. present a fuzzy system for fall detection in [28], grounded on the joint analysis of the signals captured by a Kinect device, an accelerometer and a gyroscope.

**Table 1.** Wearable FDSs that employ the information (acronyms are defined at the end of the table).

Authors	Ref.	Method	Sensitivity	Specificity	No. Subjects
Andò et al. (2016)	[29]	Threshold-based	0.55%–0.90%	100%	10
Astriani et al. (2018)	[30]	Threshold-based		95.82% (Accuracy)	<i>n.i.</i>
Baek et al. (2013)	[31]	Threshold-based	81.6%	100%	5
Bourke, O'Donovan et al. (2008)	[32]	Threshold-based	100%	100%	5
Bourke, Lyons et al. (2008)	[33]	Threshold-based	100%	100%	10
Boutellaa et al. (2019)	[34]	k-NN	Acc: 96.37%–96.85% (DLR Dataset)		19
			Acc: 88.76%–92.24% (Cogent Dataset)		42
Chen et al. (2010)	[35]	Threshold-based	<i>n.i.</i>	<i>n.i.</i>	3
Chernbumroong et al. (2015)	[36]	Fusion of RBF, SVM, MLP	Accuracy: 96.93%–97.29% (Acc.)		12
Choi et al. (2011)	[37]	NB	97.73%	100%	<i>n.i.</i>
Dau et al. (2014)	[38]	Genetic Programming	79–97%	76–97%	1
De Cillis et al. (2015)	[39]	Threshold-based	100%	100%	16
Dinh, Chew et al. (2015)	[40]	Threshold-based	30–90%	100%	<i>n.i.</i>
Dinh, Teng et al. (2009)	[41]	NB, SVM, RDRL	92.3%–97.3% (Accuracy)		<i>n.i.</i>
Dinh, Shi et al. (2009)	[42]	NB, SVM, RBF, C4.5	92.8%–97.3% (Accuracy)		1
Dzeng et al. (2014)	[43]	Threshold-based	4%–88% (Accuracy)		4
Figueiredo et al. (2016)	[44]	Threshold-based	100%	93%	2
Guo et al. (2016)	[45]	Threshold-based	<i>n.i.</i>	Up to 100%	6
Hakim et al. (2017)	[46]	k-NN, SVM, DT, Discriminant Analysis	74.3%–99% (Accuracy)		8
He, Hu et al. (2016)	[47]	kNN, NB, DT (J48), Bayes Net, MLP, Bagging, Ripper	91.1%–93.8%	97.6%–99.1%	15
He, Li et al. (2013)	[48]	Threshold-based DT (Fisher's discriminant ratio and J3 Criteria were used for feature selection)	97.63% (accuracy)		10
Huynh et al. (2015)	[49]	Threshold-based	96.55%	89.50%	36
Hwang et al. (2004)	[50]	Threshold-based	95.55%	100%	3
Lai et al. (2014)	[51]	K-Means and Bayesian inference	75%–95%	<i>n.i.</i>	10
Li et al. (2009)	[52]	Threshold-based	<i>n.i.</i>	<i>n.i.</i>	3
Majumder, Zerin et al. (2013)	[53]	DT	71%–98% (Accuracy)		15
Majumder, Rahman et al. (2013)	[54]	DT	28%–98% (Accuracy)		5
Martinez-Villaseñor et al. (2018)	[55]	LDA, DT, NB, SVM, RF, K-NN	15.25%–69.18% (Accuracy)		4
Nari et al. (2016)	[56]	Threshold-based	90%	86.7%	1
Ntanasis et al. (2017)	[57]	DT(J-48), k-NN, RF, Random Committee, SVM	84.13%–99.64%	92.37%–99.82%	14 (Erciyes Dataset)
Nyan et al. (2008)	[58]	Threshold-based	95.2%	100%	21
Ojetola et al. (2011)	[59]	DT (C.45)	62.5%–100%	<i>n.i.</i>	8
Özdemir et al. (2016)	[60]	k-NN, BDM, SVM, LSM, DTW, ANN	92.40–99.91% (Accuracy)		14 (Erciyes Dataset)
Park et al. (2011)	[61]	Threshold-based	<i>n.i.</i>	<i>n.i.</i>	3
Quadros et al. (2018)	[62]	Threshold-based, DT, k-NN, LDA, LR, SVM	97.9%–100%	93.8%–97.9%	22
Rakhecha et al. (2013)	[63]	Threshold-based	86%–94%	73%–88%	<i>n.i.</i>
Rungnapakan et al. (2018)	[64]	Threshold-based	79%–100%	96.67%–100%	6
Santiago et al. (2017)	[65]	Threshold-based	83%–92%(Accuracy)	<i>n.i.</i>	
Sorvala et al. (2012)	[66]	Threshold-based	95.6%	99.6%	2
Tamura et al. (2009)	[67]	Threshold-based	93% (Accu.)		16
Wibisono et al. (2013)	[68]	Threshold-based	85%–100%	<i>n.i.</i>	<i>n.i.</i>
Yang et al. (2013)	[69]	Threshold-based	<i>n.i.</i>	6.67%–100%	12
Zhao et al. (2012)	[70]	Threshold-based	<i>n.i.</i>	<i>n.i.</i>	8

Acronyms: ANN: Artificial Neural Networks; BDM: Bayesian Decision Making; DT: Decision Tree; DTW: Dynamic Time Warping; LDA: Linear Discriminant Analysis; LR: Logistic Regression; LSM: Least Squares Method; MLP: Multi-Layer Perceptron; NB: Naive Bayes; *n.i.*: Not indicated in the corresponding publication; RBF: Radial Basis Function; RDRL: Ripple Down Rule Learner; RF: Random Forest; RDRL: Ripple Down Rule Learner; SVM: Support Vector Machine.

**Table 2.** Wearable FDSs based on artificial neural networks that employ the information from the gyroscope.

Authors (Date)	Ref.	Sensitivity	Specificity	Input: Raw Data/Derived Features
Ahmed et al. (2017)	[71]	67.9%–77.8% (Accuracy)		Raw data
Chelli and Patzold (2019)	[72]	96.8%–99.11%	100%	328 derived features
Chernbumroong et al. (2015)	[36]	96.93%–97.29% (Acc),		202 derived features
Ghazal et al. (2015)	[73]	73.31%	93.33%,	Derived features
He et al. (2019)	[15]	100%	99.74%	Raw data
Martínez Villaseñor et al. (2018)	[55]	64.03%–68.66% (Accuracy)		33 derived features
Nukala et al. (2014)	[74]	96%–98%	96.5%–98.1%	6 derived features
Özdemir and Barshan (2014)	[75]	97.47%	93.44%	78 derived features
Özdemir and Turan (2016)	[60]	94.20%–96.27% (Accuracy)		78 derived features
Rashidpour et al. (2016)	[76]	100%	100%	Raw data
Wang and Zhang (2015)	[77]	95%–100%	100%	3 derived features
Yang et al. (2013)	[78]	70%	92.26%	Raw data
Yodpijit et al. (2017)	[79]	99.37%	99.23%	Raw data

In spite of the small—but not negligible—number of papers that have addressed the use of the gyroscope in FDSs, the possible incremental benefits of introducing this element (or other sensors) in the detection process (when compared to the case in which only the accelerometer is contemplated) have been discussed in only a few studies.

For example, Bianchi showed that the performance of an accelerometer-based detector can produce better results if it is complemented by a barometric pressure sensor [80].

Authors in [81] state that accelerometer-only based detectors can offer a high detection accuracy that could be improved if additional motion sensors (including gyroscopes) are employed. However, no systematic analysis of the advantages of using a gyroscope is provided. De Cillis et al. show in [39] that the combined use of the information provided by the accelerometer and the gyroscope can reduce the number of false alarms. In particular, the detection algorithms integrate the measurements of the vertical component of the angular velocity to estimate the heading of a user equipped with a waist-worn mobile sensor and a smartphone. Nonetheless, the proposed threshold-based system is tested against only 30 samples (14 falls and 16 ADLs).

The study of the prototype presented by Astriani et al. in [30] suggests that the combined use of accelerometry and gyroscope signals in basic threshold-methods seems to improve the accuracy of the detector, although the system was evaluated with a small set of falls (only 84). Similarly, according to the results presented by Hakim et al. in [46], the accuracy of a smartphone-based fall detector (using different machine learning strategies) augments as the number of considered IMU sensors increases. However, only six experimental subjects (and a very small sampling rate of 10 Hz) were employed to generate the training samples of the detection methods. In addition, the process of feature extraction to feed the classifiers is not described.

The study by Dzenget al. [43] compares an accelerometer-based and a gyroscope-based algorithm intended for fall detection with smartphones. Results indicate that the accuracy of the algorithm that utilizes the gyroscope signals is more affected in case of work-related motions.

The study by Yang et al. in [69] analyses the advantages of combining the detection decision of two separate threshold-based algorithms: one using the signals captured by the accelerometer and the other one the measurements from the gyroscope. As a fall is only presumed when both algorithms detect it, the specificity is obviously increased (although the cost in terms of sensitivity is not studied).

Chelli et al. showed in [72] that the detection ratio of four different machine learning strategies can be enhanced if a wide set of features extracted from the signals of the accelerometer and gyroscope are considered. This improvement is achieved at the cost of requiring of a high number of input features (328). In a real scenario, these sophisticated features should be computed in real time to continuously feed the classifier, which could pose a problem to the limited computing capacity of current wearables.

