Autonomous on-wrist acceleration-based fall detection systems: unsolved challenges

José R. Villar[a,*], Camelia Chira[b], Enrique de la Cal[a], Víctor M. González[c], Javier Sedano[d], Samad B. Khojasteh[e]

[a] *Computer Science Department, University of Oviedo, Oviedo, Spain*
[b] *Computer Science Department, Babes-Bolyai University, Romania*
[c] *Automatica Department, University of Oviedo, Spain*
[d] *Instituto Tecnológico de Castilla y León, Burgos, Spain*
[e] *Department of Computer Engineering, Selcuk University, Konya, Turkey*

Abstract

Fall detection (FD) has been the focus of many research studies during the last years. Developing reliable FD systems is relevant, for instance, to provide support to the elderly population in their everyday life. Besides, the generalization of the use of wearable devices (and more specifically, on-wrist devices) to measure the daily activity strongly suggests that in a short period of time, the elderly people will be making use of this type of devices. On-wrist devices can be used as the FD basic sensing unit; while the intelligent classification can be obtained either autonomously (on the device) or requested to a remote service (via the paired smartphone or via web services). This study tries to analyze the current challenges in autonomous on-wrist wearable devices for producing a reliable and robust FD system. To do so, we analyze the related work; one of the possible solutions is implemented with several alternatives and evaluated with publicly available simulated falls data sets. The most remarkable findings in this research are that i) real fall data sets are needed, at least, a valid merging method to produce real fall like Time Series, ii) generalized solutions might not be enough and research is needed in models that learns from the user, iii) the need of tuning and fitting to the

*Corresponding author

Email addresses: villarjose@uniovi.es (José R. Villar), cchira@cs.ubbcluj.ro (Camelia Chira), delacal@uniovi.es (Enrique de la Cal), vmsuarez@uniovi.es (Víctor M. González), javier.sedano@uniovi.es (Javier Sedano), samad.khojasteh@lisansustu.selcuk.edu.tr (Samad B. Khojasteh)

current user performance, iv) the amount of fall types suggests that hybrid and ensemble approaches might be interesting.

Keywords: Fall Detection, Machine Learning, Elderly Population

1. Introduction

Fall Detection (FD) refers to the detection of fall events of human beings while performing their usual Activities of Daily Living (ADL); it might be considered as part of the Human Activity Recognition (HAR). FD can be applied in several different fields, the support for the elderly population [1] among them. In case of the elder, a fall might be due to many different issues (accidentally trip over an obstacle, a health problem, etc.), like any other subpopulation. But in the case of the elder, it could be necessary to provide help and assistance; the faster the assistance, the lower the consequences in the normal life of the affected person. Therefore, correct fall detection can be considered of major importance in the case of elderly people.

FD has been performed either using video image analysis or using wearable devices. This study focuses on to this latter option; more specifically, this research is focused autonomous on-wrist wearable devices, where the whole computation is performed on the device.

Basically, the FD methods so far developed include an event detection stage (responsible of extracting the corresponding set of features) followed by a Machine Learning (ML) stage to generate the classification model. Alternatively, some methods directly apply an ML method to each sliding window without any event detection. The most commonly used sensor in FD is the tri-axial accelerometers (3DACC), used independently or combined with other sensors (such as the inertial sensors it belongs to, barometer, etc.). The sensory system is located mainly on the waist or on the wrist, and in some cases on the thigh.

In most of the published studies, the solutions made use of data gathered from simulated falls in order to train, test and validate the different solutions; the data might be publicly available source or might be a private data collection. The publicly available data sets are gathered from participants performing a set of ADL; the participants are usually members of the different research groups but volunteers are included as well. The fall events are simulated through participants falling over a mattress from a standing still posture.

Several commercial devices have been deployed into the market, some of them from manufacturers that are leaders in the mobility market. Still today, the user plays an important role in the performance of these FD devices as a fall is just a detected event that the user does not reset. Besides, the main public may consider the problem is solved once those big companies introduce their solutions into the market.

Nevertheless, the FD problem should not be considered solved according to the reported results in both the research and the applied fields. There are several issues, particularly for the elderly population, that must be addressed before considering FD satisfactorily solved. These issues include the definition of a fall event, the quality and representativeness of the available data sets and the criteria to evaluate the performance of a solution. The current study addresses these concerns through an in-depth analysis of the related work (included in the next Section), a complete experimentation to evaluate the reliability of current solutions (see Sections 3 for its description and 4 for the showing and discussing the results). The study ends with the conclusions and some ideas for developing reliable FD systems.