In this regard, there is a family of FDS that try to benefit from a “multi-sensor fusion approach”, i.e., from the simultaneous use of several sensors. Gyroscopes and altimeters are assumed to be helpful to improve our understanding of the falling dynamics [82].

In some cases, the gyroscope, or the information about the orientation, is used to confirm the horizontal position of the user after the fall. For example, this strategy is followed by the smartphone only based FDS presented by Viet et al. in [83]. A similar policy is assumed by Rashidpour et al. in [76]. These authors suggest that falls are better identified by the accelerometer rather than by the gyroscope. By separately employing both elements, the number of undetected falls could diminish (but at the cost of decreasing the specificity).

Tsinganos et al. [84] compare four different algorithms extracted from literature to assess the effects of the fusion of data captured by diverse sensors. Nevertheless, the incremental improvement achieved by using the data from the gyroscope is not analyzed. In a multisensory scheme, mutual information analysis is utilized by Chernbumroong [36] to evaluate the importance of different sensors for the movement classification, concluding that the most relevant features are those derived by two particular (Z- and Y-) acceleration components.

Furthermore, other studies highlighted the different costs linked to the use of the gyroscope. For example, it was proved [85,86] that the operation of this sensor normally requires more energy than the accelerometer. Figueiredo et al. present a smartphone-based FDS in [44], in which diverse features (calculated from the acceleration and angular velocity) are used to produce the detection decision. In their analysis, the authors also highlight that the demanded energy and economical costs of the gyroscope are higher than those of an accelerometer. On the contrary, Nguyen et al. report in [87] an increase of less than 3% of the consumption in a sensing module (comprising a gyroscope and an accelerometer) when both sensors are activated (with respect to the case in which only the acceleration magnitudes are measured).

In the next sections, we propose to optimize the hyperparameters of a Convolutional Neural Network to detect falls from the raw signals captured by the accelerometer and the gyroscope. Once the final deep learning architecture is parameterized, in order to assess the benefits achieved by the use the gyroscope, the detector is also trained and tested taking into consideration only the accelerometry signals.

### 3. Employed Dataset

The effectiveness of an FDS is evaluated by assessing the binary responses that it provides when it is fed with a set of known mobility patterns (ADL or fall). Due to the evident and intrinsic complexity of testing an FDS against the actual falls suffered by the target population (the elderly) captured in a real scenario, almost all authors in the related literature utilize datasets obtained from a group of volunteers that systematically emulate a set of predefined ADLs and mimicked falls. Thus, these “laboratory-created” falls are normally collected by monitoring the movements of young volunteers that simulate falling on a cushioned surface to avoid injuries. In this regard, we have to emphasize that the appropriateness of testing a FDS with ADLs and, in particular, falls mimicked by a set of young healthy experimental subjects on a cushioning element is a controversial issue still under discussion and out of the scope of this work (see, for example, the discussion presented by Kangas et al. in [88]). In this respect, some researchers, such as Klenk et al. in [89], discovered non-minor discrepancies between the patterns of real-world falls and those mimicked or executed on a padded element. On the other hand, after analyzing the dynamics of actual falls experienced by elderly people, Jämsä et al. conclude in [90] that intentional and real life falls may exhibit a similar behavior.

Although many authors do not make the traces generated for their experiments publicly available, during the few last years, some repositories (specifically designed for the analysis of wearable FDSs) were published to be used as a benchmarking tool by the research community.

Table 3 summarizes the main characteristics of most of these released datasets (see [91] for a further insight), which contain a list of numerical series with the measurements obtained from one or

several sensing units transported by the experimental users during the execution of the movements. The series provided by these repositories are labelled to indicate if the action corresponds to an ADL or a fall.

**Table 3.** Summary of most popular datasets for the study of FDS.

Name or Origin of the Dataset and Reference	No. of Types of Emulated ADLs/Falls	No. of Samples (ADLs/falls)	No. of Subjects (F/M)	Positions of the Sensing Motes	Sensors in the Motes
DLR [92]	15/1	1017 (961/56)	19 (8/11)	Waist (belt)	A, G, M
Cogent Labs [93]	8/6	1968 (1520/448)	42(6/36)	Chest and Thigh	A, G
MobiFall [94]	9/4	630 (342/288)	24 (7/17)	Trouser pocket	A, G, O
MobiAct [95]	9/4	2526 (879/647)	57 (15/42)	Trouser pocket	A, G, O
TST [96]	4/4	264 (132/132)	11 ( <i>n.i.</i> )	Right wrist and Waist	A
SINTEF ICT [97]	7/12	117 (45/72)	2 ( <i>n.i.</i> )	Waist and Wrist	A
tFall [98]	Real Life conditions/8	10,909 (9883/1026)	10 (3/7)	Pocket, Hand bag	A
SisFall [16]	19/15	4505 (2707/1798)	38 (19/19)	Waist	A, G
UR Fall Detection [99]	5/3	70 (40/30)	5 (0/5)	Near pelvis (waist)	A
Erciyes University	16/20	3302(1476/1826)	17 (7/10)	Chest, Head, Ankle, Thigh, Wrist, Waist	A, G, O
UMAFall [100]	8/3	531 (322/209)	17 (7/10)	Ankle, Chest, Thigh, Waist Wrist	A, G, M
UniMiB SHAR [101]	9/8	7013 (5314/1699)	30 (24/6)	Left or right trouser pocket	A
Graz University of Technology [102]	10/4	492 (74/418)	10 ( <i>n. i.</i> )	Waist (belt bag)	A, O

*n.i.*: Not indicated in the corresponding publication.

By analyzing the data provided by this table, we select the SisFall dataset [16,17] not only because it integrates the measurements of both a gyroscope and two accelerometers, but also in the light of the wide size and age range of the experimental volunteers (38 participants including 19 males and 19 females between 19–75 years), the number and duration of samples (4505 samples between 10 and 180 s, comprising 2707 ADLs and 1798 falls) and the large diversity of emulated movements (19 classes of ADLs including basic and sporting activities- and 15 classes of falls).

In this dataset, every experimental subject repeated each type of movement five times, except for the cases of jogging and walking, which were carried out just once but during a longer time interval (100 s).

The sensing node was deployed on a microcontroller-based platform, specifically designed for this purpose, which integrated an ITG3200 gyroscope, two accelerometers (models ADXL345 and MMA8451Q) and an SD card to store the traces. By using an elastic belt, the mote was fixed to the waist. This position is considered to offer an adequate location to characterize the dynamics of the trunk [57] as it is normally near the human body center of gravity and not particularly affected by the individual movements of the limbs. The employed sampling frequency was 200 Hz, which is far above the minimum rate (20–40 Hz), which was suggested for an adequate modelling of the fall mobility [103]. In our analysis, we consider the series captured by the MMA8451Q accelerometer as it was programmed with a higher range ( $\pm 16$  g) than the ADXL345 model ( $\pm 8$  g). The range of the gyroscope is  $2000^\circ/\text{s}$ .

#### 4. Discussion on the Input Features

The election and design of a fall detection algorithm for an FDS cannot overlook the fact that most wearables present remarkable limitations in terms of storing capacity, power consumption and real-time computation. Consequently, a complex preprocessing of the data measured by the sensors, which could facilitate the task of the detector, may be unviable if it implies an intensive use of the hardware device or a rapid depletion of the battery. Thus, we opt for using the raw measurements obtained from the sensors (the accelerometer and the gyroscope) as the input features of the classifier. Under this approach, in contrast to other proposals, the real-time calculation of a set of arbitrarily chosen statistics computed from the measurements (moments, autocorrelation coefficients, etc.) is not required.

In particular, we propose to focus the analysis of the detector on those intervals (or observation windows) where a fall is most likely to have occurred.

Human falls are typically connected to one or several brusque peaks of the acceleration originated when the body hits the ground [104]. This acceleration peak (or peaks) is preceded by a period of “free fall” during which the acceleration module rapidly decays. In addition, both before and after the impact, the body orientation is normally strongly altered, which is reflected in abrupt alterations of the values of the three coordinates of the acceleration as well as in noteworthy changes of the angular velocity in the three axes measured by the gyroscope. Therefore, we take the peak of the acceleration module as the basis to define the observation window of the detector.

This acceleration module or Signal Magnitude Vector ( $SMV_i$ ) for the  $i$ -th sample is computable as:

$$SMV_i = \sqrt{A_{x_i}^2 + A_{y_i}^2 + A_{z_i}^2} \quad (1)$$

being  $A_{x_i}$ ,  $A_{y_i}$  and  $A_{z_i}$  the  $x$ ,  $y$ , and  $z$  components of the vector measured by the triaxial accelerometer, respectively, while  $A_{Y_i}$  denotes the component in the vertical direction (perpendicular to the floor plane) when the monitored subject is standing.

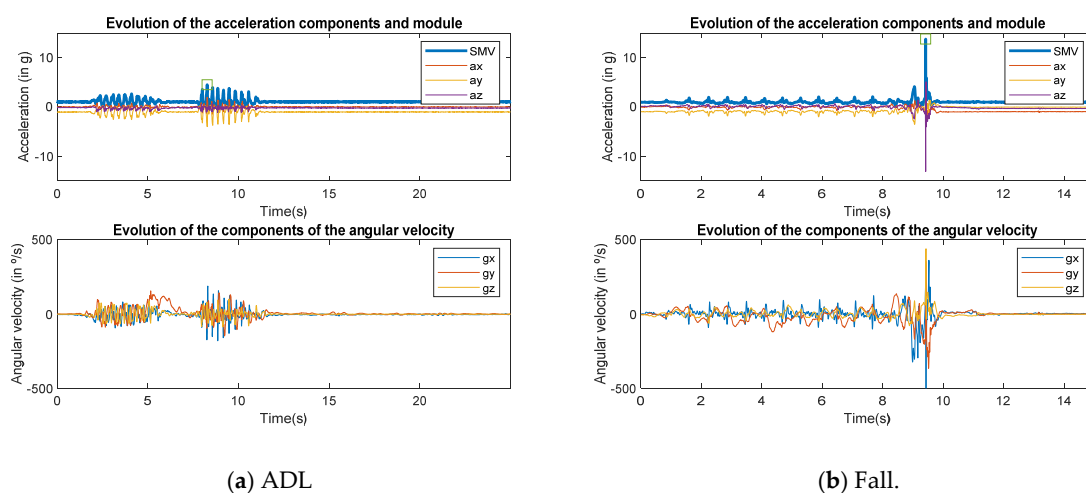
For each trace, the maximum (or peak) of the SMV is defined as:

$$SMV_{max} = SMV_{k_0} = \max(\{SMV_i : i \in [1, N]\}) \quad (2)$$

where  $n$  is the number of samples of the movement captured in the trace while  $k_0$  indicates the instant (sample index) at which this maximum is found.

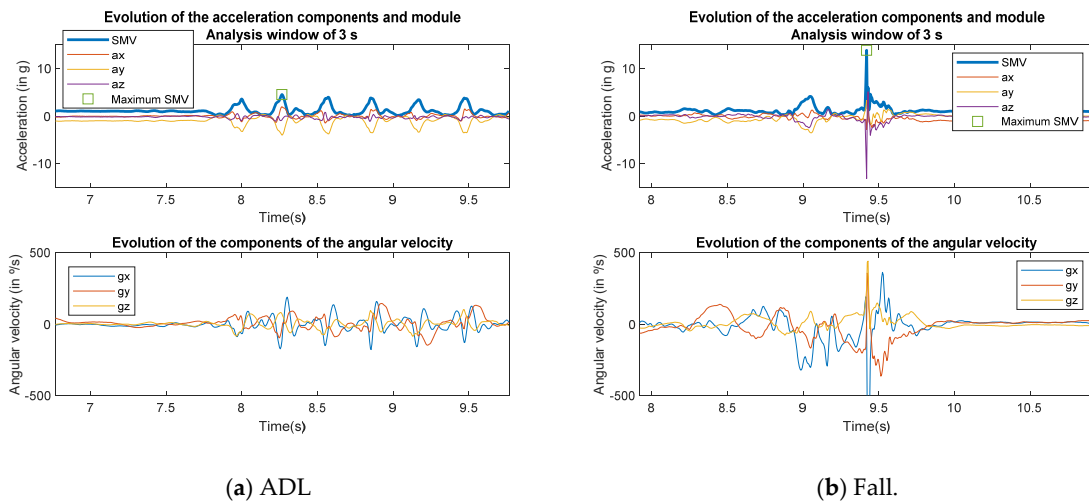
As long as the movements associated to a fall typically extend between 1 s and 3 s [105], we will consider an observation time window of up to 8 s around  $SMV_{max}$  (4 s before and after the peak) to select the measurements that will feed (as input mobility patterns) the detection algorithm. This procedure of selecting an time window around the detected peak is followed in other studies such as [98,106–108].

To illustrate the dynamics of the two different types of activities that must be discriminated, Figure 1 displays the measured acceleration components and SMV, as well as the components of the angular velocity, for a certain ADL (walking upstairs and downstairs quickly) and a fall (caused by a mimicked slip while walking). In turn, Figure 2 zooms into the same series to show a 3-s observation window around the detected acceleration peak (which is indicated in the figures with a square marker).



**Figure 1.** Example of the progress of the accelerometer and gyroscope measurements captured for an Activities of Daily Living (ADL) and a fall.





**Figure 2.** Example of the progress of the accelerometer and gyroscope measurements captured for an ADL and a fall during an observation window of 3 s ( $\pm 1.5$  s).

In Section 6, the impact of the election of the observation window is investigated.

Once the window size is fixed and the maximum is found, the input sequence ( $I$ ) that feeds the detector is formed by simply concatenating the series of triaxial measurements of the accelerometer and the gyroscope (forming a tuple of six elements) in that period of time:

$$I = \left\{ A_{x_j}, A_{y_j}, A_{z_j}, G_{x_j}, G_{y_j}, G_{z_j} \mid j \in \left[ k_0 - \frac{T_W}{2} f_s, k_0 + \frac{T_W}{2} f_s \right] \right\} \quad (3)$$

where  $G_{x_j}, G_{y_j}, G_{z_j}$  describe the three components of the angular velocity measured by the gyroscope for the  $j$ -th sample,  $T_W$  is the duration of observation window and  $f_s$  the sampling rate of the sensors (200 Hz).

## 5. Architecture of the CNN

The general objective of a fall detector is to discriminate falls from ADLs based on the measurements collected by the mobility sensors. In the case of using a supervised machine learning technique, the classifier must be trained with pre-recorded samples to map a certain number of “input features” (computed from the sensor measurements) and the corresponding binary decision expressed in terms of an output probability of 0 or 1 in the outputs.

As already mentioned, one of the main advantages of Convolutional Neural Networks is that they are able to autonomously learn the internal structure of the data to generate the output, avoiding the preprocessing and the heuristic choice and calculation of “intermediate” variables derived from the raw measurements, which are required by other artificial intelligence methods.

A CNN essentially consists of a series of sequential (and alternate) convolutional and pooling layers that extract the features to feed a final classifying, which produces the final output decision.

Every convolution layer applies to the input features a series of linear convolution filters followed by a non-linear activation function, typically the Rectified Linear Unit—ReLU—function, defined as  $f(z) = \max(z, 0)$  [12]. The obtained results are then down-sampled by a pooling layer, commonly a max-pooling operator that moves across the input values to select only the maximum for every region of a predetermined size within the input data map.

A parameter called “stride” defines the step size utilized by the convolutional filters to slide across the data. Thus, depending on the selected value of the stride, the regions to which this processing is applied may overlap or not.

In any case, the combination of these two layers (convolutional and pooling) yields a certain abstract representation of the data, which is employed as the input features of the next layer. During the training process, the coefficients of the convolutional operators are jointly tuned to optimize the response of the final classifier whose architecture (i.e., neuron’s weights) is also adapted to produce the output that corresponds to the input training patterns.

Although CNNs were conceived for image processing, they can also be successfully utilized in the analysis of temporal (one-dimensional) sequences, as it is the case of the inputs received by an FDS.

The use of small-size and adjustable convolutional filters allows CNNs to be responsive to a set of very particular characteristics of the input signals (e.g., a sudden acceleration peak or decay or a short period in which the variation of the angular velocity in a particular direction is noteworthy), which are extracted and learned from the data without requiring any further intervention of the user. The pooling layers in turns enables the “translation invariance” of the key features (that is to say, they are detected with independence of their position—the moment within the sequence—or observation interval). Thus, the features learned by the precedent convolutional layer are filtered into a condensed map of features that summarize the main characteristics of a certain region of the input data.

The learned features of the last convolutional/pooling layer are fed to the classifier, in our case, a fully-connected neural network layer. The linear weighted sum of the features produced by the neurons is passed through a softmax (or normalized exponential) function, which normalizes the values into a probability distribution that is used by a final stage to assign one of the two mutually exclusive output classes (fall or ADL).

#### *Training of the CNN*

To prevent overfitting and, consequently, a loss in the capability of the model to generalize, a cross-validation method is conducted. For this purpose, the dataset is partitioned into three independent set of samples: training, validation and test sets, with 60%, 20% and 20% of the samples, respectively. This data segmentation is randomly executed, but, previously, it was ensured that all the sample groups have the same proportion of falls and ADLs.

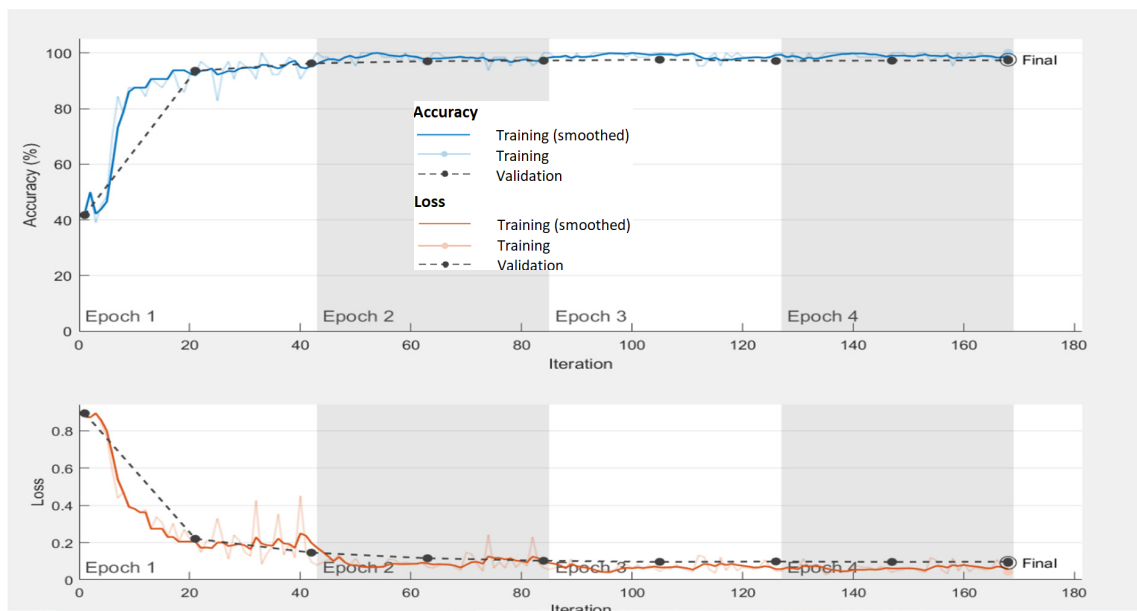
The validation set is employed to assess the performance of the CNN once that it is trained for an epoch (a single pass through the entire training set). In the case that the mean error of the outputs obtained for the validation samples increases for a certain number of times (validation patience), the training phase is stopped as the CNN is supposed to start overfitting the training data. If no overfitting is detected, the training process finishes after a predetermined number of epochs (in our case, five epochs). Consequently, the reported global effectiveness of the CNN is computed by employing the test set.