2. Related work on FD using Wearable devices

FD using wearable devices has been studied for more than a decade now. Several reviews have been published and are available for an in-depth reading [2, 3, 4, 5, 6]. From now on, the following acronyms apply: Finite State Machine (FSM), threshold (TH), Neural Network (NN), Rule set (RS), K-Nearest Neighbor (kNN), Decision Trees (DT), Discriminant Analysis (DA), Support Vector Machine (SVM), One-class SVM (OSVM), Classification and Regression Trees (CART), Logistic Regression (LR). The related work is included in the next paragraphs.

The most common sensor used for FD is the 3DACC. For instance, Zhang et al ([7]) used a 3DACC sensor placed on a belt of the subjects while performing several ADLs, some participants were elderly people; they also used a dummy to simulate the falls. An OSVM model classifies the time window as normal, signaling the remaining cases as fall alarms. Two 3DACC (one placed on the chest and one on the thigh) are used in [8]. A set of thresholds are used to determine if a signal belongs or not to a fall event. This solution was compared with other related studies in [9]. Recently, the authors [10] analyzed the real fall data set from patients of Parkinson [11], where the patients wore a 3DACC plus a gyroscope on either the waist or the thigh.

The event detection was performed through a threshold and several features were calculated for each detected event. The generated data set was balanced using SMOTE and a C4.5 decision tree classifier was proposed.

Several different measurements of the fall dynamics were used in [12] with data gathered from a 3DACC on the waist (or head). Then, several sequences of these measurements surpassing predefined thresholds were used as the algorithm for fall detection. Using this solution, the authors compared the dynamic of real and simulated fall events [13].

Moreover, 3DACC has been also combined with barometric sensors to detect fall events in [14]. The sensor was located on the waist and an heuristic set of rules and thresholds were proposed to determine whether there is a fall or not. Besides, Sorvala et al ([15]) combined 3DACC on the waist and a gyroscope on the ankle, using the magnitude of the acceleration and the angular velocity together with an heuristic algorithm based on thresholds to classify the signals. Similarly, 3DACC was also combined with gyroscope in [16], where a study of the performance of several thresholds based FD methods were analyzed when run on a Smartwatch and on a Smartphone. The threshold based methods were also used to change the current state, similar to a FSM. The same combination of sensors but placed on the chest are used in [17] for FD. The decision is based on three thresholds: if a small acceleration magnitude is followed by a high acceleration magnitude and a high angular velocity, then a fall is alarmed.

In [18] combined 3DACC and barometer sensors on a device; this device is placed on a wrist. The acceleration magnitude is used to detect the peak events; for each detected peak, 3 features are extracted from a 6 second length pressure window centered on the peak. These features are classified using a SVM. The work presented in [19] combined 3DACC together with gyroscope and barometer. In this study, the sensory system is located on the waist, the event detection is based on thresholds of the vertical velocity. Whenever an increase in this signal is observed, up to 7 seven different combinations of the acceleration, posture and height are surveyed; if any of these combinations is higher than the corresponding threshold, an alarm is signaled.

Using a single 3DACC on the waist, the study in [20] proposed a FD system based on an event detection plus a classifier. The event detection stage was a peak detection based on a FSM and predefined thresholds; then, a feature extraction is performed on the time slice surrounding the detected peaks. Finally, a NN is used to classify each instance of 8 transformations. Previously, the authors performed an in-depth analysis of the falls and their

dynamics, taxonomy and causes [21].

The solution of Abbate et al in [20] was extended in [22] and [23]. In the former, kNN was used instead of NN. In the study in [23], the approach was adapted to be used with the sensor on a wrist, several features were revised and, finally, different models were evaluated (NN, SVN, kNN, DT and RBS). The same research team analyzed the use of kNN with a reduced data set including selected instances from clustering [24]. In [25], the authors proposed a one-class SAX-based dictionary to learn the user behavior; these dictionaries were developed for each specific user considering only the ADLs.

Instead of a FSM, [26] proposed to detect high peaks, low peaks and the time between a sequence of a high and low consecutive peaks. They developed an Android Wear app to use the 3DACC measurements from a Smartwatch. FSM were also used as event detection in [27]; whenever a peak was detected, the surrounding window was analyzed and several transformations were computed. The classification of these features was performed using a classification and regression tree, a kNN, LR and a SVM.