To diminish the impact of the random division of the dataset, we perform a five-fold validation, by executing five rounds of training, validation and testing with five different subsets of samples, randomly extracted from the dataset. Accordingly, the final presented results correspond to the average of the quality metrics obtained for the five rounds.

In addition, during training, the weight decay (or L2 Regularization) technique is applied to reduce the values of the weights as another method to prevent overfitting.

Figure 3 shows an example of the evolution of the loss and accuracy of the training progress. The figure illustrates that the convergence is attained after very few epochs.

Tables 4 and 5 presents the parameters (or hyper-parameters) that define the initial architecture and training process of the CNN. The impact of the election of the hyper-parameters included in Table 4 is investigated in the next section.



**Figure 3.** Training progress: Training and validation accuracy curves.

**Table 4.** Initial set of hyperparameters for the employed Convolutional Neural Networks (CNN).

Parameter	Value
Number of convolutional layers	4
Number of max pooling layers	3
Activation functions	ReLU for convolutional layers, Softmax for output layer
Type of output layer	Fully connected
Max Pooling Window	$1 \times 5$
Number of filters in each layer	16 (1st layer)-32 (2nd)-64 (3rd)-128 (4th)
Size of the filters (for all layers)	$1 \times 5$
Pool size of the max-pooling layer (Max Pooling Window. MPW = 5)	$1 \times 5$
Mini batch size	64 training instances

**Table 5.** Other parameters of the employed CNN.

Training Method	Stochastic Gradient Descent with Momentum
Learning rate	0.0001
Validation patience	2
Iterations per epoch	42
Maximum number of training epochs	5
Size of zero-padding	2 samples

## 6. Performance Analysis

We utilized scripts using the Matlab Neural Network Toolbox<sup>TM</sup> [108] to implement the CNN. As the employed functions are conceived for image processing, we define an equivalent “image” of ( $1 \times width$ ) “pixels” as the input features of the classifier. The parameter “width” defines the size of the vector with the measurements (triaxial components of the acceleration and angular velocity) captured during the observation interval around the peak. This size or number of input features ( $N_i$ ) straightforwardly depends of the duration of the observation window ( $T_w$ ):

$$N_i = 6 \cdot (T_w \cdot f_s + 1) \quad (4)$$

To appraise the performance of the detector, three typical metrics quality metrics (commonly used for binary pattern classifiers) are considered:

(1) Sensitivity (or recall), which describes the ability to identify falls:

$$\text{Sensitivity} = \frac{TP}{FN + TP} \quad (5)$$

where  $TP$  (“True Positives”) and  $FN$  (“False Negatives”) respectively indicate the numbers of test movement samples containing falls that were correctly and wrongly identified.

(2) Specificity, which assesses the competence of the FDS to prevent ADLs misidentified as falls (which are supposed to trigger a false alarm):

$$\text{Specificity} = \frac{TN}{FP + TN} \quad (6)$$

where  $TN$  and  $FP$  denote the numbers of “True Negatives” (ADLs that are properly detected) and “False Negatives” (ADLs misclassified as falls).

(3) Accuracy, which provides a global metric of the effectiveness of the system, defined as the ratio:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (7)$$

The results obtained (for an observation time window ( $T_W$ ) of three seconds) with the test data and the initial (reference) architecture of the network are offered in Table 6.

**Table 6.** Performance parameters for the reference architecture ( $T_W = 3$  s).

Metric	Result
Accuracy	0.949
Sensitivity	0.911
Specificity	0.974

To assess the tenability of the detection model, we modify the most relevant hyperparameters of the configured network as well as the size of the observation window. As this analysis progresses, we choose those combinations of parameters that yield the best performance metrics (marked in bold in the corresponding tables).

Firstly, we investigate the effects of the size of the Max Pooling Window (MPW) (i.e., the size of the region where max pooling operator is applied). From the results tabulated in Table 7, we can conclude that a smaller value of this window is preferable, thus MPW is set to three for the next experiments.

**Table 7.** Results for different sizes of the Max Pooling Window (MPW).

Metric	MPW = 5	MPW = 3
Accuracy	0.949	0.970
Sensitivity	0.911	0.948
Specificity	0.974	0.985

The study of the impact of the considered duration of the observation window ( $T_W$ ) is presented in Table 8, which reveals that the best global efficiency of the FDS is achieved with an observation interval of five seconds ( $\pm 2.5$  s) around the peak.

The repercussion of the dimension of the convolutional filters is summarized in Table 9, which shows that a higher size ( $1 \times 30$ ) improves the specificity. Accordingly, we set the dimension of the filters to  $1 \times 30$  (except for the last convolutional layer, for which a filter of  $1 \times 10$  samples is configured as long as the number of inputs in this layer is very reduced).

**Table 8.** Results for different values of the observation window ( $T_W$ ).

Duration of the Observation Window Around the Peak ( $T_W$ )					
Metric	1 s	3 s	5 s	6 s	8 s
Accuracy	0.956	0.970	0.975	0.973	0.973
Sensitivity	0.910	0.948	0.956	0.957	0.955
Specificity	0.987	0.985	0.987	0.984	0.985

**Table 9.** Results for different sizes of the convolutional filters.

Dimension (in Samples) of the Filters				
Metric	$1 \times 5$	$1 \times 10$	$1 \times 20$	$1 \times 30$ <sup>1</sup>
Accuracy	0.975	0.976	0.975	0.979
Sensitivity	0.956	0.954	0.958	0.957
Specificity	0.987	0.990	0.987	0.993

<sup>1</sup> The fourth convolutional layer employs a  $1 \times 10$  filter.

Table 10, in turn, displays the influence of the number of convolutional layers. Results indicate that an architecture of only three layers (which reduces the complexity of the system) improves the sensitivity at the cost of a slight decay of the specificity. Thus this three-layer topology is preferred.

**Table 10.** Results for different numbers of convolutional layers.

Number of Convolutional Layers			
Metric	2	3	4
Accuracy	0.970	0.978	0.979
Sensitivity	0.945	0.960	0.957
Specificity	0.987	0.990	0.993

Tables 11 and 12 investigate in turn the effects of the election of the number of filters per layer and the use of the ReLU layer (which can be optionally employed in a CNN). As no improvement is achieved with respect to the reference case, these hyperparameters are not altered. On the contrary, Table 13 shows that a better performance is obtained if a smaller size is selected for the mini-batches (subset of the training samples that is utilized to compute the gradient of the loss function in order to update the weights).

**Table 11.** Results for different numbers of filters per layer.

Number of Filters in Each Layer			
Metric	16-32-64	4-4-4	64-64-64
Accuracy	0.978	0.923	0.977
Sensitivity	0.960	0.880	0.957
Specificity	0.990	0.952	0.991

**Table 12.** Effect of using the ReLU layers.

Metric	Results without ReLU Layers	Results with ReLU Layers
Accuracy	0.978	0.938
Sensitivity	0.960	0.902
Specificity	0.990	0.961

**Table 13.** Effect of the size of the mini-batch used during training.

Metric	Mini-Batch Size		
	64	32	16
Accuracy	0.978	0.981	0.982
Sensitivity	0.960	0.964	0.968
Specificity	0.990	0.992	0.991

Once that the hyper-parameters and the observation window were chosen to maximize the results, we evaluate the advantages of a multi-sensory approach. Thus, we compare in Table 14 the obtained performance of the final architecture with that achieved when the CNN is trained to discriminate falls just based on the data captured by the accelerometer.

**Table 14.** Comparison with the case in which only the accelerometer signals are employed.

Metric	Employed Signals	
	Accelerometer and Gyroscope	Only Accelerometer
Accuracy	0.981	0.994
Sensitivity	0.964	0.988
Specificity	0.992	0.997

Surprisingly, Table 14 shows that the best metrics are obtained in the case in which only the acceleration signals are considered. This seems to indicate that the information provided by the gyroscope does not offer any supplementary information to improve the effectiveness of the detection decision in the CNN. Thus, the dimension and complexity of the system can be reduced by half (by just considering the acceleration signals) without any performance loss.

This results are coherent with those recently presented by Boutella et al. in [34]. These authors also analyze the benefits the using the combined information retrieved from multiple sensors when a k-NN classifier is considered. The classifier is fed with the covariance matrix of the signals of the sensors collected during a fixed interval. When the detection method is applied to two public datasets (Cogent and DLR, described in Table 3), the obtained performance metrics (which do not exhibit a high accuracy, in particular for the Cogent dataset) also suggest that the joint utilization of gyroscopes and accelerometer may even deteriorate the effectiveness of the detector.