An FSM is proposed to detect the fall events if the subject does not move after the fall [26]. Thresholds of the acceleration and the angle of the gravity are used together with the time in each state to drive the FSM. The 3DACC sensor is located on the waist in this study. Similarly, thresholds are used to detect fall events in [28]; if a fall event is followed by a 20 seconds calm period (that is, with a reduced amount of movement), then the fall alarm is signaled. Thresholds were also used for FD in [29], the sensor was the 3DACC signal from a Smartphone.

A comparison of several published simulated falls data set presented in [30] used a threshold on the acceleration magnitude to detect fall candidates; afterwards, 6 second windows are classified using either SVM or NN. NN have been also used in [31]. In this study, a 3DACC was located on a wrist and three different NN models were obtained: i) using 3 seconds of acceleration magnitude windows as inputs of the NN, ii) the acceleration magnitude peak and the times of the fall event as the input features of the NN and, iii) these three features plus the mean and deviation were the inputs of the NN. In all the cases, a Multi-layer Perceptron was proposed.

Furthermore, Medrano et al [32] analyzed the 3DACC signals from Smartphones placed inside the frontal pockets. They used three different models (kNN, SVM and OSVM). The inputs were the three acceleration components during the 6 second windows centered on the acceleration peak; the acceleration peaks are found whenever the acceleration is higher than a threshold.

Similarly, Ngu et al proposed to classify 250 milliseconds windows of the acceleration magnitude; these windows were transformed into a 4 dimensional vector and considered the inputs of the two modeling techniques (SVM and kNN).

Finally, Deep Learning is currently being employed in FD, although developing such models on wearable devices will need more powerful Smartwatches than the ones in the market nowadays. Nevertheless, the study in [33] proposed to pair the Smartwatch to a Smartphone, which is the responsible of running the Recurrent Neural Network. For a more in-depth review on this topic, please refer to [6]. However, because nowadays Deep Learning is not feasible to be deployed on wearable devices such as Smartwatches (as stated in [33]), we do not develop on this type of solutions further.

Several facts can be extracted from these studies. Firstly, the location of the sensor is related to which the final population is. Using belts can be related with people suffering from severe illnesses, such as Parkinson Disease, because in their case the service has higher priority than the personal image and aesthetics. However, when focusing on healthy subjects, priorities are not so clear; for this population, wrist based solution are more suitable because the inconspicuous character of these type of devices. Other type of combined sensors and locations could be useful in some specific scenarios, such as performing FD in factories.

Secondly, the majority of the methods include an event detection stage that usually relies on thresholds. These thresholds have been developed based on mechanical studies and the relationships between the features used and their corresponding span.

Thirdly, well-known machine learning methods have been applied in the FD classification task. This classification task maps the feature extraction domain when a relevant event occurs (i.e., a peak in the magnitude of the acceleration) in order to assign the corresponding label Fall or Not Fall. One interesting point in this concern is that solutions that produce high computational models might not be suitable because of the compromise with the battery consumption [4, 34].

Concerning the experimentation, almost all the studies make use of data sets of simulated falls, either private or publicly available, performing the evaluation of the proposals with different ADL coming from the available sources and participants. Rarely, some studies employed data from real falls, mainly people suffering an illness (i.e., Parkinson Disease or similar).

3. Materials and methods

The main goal in this study is to evaluate an up-to-date FD method in a scenario that might be similar to the real case of deploying a solution in the market. In this case, the training is performed using the available data and then it is deployed and evaluated with different participants with possibly different devices. Therefore, there are several issues to cover: i) the data set for training, testing and validation, ii) the FD method to evaluate, iii) the experimental design mimicking the real life. The next three subsections deal with each of the previous referred issues in the same order.

3.1. The collection of data sets

In a recent study [35] up to twelve publicly available data sets related with FD and ADL were compared; these data sets have the common characteristic of using 3DACC sensors located on different parts of the body. Recently, a new data set has been also published in [36]. As stated in [35], it was found a lack "of a common experimental bench-marking procedure and, consequently, the large heterogeneity of the data sets from a number of perspectives (length and number of samples, typology of the emulated falls and ADLs, characteristics of the test subjects, features and positions of the sensors, etc.)". It was also stated the relevance of suitable sensor ranges and a good analysis of the ADLs, grouping them according to the objective of the study -i.e, in order to the activity level of the ADLs-.

This research is restricted to on-wrist 3DACC wearable devices; therefore, from those data sets publicly available, only those containing data gathered from sensors located on the wrist are considered. There are up to five data sets including a sensor on the wrist (UMA Fall, TST, DaLiaC, Gravity Project and UNIOVI Simulated Epilepsy), all of them are summarized in Table 1. Unfortunately, we were not able to use the data from the Gravity Project because the data collection we got did not include the wrist worn sensory data; therefore, we have not included it in this study. Please refer to [35, 36] for information related to all the published data sets.

Considering the four remaining data sets, they sum up a total of 1414 Time Series (TS) instances. Each TS instance correspond to a participant performing and ADL or staged fall; the TS instance includes the 3DACC acceleration signals (that is, the components in each of the three axis). Up to 412 of these TS correspond to simulated falls using different sensors and

sampling frequencies, with different behavior performed by up to 55 participants. Each TS will be assigned with a label, either FALL or NOT_FALL, accordingly to the TS including a fall or not.

Dataset	**NP**	**NTS**	**NF**	**Fqcy**	**Description**
UMA Fall [37]	17	531	208	20	Includes forward, backward and lateral falls, running, hopping, walking and sitting. Neither all the participants have every type of activities nor the same number of goes. Sensors on the wrist, waist, ankle, chest and trouser pocket.
TST FD [38]	11	264	132	100	Includes forward, backward and lateral falls, with two sensors, one on the waist and one on the right wrist.
DaLiac [39]	19	247	0	204.8	Sitting, standing, lying, vacuuming, washing dishes, sweeping, walking, up and down stairs, using a treadmill, cycling and rope jumping. Sensors on the wrist, waist, ankle and chest.
UNIOVI [40]	6	275	0	16	Walking at different paces, running, sawing, and simulating epileptic partial tonic-clonic seizures Sensor on the dominant wrist.

Table 1: Descriptions of the different data sets used in this research. Columns NP, NTS and NF stand for the number of participants, the number of available TS, and the number of falls in the data set, respectively. The sampling frequency used in gathering the data set is stated in Hz in frequency column (Fqcy).

3.2. The FD method to analyze

The FD method proposed in [20] is one of the representative solutions within the state of the art; this solution has been extended in many research studies. The solution in [20] includes a complex event detection based on a finite state machine plus a modeling stage to classify each event as Fall or Not Fall. In this study, we choose the extension proposed in [23] with different modeling techniques, so a wide variety of published solutions are covered.

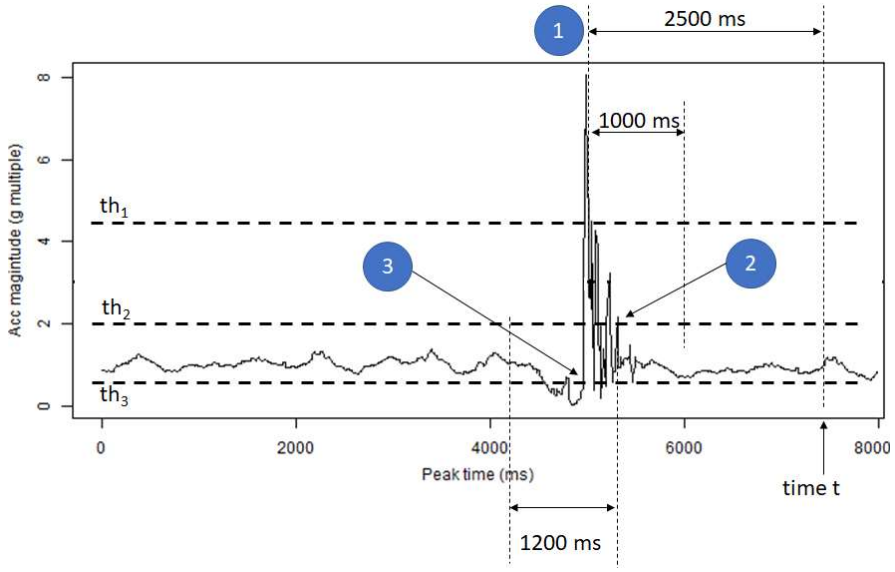

Figure 1: Dynamics of a fall measured through the acceleration magnitude when a sensor is located on the waist. Time 1 is the time stamp of the detected peak, Time 2 is the last activity higher than 1.5 g (the end of the fall event), Time 3 is the starting of the fall event.