Nevertheless, the proposal yields better results (with values for the specificity and the sensitivity around 99%) than those achieved by other works in the literature that use the same SisFall repository as the benchmarking tool [16,103,109–111]. Similarly, our system also outperforms the FDS based on a Recurrent Neural Networks (RNNs) analyzed in [112], which is tested with three different datasets.

In any case, we cannot forget that the most adequate validation of any FDS should be carried in a realistic scenario with real participants. Although there is a non-negligible number of commercial devices intended for fall detection (normally marketed in the form of a pendant or a watch), vendors do not usually inform about neither the detection method that the detector implements nor about the procedure that was followed to test its effectiveness when it is worn by a real user (in some cases, the prototypes are just tested with mannequins). In fact, in some clinical analyses, such as that provided by Lipsitz et al. in [113], the authors showed that the fall detection ratio of this type of on-the-shelf device is still far from being satisfactory. In that study, after monitoring nursing home residents over six months, only 17 of the 89 actual falls recorded by the nursing staff did not remain unnoticed while 111 false alarms (totaling 87% of the alerting messages) were generated by the detection system.

In this regard, aspects such as usability, ergonomics and comfort are key factors to guarantee the social acceptance of fall detectors among older adults (the main target public of these monitoring applications). Unfortunately, these subjective characteristics are not considered in most articles dealing with new proposals of prototypes for FDS. However, these topics are arousing an increasing interest in the research on fall prevention and detection in the area of geriatrics (see, for example, [114–116] for

a systematic review on these subjects). In a recent study by Thillo et al. [117], community-dwelling older people were asked to carry a prototype of a wearable FDS during nine days. After several types of qualitative analysis based on discussions with the users, authors conclude that the usability is not only determined by technical requirements but also strongly influenced by the habits and personal preferences of the final users. Thus, an adequate design of an FDS can be only achieved by involving the target users. It was shown that the performance of an FDS can improve notably if its parameters are “personalized” (tuned or configured) after analyzing the particular characteristics of the mobility of the patient to be tracked [118]. In this sense, it is not clear if this configuration phase of the detection method may cause discomfort. If it does, users could be reluctant to accept detectors if they are obliged to participate in a configuration procedure during which a set of mobility samples must be generated to train the system.

## 7. Conclusions

Falls among the elderly constitute a serious medical and social concern. In this context, wearable Fall Detection Systems have been proposed as an economic and non-intrusive alternative to context-aware techniques aimed at deploying alerting systems capable of transmitting an alarm to a remote monitoring point whenever a fall is suspected.

This work has investigated the capability of a deep learning strategy (a Convolutional Neural Network) to discriminate falls from other ordinary movements (ADLs) based on the measurements of a gyroscope and an accelerometer. CNNs avoid the complex pre-processing pipeline and the need of manually engineering input features of other artificial intelligence techniques. Thus, the deep learning architecture is able to directly extract from the raw data captured by the sensors those features that maximize the capability of the classifier to discriminate the movements.

By using a long and well-known public repository containing traces of falls and ADLs, the CNN is carefully tailored and hyper-parameterized to optimize the classification performance when it is fed with signals from both sensors. However, even in that case, the performed experiments show that the configured CNN obtains better results when the gyroscope measurements are ignored and the accelerometry signals are the only parameters used as input features to train and test the convolutional neural classifier.

In spite of the fact that gyroscopes have been employed in several studies on wearable FDS, the particular benefits of combining both sensors have not usually been specifically analyzed by most studies. Our achieved results imply that a deep learning classifier can properly characterize the mobility of the user just focusing on the raw signals provided by the accelerometer, without needing the extra-costs (energy, hardware complexity, processing) of employing the information of the gyroscope. In addition, by considering the acceleration components exclusively, the dimension of the input features of the detection algorithm can be reduced, which eases the ability of the detector to operate in real time (a key factor when the fall detection system is deployed on a wearables with limited computing resources). In any case, future studies should confirm this conclusion by using other public datasets.

**Author Contributions:** E.C. proposed the experimental setup, defined the architecture and the evaluation procedure, co-analyzed the results, elaborated the critical review, wrote the paper; M.Á.-M. programmed the classifier and executed the tests to evaluate the algorithms; F.G.-L. proposed the neural architecture, co-discussed the results, revised the paper. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by FEDER Funds (under grant UMA18-FEDERJA-022) and Universidad de Málaga, Campus de Excelencia Internacional Andalucía Tech.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. World Health Organization. Ageing & Life Course Unit. In *WHO Global Report on Falls Prevention in Older Age*; World Health Organization: Geneva, Switzerland, 2008.
2. Fleming, J.; Brayne, C. Inability to get up after falling, subsequent time on floor, and summoning help: Prospective cohort study in people over 90. *BMJ* **2008**, *337*, 1279–1282. [[CrossRef](#)] [[PubMed](#)]
3. Wild, D.; Nayak, U.S.; Isaacs, B. How dangerous are falls in old people at home? *Br. Med. J. (Clin. Res. Ed.)* **1981**, *282*, 266–268. [[CrossRef](#)] [[PubMed](#)]
4. Van De Ven, P.; O'Brien, H.; Nelson, J.; Clifford, A. Unobtrusive monitoring and identification of fall accidents. *Med. Eng. Phys.* **2015**, *37*, 499–504. [[CrossRef](#)] [[PubMed](#)]
5. Florence, C.S.; Bergen, G.; Atherly, A.; Burns, E.; Stevens, J.; Drake, C. Medical Costs of Fatal and Nonfatal Falls in Older Adults. *J. Am. Geriatr. Soc.* **2018**, *66*, 693–698. [[CrossRef](#)] [[PubMed](#)]
6. Mubashir, M.; Shao, L.; Seed, L. A survey on fall detection: Principles and approaches. *Neurocomputing* **2013**, *100*, 144–152. [[CrossRef](#)]
7. Igual, R.; Medrano, C.; Plaza, I. Challenges, issues and trends in fall detection systems. *Biomed. Eng. Online* **2013**, *12*, 66. [[CrossRef](#)]
8. Chaccour, K.; Darazi, R.; El Hassani, A.H.; Andres, E. From Fall Detection to Fall Prevention: A Generic Classification of Fall-Related Systems. *IEEE Sens. J.* **2017**, *17*, 812–822. [[CrossRef](#)]
9. Zhang, D.; Wang, H.; Wang, Y.; Ma, J. Anti-Fall: A Non-intrusive and Real-time Fall Detector Leveraging CSI from Commodity WiFi Devices. In Proceedings of the International Conference on Smart Homes and Health Telematics (ICOST'2015), Geneva, Switzerland, 10–12 June 2015; Volume 9102, pp. 181–193.
10. Casilari, E.; Luque, R.; Morón, M. Analysis of android device-based solutions for fall detection. *Sensors* **2015**, *15*, 17827–17894. [[CrossRef](#)]
11. Aziz, O.; Musngi, M.; Park, E.J.; Mori, G.; Robinovitch, S.N. A comparison of accuracy of fall detection algorithms (threshold-based vs. machine learning) using waist-mounted tri-axial accelerometer signals from a comprehensive set of falls and non-fall trials. *Med. Biol. Eng. Comput.* **2017**, *55*, 45–55. [[CrossRef](#)]
12. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)]
13. Ordóñez, F.; Roggen, D.; Ordóñez, F.J.; Roggen, D. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors* **2016**, *16*, 115. [[CrossRef](#)] [[PubMed](#)]
14. Hammerla, N.Y.; Halloran, S.; Plötz, T. Deep, convolutional, and recurrent models for human activity recognition using wearable. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (AAAI), New York, NY, USA, 9–15 July 2016; pp. 1533–1540.
15. He, J.; Zhang, Z.; Wang, X.; Yang, S. A low power fall sensing technology based on fd-cnn. *IEEE Sens. J.* **2019**, *19*, 5110–5118. [[CrossRef](#)]
16. Sucerquia, A.; López, J.D.; Vargas-bonilla, J.F. SisFall: A Fall and Movement Dataset. *Sensors* **2017**, *17*, 198. [[CrossRef](#)] [[PubMed](#)]
17. SisFall Dataset (SISTEMIC Research Group, University of Antioquia, Colombia). Available online: <http://sistemic.udea.edu.co/en/investigacion/proyectos/english-falls/> (accessed on 12 February 2020).
18. Casilari-Pérez, E.; García-Lagos, F. A comprehensive study on the use of artificial neural networks in wearable fall detection systems. *Expert Syst. Appl.* **2019**, *138*, 112811. [[CrossRef](#)]
19. Pierleoni, P.; Belli, A.; Palma, L.; Pellegrini, M.; Pernini, L.; Valenti, S. A High Reliability Wearable Device for Elderly Fall Detection. *IEEE Sens. J.* **2015**, *15*, 4544–4553. [[CrossRef](#)]
20. Tomkun, J.; Nguyen, B. Design of a Fall Detection and Prevention System for the Elderly. Master's Thesis, McMaster University, Hamilton, ON, Canada, 2010.
21. Ge, W.; Shuwan, X. Portable Preimpact Fall Detector with Inertial Sensors. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2008**, *16*, 178–183. [[CrossRef](#)] [[PubMed](#)]
22. Lee, J.K.; Robinovitch, S.N.; Park, E.J. Inertial Sensing-Based Pre-Impact Detection of Falls Involving Near-Fall Scenarios. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2015**, *23*, 258–266. [[CrossRef](#)]
23. Felisberto, F.; Fdez-Riverola, F.; Pereira, A. A ubiquitous and low-cost solution for movement monitoring and accident detection based on sensor fusion. *Sensors* **2014**, *14*, 8961–8983. [[CrossRef](#)]
24. Mao, A.; Ma, X.; He, Y.; Luo, J. Highly portable, sensor-based system for human fall monitoring. *Sensors* **2017**, *17*, 2096. [[CrossRef](#)]