Therefore, the basis is the detection of fall events when the participants use a 3DACC wearable device on a wrist. Concerning the event detection, the FSM proposed in [20, 21] is used. In this solution, the dynamics of a fall event are described as stated in Fig. 1.

3.2.1. Event detection

Let us assume that gravity be $g = 9.8m/s^2$. From a standing still position, a fall starts with a sudden sequence of changes in the magnitude of the acceleration: first it evolves below g and then it performs a peak several times the value of g to end with a period of time without relevant movements. Therefore, the aim is to detect these peaks measured during a fall event.

The feature extraction is executed whenever a peak is detected and follows the dynamics within a fall -refer to Fig. 1-. Given the current time-sstamp t, we find a peak at **peak time** $pt = t - 2500ms$ (point 1) if at time pt the magnitude of the acceleration a_t -see Equation 1- is higher than $th_1 = 3 \times g$ and there is no other peak in the period $(t - 2500ms, t]$ (no other a value higher than th_1). If this condition holds, then it is stated that a peak occurred at pt. A FSM is used to determine the current state of the fall event; for

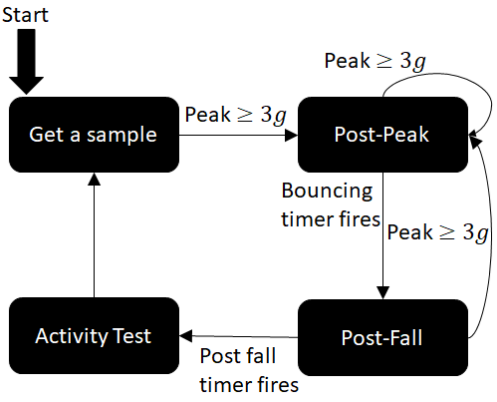

Figure 2: Finite State Machine proposed in [20]. Whenever a peak is detected, the state moves to Post-Peak. After the bouncer timer fires, the state moves from Post-Peak to Post-Fall. When the post-fall timer fires, the state moves to Activity Test. In this state, the features are computed and the classification takes place.

more details, please refer to [20, 23]; the FSM is also outlined in Fig. 2.

From now on and without losing generalization, as long as we know the sampling frequency, we can refer to time-stamp or to positions within a sliding window that includes the samples in $[is, ie]$ (where is and ie stand for **impact start** and **impact end**, correspondingly, the limits of the impact window; see next subsection for details on how to determine these values). When using sub-index i we refer to the sample position within the sliding window, when using sub-index t we refer to a time-stamp; however, they are interchangeable because we are using a constant sampling frequency.

$$a_t = \sqrt{a_{tx}^2 + a_{ty}^2 + a_{tz}^2} \tag{1}$$

It is worth mentioning that setting thresholds always becomes a compromise because their values may rely on the subject's behavior. Actually, this is one of the main drawbacks of the proposal of Abbate et al. In [23], a genetic algorithm was use to drive an optimization of the peak threshold, improving the performance of the FSM. Nevertheless, further study is needed to avoid the use of thresholds as much as possible.

3.2.2. Feature extraction

When a peak is detected, the feature extraction is performed computing for this peak time several parameters and features. The ie (point 2) denotes

the end of the fall event; it is the last time for which the a value is higher than $th_2 = 1.5 \times g$. Finally, the is (point 3) denotes the starting time of the fall event, computed as the time of the first sequence of an $a <= th_3$ ($th_3 = 0.8 \times g$) followed by a value of $a >= th_2$. The impact start must belong to the interval $[ie - 1200\ ms, pt]$. If no impact end is found, then it is fixed to $pt + 1000$ ms. If no impact start is found, it is fixed to pt.

With these three times -is, pt, ie- calculated, the following transformations should be computed. These features were designed following the dynamic of a fall in [20], and slightly modified in [23]. Again, for more detail on the features, please refer to those studies.

- Average Absolute Acceleration Magnitude Variation (AAMV), calculated as the sum of the absolute value of the differences between consecutive samples of the acceleration magnitude within the interval $[is, ie]$, divided by the total number of samples in the interval. This feature measures whether the user is moving or staying still, that is, measures the activity level.