25. Chang, S.-Y.; Lai, C.-F.; Chao, H.-C.J.; Park, J.H.; Huang, Y.-M. An environmental-adaptive fall detection system on mobile device. *J. Med. Syst.* **2011**, *35*, 1299–1312. [[CrossRef](#)]
26. Dai, J.; Bai, X.; Yang, Z.; Shen, Z.; Xuan, D. Mobile phone-based pervasive fall detection. *Pers. Ubiquitous Comput.* **2010**, *14*, 633–643. [[CrossRef](#)]
27. Nyan, M.N.; Tay, F.E.H.; Tan, A.W.Y.; Seah, K.H.W. Distinguishing fall activities from normal activities by angular rate characteristics and high-speed camera characterization. *Med. Eng. Phys.* **2006**, *28*, 842–849. [[CrossRef](#)] [[PubMed](#)]
28. Kepski, M.; Kwolek, B.; Austvoll, I. Fuzzy inference-based reliable fall detection using kinect and accelerometer. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7267 LNAI, pp. 266–273, ISBN 9783642293467.
29. Ando, B.; Baglio, S.; Lombardo, C.O.; Marletta, V. A multisensor data-fusion approach for ADL and fall classification. *IEEE Trans. Instrum. Meas.* **2016**, *65*, 1960–1967. [[CrossRef](#)]
30. Astriani, M.S.; Heryadi, Y.; Kusuma, G.P.; Abdurachman, E. Human fall detection using accelerometer and gyroscope sensors in unconstrained smartphone positions. *Int. J. Recent Technol. Eng.* **2019**, *8*, 69–75.
31. Baek, W.S.; Kim, D.M.; Bashir, F.; Pyun, J.Y. Real life applicable fall detection system based on wireless body area network. In Proceedings of the 2013 IEEE 10th Consumer Communications and Networking Conference (CCNC 2013), Las Vegas, NV, USA, 11–14 January 2013; pp. 62–67.
32. Bourke, A.K.; O'Donovan, K.J.; O'laighin, G. The identification of vertical velocity profiles using an inertial sensor to investigate pre-impact detection of falls. *Med. Eng. Phys.* **2008**, *30*, 937–946. [[CrossRef](#)]
33. Bourke, A.K.; Lyons, G.M. A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. *Med. Eng. Phys.* **2008**, *30*, 84–90. [[CrossRef](#)]
34. Boutellaa, E.; Kerdjidj, O.; Ghanem, K. Covariance matrix based fall detection from multiple wearable sensors. *J. Biomed. Inform.* **2019**, *94*, 103189. [[CrossRef](#)]
35. Chen, G.C.; Huang, C.N.; Chiang, C.Y.; Hsieh, C.J.; Chan, C.T. A reliable fall detection system based on wearable sensor and signal magnitude area for elderly residents. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6159 LNCS, pp. 267–270. ISBN 3642137776.
36. Chernbumroong, S.; Cang, S.; Yu, H. Genetic algorithm-based classifiers fusion for multisensor activity recognition of elderly people. *IEEE J. Biomed. Health Inform.* **2015**, *19*, 282–289. [[CrossRef](#)]
37. Choi, Y.; Ralhan, A.S.; Ko, S. A study on machine learning algorithms for fall detection and movement classification. In Proceedings of the 2011 International Conference on Information Science and Applications (ICISA 2011), Jeju Island, South Korea, 26–29 April 2011.
38. Dau, H.A.; Salim, F.D.; Song, A.; Hedin, L.; Hamilton, M. Phone based fall detection by genetic programming. In Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia (MUM), Melbourne, Australia, 25–27 November 2014; pp. 256–257.
39. De Cillis, F.; De Simio, F.; Guido, F.; Incalzi, R.A.; Setola, R. Fall-detection solution for mobile platforms using accelerometer and gyroscope data. In Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), Milan, Italy, 25–29 August 2015; Institute of Electrical and Electronics Engineers Inc.: Milan, Italy, 2015; pp. 3727–3730.
40. Dinh, T.A.; Chew, M.T. Application of a commodity smartphone for fall detection. In Proceedings of the 6th International Conference on Automation, Robotics and Applications (ICARA 2015), Queenstown, New Zealand, 17–19 February 2015; IEEE: Queenstown, New Zealand, 2015; pp. 495–500.
41. Dinh, A.; Teng, D.; Chen, L.; Shi, Y.; McCrosky, C.; Basran, J.; Del Bello-Hass, V. Implementation of a physical activity monitoring system for the elderly people with built-in vital sign and fall detection. In Proceedings of the 6th International Conference on Information Technology: New Generations (ITNG 2009), Las Vegas, NV, USA, 27–29 April 2009; pp. 1226–1231.
42. Dinh, A.; Shi, Y.; Teng, D.; Ralhan, A.; Chen, L.; Dal Bello-Haas, V.; Basran, J.; Ko, S.-B.; McCrosky, C. A fall and near-fall assessment and evaluation system. *Open Biomed. Eng. J.* **2009**, *3*, 1–7. [[CrossRef](#)]
43. Dzung, R.J.; Fang, Y.C.; Chen, I.C. A feasibility study of using smartphone built-in accelerometers to detect fall portents. *Autom. Constr.* **2014**, *38*, 74–86. [[CrossRef](#)]
44. Figueiredo, I.N.; Leal, C.; Pinto, L.; Bolito, J.; Lemos, A. Exploring smartphone sensors for fall detection. *mUX J. Mob. User Exp.* **2016**, *5*, 2. [[CrossRef](#)]

45. Guo, H.W.; Hsieh, Y.T.; Huang, Y.S.; Chien, J.C.; Haraikawa, K.; Shieh, J.S. A threshold-based algorithm of fall detection using a wearable device with tri-axial accelerometer and gyroscope. In Proceedings of the International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS 2015), Okinawa, Japan, 28–30 November 2015; IEEE: Okinawa, Japan, 2015; pp. 54–57.
46. Hakim, A.; Huq, M.S.; Shanta, S.; Ibrahim, B.S.K.K. Smartphone Based Data Mining for Fall Detection: Analysis and Design. *Procedia Comput. Sci.* **2017**, *105*, 46–51. [[CrossRef](#)]
47. He, J.; Hu, C.; Wang, X. A Smart Device Enabled System for Autonomous Fall Detection and Alert. *Int. J. Distrib. Sens. Networks* **2016**, *12*, 2308183. [[CrossRef](#)]
48. He, Y.; Li, Y. Physical Activity Recognition Utilizing the Built-In Kinematic Sensors of a Smartphone. *Int. J. Distrib. Sens. Networks* **2013**, *9*, 481580. [[CrossRef](#)]
49. 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. Sensors* **2015**, *2015*, 1–8. [[CrossRef](#)]
50. Hwang, J.Y.; Kang, J.M.; Jang, Y.W.; Kim, H.C. Development of novel algorithm and real-time monitoring ambulatory system using Bluetooth module for fall detection in the elderly. In Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology, San Francisco, CA, USA, 1–5 September 2004; pp. 2204–2207.
51. Lai, C.F.; Chen, M.; Pan, J.S.; Youn, C.H.; Chao, H.C. A collaborative computing framework of cloud network and WBSN applied to fall detection and 3-D motion reconstruction. *IEEE J. Biomed. Health Inform.* **2014**, *18*, 457–466. [[CrossRef](#)] [[PubMed](#)]
52. Li, Q.; Stankovic, J.A.; Hanson, M.A.; Barth, A.T.; Lach, J.; Zhou, G. Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information. In Proceedings of the 6th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2009), Berkeley, CA, USA, 3–5 June 2009; pp. 138–143.
53. Majumder, A.J.A.; Zerin, I.; Uddin, M.; Ahamed, S.I.; Smith, R.O. SmartPrediction: A real-time smartphone-based fall risk prediction and prevention system. In Proceedings of the 2013 Research in Adaptive and Convergent Systems (RACS 2013), Montreal, QC, Canada, 1–4 October 2013; ACM: Montreal, QC, Canada, 2013; pp. 434–439.
54. Majumder, A.J.A.; Rahman, F.; Zerin, I.; Ebel, W., Jr.; Ahamed, S.I. iPrevention: Towards a novel real-time smartphone-based fall prevention system. In Proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC 2013), Coimbra, Portugal, 18–22 March 2013; ACM: Coimbra, Portugal, 2013; pp. 513–518.
55. Martínez-Villaseñor, L.; Ponce, H.; Espinosa-Loera, R.A. Multimodal Database for Human Activity Recognition and Fall Detection. In Proceedings of the 12th International Conference on Ubiquitous Computing and Ambient Intelligence (UCAmI 2018), Punta Cana, Dominican Republic, 4–7 December 2018; Volume 2.
56. Nari, M.I.; Suprpto, S.S.; Kusumah, I.H.; Adiprawita, W. A simple design of wearable device for fall detection with accelerometer and gyroscope. In Proceedings of the 2016 International Symposium on Electronics and Smart Devices (ISESD 2016), Bandung, Indonesia, 29–30 November 2017; pp. 88–91.
57. Ntanasis, P.; Pippa, E.; Özdemir, A.T.; Barshan, B.; Megalooikonomou, V. Investigation of Sensor Placement for Accurate Fall Detection. In Proceedings of the International Conference on Wireless Mobile Communication and Healthcare (MobiHealth 2016), Milan, Italy, 14–16 November 2016; Springer: Milan, Italy, 2016; pp. 225–232.
58. Nyan, M.N.; Tay, F.E.H.; Murugasu, E. A wearable system for pre-impact fall detection. *J. Biomech.* **2008**, *41*, 3475–3481. [[CrossRef](#)] [[PubMed](#)]
59. Ojetola, O.; Gaura, E.I.; Brusey, J. Fall Detection with Wearable Sensors—Safe (Smart Fall Detection). In Proceedings of the 7th International Conference on Intelligent Environments, Nottingham, UK, 25–28 July 2011; pp. 318–321.
60. Özdemir, A.T. An analysis on sensor locations of the human body for wearable fall detection devices: Principles and practice. *Sensors* **2016**, *16*, 1161. [[CrossRef](#)] [[PubMed](#)]
61. Park, C.; Suh, J.W.; Cha, E.J.; Bae, H.D. Pedestrian navigation system with fall detection and energy expenditure calculation. In Proceedings of the IEEE Instrumentation and Measurement Technology Conference, Binjiang, China, 10–12 May 2011; pp. 841–844.
62. De Quadros, T.; Lazzaretti, A.E.; Schneider, F.K. A Movement Decomposition and Machine Learning-Based Fall Detection System Using Wrist Wearable Device. *IEEE Sens. J.* **2018**, *18*, 5082–5089. [[CrossRef](#)]

63. Rakhecha, S.; Hsu, K. Reliable and secure body fall detection algorithm in a wireless mesh network. In Proceedings of the 8th International Conference on Body Area Networks (BODYNETS 2013), Boston, MA, USA, 30 September–2 October 2013; ICST: Boston, MA, USA, 2013; pp. 420–426.
64. Rungnapakan, T.; Chintakovid, T.; Wuttidittachotti, P. Fall detection using accelerometer, gyroscope & impact force calculation on android smartphones. In Proceedings of the 4th International Conference on Human-Computer Interaction and User Experience in Indonesia (CHIuXiD '18), Yogyakarta, Indonesia, 23–29 March 2018; Association for Computing Machinery: Yogyakarta, Indonesia, 2018; pp. 49–53.
65. Santiago, J.; Cotto, E.; Jaimes, L.G.; Vergara-Laurens, I. Fall detection system for the elderly. In Proceedings of the 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 9–11 January 2017; Institute of Electrical and Electronics Engineers Inc.: Las Vegas, NV, USA, 2017.
66. Sorvala, A.; Alasaarela, E.; Sorvoja, H.; Myllyla, R. A two-threshold fall detection algorithm for reducing false alarms. In Proceedings of the 2012 6th International Symposium on Medical Information and Communication Technology (ISMICT 2012), La Jolla, CA, USA, 25–29 March 2012.
67. Tamura, T.; Yoshimura, T.; Sekine, M.; Uchida, M.; Tanaka, O. A Wearable Airbag to Prevent Fall Injuries. *IEEE Trans. Inf. Technol. Biomed.* **2009**, *13*, 910–914. [[CrossRef](#)]
68. Wibisono, W.; Arifin, D.N.; Pratomo, B.A.; Ahmad, T.; Ijtihadie, R.M. Falls Detection and Notification System Using Tri-axial Accelerometer and Gyroscope Sensors of a Smartphone. In Proceedings of the 2013 Conference on Technologies and Applications of Artificial Intelligence (TAAI), Taipei, Taiwan, 6–8 December 2013; IEEE: Taipei, Taiwan, 2013; pp. 382–385.
69. Yang, B.-S.; Lee, Y.-T.; Lin, C.-W. On Developing a Real-Time Fall Detecting and Protecting System Using Mobile Device. In Proceedings of the International Conference on Fall Prevention and Protection (ICFPP 2013), Tokyo, Japan, 23–25 October 2013; pp. 151–156.
70. Zhao, G.; Mei, Z.; Liang, D.; Ivanov, K.; Guo, Y.; Wang, Y.; Wang, L. Exploration and Implementation of a Pre-Impact Fall Recognition Method Based on an Inertial Body Sensor Network. *Sensors* **2012**, *12*, 15338–15355. [[CrossRef](#)] [[PubMed](#)]
71. Ahmed, M.; Mehmood, N.; Nadeem, A.; Mehmood, A.; Rizwan, K. Fall detection system for the elderly based on the classification of shimmer sensor prototype data. *Healthc. Inform. Res.* **2017**, *23*, 147–158. [[CrossRef](#)] [[PubMed](#)]
72. Chelli, A.; Patzold, M. A Machine Learning Approach for Fall Detection and Daily Living Activity Recognition. *IEEE Access* **2019**, *7*, 38670–38687. [[CrossRef](#)]
73. Ghazal, M.; Khalil, Y.A.; Dehbozorgi, F.J.; Alhalabi, M.T. An integrated caregiver-focused mHealth framework for elderly care. In Proceedings of the 2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Abu Dhabi, UAE, 19–21 October 2015; IEEE: Abu Dhabi, UAE, 2015; pp. 238–245.
74. Nukala, B.T.; Shibuya, N.; Rodriguez, A.I.; Tsay, J.; Nguyen, T.Q.; Zupancic, S.; Lie, D.Y.C. A real-time robust fall detection system using a wireless gait analysis sensor and an Artificial Neural Network. In Proceedings of the 2014 IEEE Healthcare Innovation Conference (HIC), Seattle, WA, USA, 8–10 October 2014; IEEE: Seattle, WA, USA, 2014; pp. 219–222.
75. Özdemir, A.T.; Barshan, B. Detecting Falls with Wearable Sensors Using Machine Learning Techniques. *Sensors* **2014**, *14*, 10691–10708. [[CrossRef](#)]
76. Rashidpour, M.; Abdali-Mohammadi, F.; Fathi, A. Fall detection using adaptive neuro-fuzzy inference system. *Int. J. Multimed. Ubiquitous Eng.* **2016**, *11*, 91–106.
77. Wang, S.; Zhang, X. An approach for fall detection of older population based on multi-sensor data fusion. In Proceedings of the 2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Goyang, South Korea, 28–30 October 2015; IEEE: Goyang, South Korea, 2015; pp. 320–323.
78. Yang, S.-H.; Zhang, W.; Wang, Y.; Tomizuka, M. Fall-prediction algorithm using a neural network for safety enhancement of elderly. In Proceedings of the 2013 CACS International Automatic Control Conference (CACS), Nantou, Taiwan, 2–4 December 2013; IEEE: Nantou, Taiwan, 2013; pp. 245–249.
79. Yodpijit, N.; Sittiwanchai, T.; Jongprasithporn, M. The development of Artificial Neural Networks (ANN) for falls detection. In Proceedings of the 2017 3rd International Conference on Control, Automation and Robotics (ICCAR), Nagoya, Japan, 24–26 April 2017; IEEE: Nagoya, Japan, 2017; pp. 547–550.
80. Bianchi, F.; Redmond, S.J.; Narayanan, M.R.; Cerutti, S.; Lovell, N.H. Barometric pressure and triaxial accelerometry-based falls event detection. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2010**, *18*, 619–627. [[CrossRef](#)]

81. Perry, J.T.; Kellog, S.; Vaidya, S.M.; Youn, J.-H.; Ali, H.; Sharif, H. Survey and evaluation of real-time fall detection approaches. In Proceedings of the 6th International Symposium on High-Capacity Optical Networks and Enabling Technologies (HONET 2009), Alexandria, Egypt, 28–30 December 2009; IEEE: Alexandria, Egypt, 2009; pp. 158–164.
82. Becker, C.; Schwickert, L.; Mellone, S.; Bagalà, F.; Chiari, L.; Helbostad, J.L.; Zijlstra, W.; Aminian, K.; Bourke, A.; Todd, C.; et al. Proposal for a multiphase fall model based on real-world fall recordings with body-fixed sens. *Z. Gerontol. Geriatr.* **2012**, *45*, 707–715. [[CrossRef](#)]
83. Viet, V.Q.; Lee, G.; Choi, D. Fall Detection Based on Movement and Smart Phone Technology. In Proceedings of the IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF 2012), Ho Chi Minh City, Vietnam, 27 February–1 March 2012; IEEE: Ho Chi Minh City, Vietnam, 2012; pp. 1–4.
84. Tsinganos, P.; Skodras, A. On the Comparison of Wearable Sensor Data Fusion to a Single Sensor Machine Learning Technique in Fall Detection. *Sensors* **2018**, *18*, 592. [[CrossRef](#)]
85. Abbate, S.; Avvenuti, M.; Corsini, P.; Light, J.; Vecchio, A. Monitoring of Human Movements for Fall Detection and Activities Recognition in Elderly Care Using Wireless Sensor Network: A survey. *Wireless Sensor Networks: Application-Centric Design*; Tan, Y.K., Merrett, G., Eds.; InTech, 2010. Available online: <https://www.intechopen.com/books/wireless-sensor-networks-application-centric-design/monitoring-of-human-movements-for-fall-detection-and-activities-recognition-in-elderly-care-using-wi> (accessed on 1 April 2020). [[CrossRef](#)]
86. Lorincz, K.; Chen, B.; Challen, G.W.; Chowdhury, A.R.; Patel, S.; Bonato, P.; Welsh, M. Mercury: A Wearable Sensor Network Platform for High-Fidelity Motion Analysis. In Proceedings of the 7th International Conference on Embedded Networked Sensor Systems (SenSys 2009), Berkeley, CA, USA, 4–6 November 2009.
87. Nguyen Gia, T.; Sarker, V.K.; Tcareno, I.; Rahmani, A.M.; Westerlund, T.; Liljeberg, P.; Tenhunen, H. Energy efficient wearable sensor node for IoT-based fall detection systems. *Microprocess. Microsyst.* **2018**, *56*, 34–46. [[CrossRef](#)]
88. Kangas, M.; Vikman, I.; Nyberg, L.; Korpelainen, R.; Lindblom, J.; Jämsä, T. Comparison of real-life accidental falls in older people with experimental falls in middle-aged test subjects. *Gait Posture* **2012**, *35*, 500–505. [[CrossRef](#)]
89. Klenk, J.; Becker, C.; Lieken, F.; Nicolai, S.; Maetzler, W.; Alt, W.; Zijlstra, W.; Hausdorff, J.M.; Van Lummel, R.C.; Chiari, L. Comparison of acceleration signals of simulated and real-world backward falls. *Med. Eng. Phys.* **2011**, *33*, 368–373. [[CrossRef](#)] [[PubMed](#)]
90. Jämsä, T.; Kangas, M.; Vikman, I.; Nyberg, L.; Korpelainen, R. Fall detection in the older people: From laboratory to real-life. *Proc. Est. Acad. Sci.* **2014**, *63*, 341–345. [[CrossRef](#)]
91. Casilari, E.; Santoyo-Ramón, J.A.; Cano-García, J.M. Analysis of public datasets for wearable fall detection systems. *Sensors* **2017**, *17*, 1513. [[CrossRef](#)] [[PubMed](#)]
92. Frank, K.; Vera Nadales, M.J.; Robertson, P.; Pfeifer, T. Bayesian recognition of motion related activities with inertial sensors. In Proceedings of the 12th ACM International Conference on Ubiquitous Computing (ACM), Copenhagen, Denmark, 26–29 September 2010; pp. 445–446.
93. Ojetola, O.; Gaura, E.; Brusey, J. Data Set for Fall Events and Daily Activities from Inertial Sensors. In Proceedings of the 6th ACM Multimedia Systems Conference (MMSys'15), Portland, OR, USA, 18–20 March 2015; pp. 243–248.
94. Vavoulas, G.; Padiaditis, M.; Spanakis, E.G.; Tsiknakis, M. The MobiFall dataset: An initial evaluation of fall detection algorithms using smartphones. In Proceedings of the IEEE 13th International Conference on Bioinformatics and Bioengineering (BIBE 2013), Chania, Greece, 10–13 November 2013; pp. 1–4.
95. Vavoulas, G.; Chatzaki, C.; Malliotakis, T.; Padiaditis, M. The Mobiact dataset: Recognition of Activities of Daily living using Smartphones. In Proceedings of the International Conference on Information and Communication Technologies for Ageing Well and e-Health (ICT4AWE), Rome, Italy, 21–22 April 2016.
96. Gasparrini, S.; Cippitelli, E.; Spinsante, S.; Gambi, E. A depth-based fall detection system using a Kinect@sensor. *Sensors* **2014**, *14*, 2756–2775. [[CrossRef](#)]

97. Vilarinho, T.; Farshchian, B.; Bajer, D.G.; Dahl, O.H.; Egge, I.; Hegdal, S.S.; Lones, A.; Slettevold, J.N.; Weggersen, S.M. A Combined Smartphone and Smartwatch Fall Detection System. In Proceedings of the 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), Liverpool, UK, 26–28 October 2015; pp. 1443–1448.
98. Medrano, C.; Igual, R.; Plaza, I.; Castro, M. Detecting falls as novelties in acceleration patterns acquired with smartphones. *PLoS ONE* **2014**, *9*, e94811. [[CrossRef](#)]
99. Kwolek, B.; Kepski, M. Human fall detection on embedded platform using depth maps and wireless accelerometer. *Comput. Methods Programs Biomed.* **2014**, *117*, 489–501. [[CrossRef](#)]
100. Casilari, E.; Santoyo-Ramón, J.A.; Cano-García, J.M. Analysis of a Smartphone-Based Architecture with Multiple Mobility Sensors for Fall Detection. *PLoS ONE* **2016**, *11*, e01680. [[CrossRef](#)]
101. Micucci, D.; Mobilio, M.; Napoletano, P. UniMiB SHAR: A new dataset for human activity recognition using acceleration data from smartphones. *Appl. Sci.* **2017**, *7*, 1101. [[CrossRef](#)]
102. Wertner, A.; Czech, P.; Pammer-Schindler, V. An Open Labelled Dataset for Mobile Phone Sensing Based Fall Detection. In Proceedings of the 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MOBIQUITOUS 2015), Coimbra, Portugal, 22–24 July 2015; pp. 277–278.
103. Nguyen, L.P.; Saleh, M.; Le Bouquin Jeannès, R. An Efficient Design of a Machine Learning-Based Elderly Fall Detector. In Proceedings of the International Conference on IoT Technologies for HealthCare (HealthyIoT 2017), Angers, France, 24–25 October 2017; Springer: Angers, France, 2017; pp. 34–41.
104. Hsieh, C.-Y.; Liu, K.-C.; Huang, C.-N.; Chu, W.-C.; Chan, C.-T. Novel Hierarchical Fall Detection Algorithm Using a Multiphase Fall Model. *Sensors* **2017**, *17*, 307. [[CrossRef](#)]
105. Yu, X. Approaches and principles of fall detection for elderly and patient. In Proceedings of the 10th International Conference on e-health Networking, Applications and Services (HealthCom 2008), Singapore, 7–9 July 2008; pp. 42–47.
106. Igual, R.; Medrano, C.; Plaza, I. A comparison of public datasets for acceleration-based fall detection. *Med. Eng. Phys.* **2015**, *37*, 870–878. [[CrossRef](#)]
107. Micucci, D.; Mobilio, M.; Napoletano, P.; Tisato, F. Falls as anomalies? An experimental evaluation using smartphone accelerometer data. *J. Ambient Intell. Humaniz. Comput.* **2017**, *8*, 87–99. [[CrossRef](#)]
108. Davis, T.; Sigmon, K. MATLAB Primer, Seventh Edition. Available online: <http://www.mathworks.com/products/matlab/> (accessed on 25 July 2019).
109. Carletti, V.; Greco, A.; Saggese, A.; Vento, M. A Smartphone-Based System for Detecting Falls Using Anomaly Detection. In Proceedings of the 19th International Conference on Image Analysis and Processing (ICIAP 2017), Catania, Italy, 11–15 September 2017; pp. 490–499.
110. Mastorakis, G. Human Fall Detection Methodologies: From Machine Learning Using Acted Data to Fall Modelling Using Myoskeletal Simulation. Ph.D. Thesis, Kingston University, London, UK, 2018.
111. Putra, I.P.E.S.; Brusey, J.; Gaura, E.; Vesilo, R. An Event-Triggered Machine Learning Approach for Accelerometer-Based Fall Detection. *Sensors* **2017**, *18*, 20. [[CrossRef](#)] [[PubMed](#)]
112. Mauldin, T.R.; Canby, M.E.; Metsis, V.; Ngu, A.H.H.; Rivera, C.C. SmartFall: A Smartwatch-Based Fall Detection System Using Deep Learning. *Sensors* **2018**, *18*, 3363. [[CrossRef](#)] [[PubMed](#)]
113. Lipsitz, L.A.; Tchalla, A.E.; Iloputaife, I.; Gagnon, M.; Dole, K.; Su, Z.Z.; Klickstein, L. Evaluation of an Automated Falls Detection Device in Nursing Home Residents. *J. Am. Geriatr. Soc.* **2016**, *64*, 365–368. [[CrossRef](#)] [[PubMed](#)]
114. Thilo, F.J.S.; Hürlimann, B.; Hahn, S.; Bilger, S.; Schols, J.M.G.A.; Halfens, R.J.G. Involvement of older people in the development of fall detection systems: A scoping review. *BMC Geriatr.* **2016**, *16*, 1–9. [[CrossRef](#)]
115. Hawley-Hague, H.; Boulton, E.; Hall, A.; Pfeiffer, K.; Todd, C. Older adults' perceptions of technologies aimed at falls prevention, detection or monitoring: A systematic review. *Int. J. Med. Inform.* **2014**, *83*, 416–426. [[CrossRef](#)]
116. Godfrey, A. Wearables for independent living in older adults: Gait and falls. *Maturitas* **2017**, *100*, 16–26. [[CrossRef](#)]

117. Thilo, F.J.S.; Hahn, S.; Halfens, R.J.G.; Schols, J.M.G.A. Usability of a wearable fall detection prototype from the perspective of older people - a real field testing approach. *J. Clin. Nurs.* **2019**, *28*, 310–320. [[CrossRef](#)]
118. Medrano, C.; Plaza, I.; Igual, R.; Sánchez, Á.; Castro, M. The Effect of Personalization on Smartphone-Based Fall Detectors. *Sensors* **2016**, *16*, 117. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).