- Impact Duration Index (IDI), the time duration of the peak window. This feature was reported useful to discriminate false alarms.

- Maximum Peak Index (MPI) the maximum value of a_t within the peak window $[is, ie]$.

- Minimum Valley Index (MVI) the minimum value of a_t within the peak window $[is, ie]$.

- Peak Duration Index, $PDI = pe - ps$, with ps the peak start defined as the time of the last magnitude sample below $th_{PDI} = 1.8 \times g$ occurred before pt, and pe the peak end defined as the time of the first magnitude sample below $th_{PDI} = 1.8 \times g$ occurred after pt. The higher the value of this feature the higher the probability the peak is a fall.

- Activity Ratio Index (ARI) is the ratio between the number of samples not in $[th_{ARIIlow}, th_{ARIIhigh}$ and the total number of samples in the 700 ms interval centered in $(is + ie)/2$. These thresholds were fixed in [20] as $th_{ARIIlow} = 0.85 \times g$ and $th_{ARIIhigh} = 1.3 \times g$. This ratio measures the activity level during the peak window.

- Free Fall Index, FFI, the average magnitude in the interval $[t_{FFI}, pt]$. The value t_{FFI} is the time between the first acceleration magnitude below $th_{FFI} = 0.8 \times g$ occurring up to 200 ms before pt; if not found, it is set to $pt - 200\ ms$. This feature was reported valid for recognizing jumps.

- Step Count Index, SCI, measured as the number of peaks in the interval $[pt - 2200, pt]$. This feature helps in discriminating between high level activities (walking, running, hopping, etc.) before the peak time.

3.2.3. The ML stage

Several well-known ML techniques are included in this study. All of these ML techniques have been chosen so their implementation in a smartwatch does not drain the battery in normal performance. The models are feed-forward NN (the method originally proposed in [20, 21] but enhanced with a better parameter selection [23]), RBS learned with the C5.0 (a R implementation of the C4.5 algorithm), DT learned with the C5.0, and SVM. Both DT learned with C5.0 and SVM were not included in the final experimentation analysis because their poor performance with the validation data set. Instead, the kNN is proposed as an alternative. However, it is worth mentioning that implementing kNN implies the selection of the optimum collection of instances as the computational costs lineally grows with the number of instances, penalizing the battery cycle (refer to [24] for a study on the selection of the kNN instances for FD).

Moreover, different ensemble of classifiers solutions have been analyzed. Two simple ensemble methods were used to merge the output of the NN, the RBS and the kNN. The aggregation of the outputs was performed using, on the one hand, the voting scheme and, on the other hand, the at-least-one-vote approach. Finally, a more complex scheme of ensemble is included with Random Forests (RF). The general overview of the deployed models is depicted in Fig. 3.

3.3. Measuring the quality of a solution

In this study we are using the Accuracy and the Kappa Factor to evaluate the merit of each possible solution. While the Accuracy is highly known, the Kappa Factor is not; the next text introduces these two measurements.

Typically, after obtaining the classification results the confusion matrix is filled in. There are four basic counters: True Positive (TP), True Negative

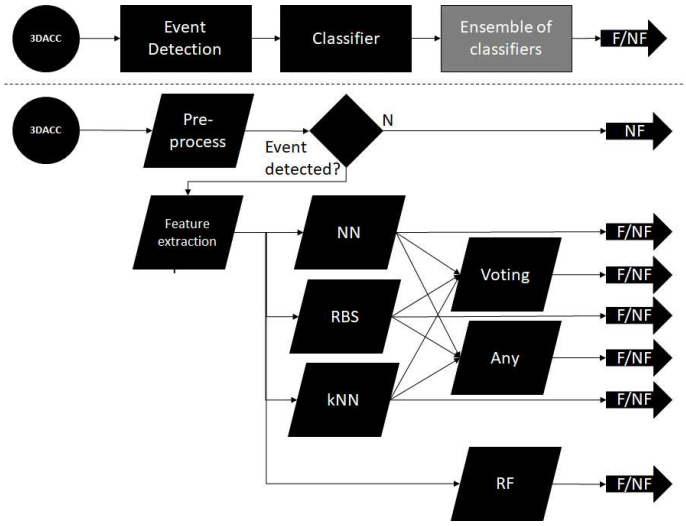

Figure 3: Methods and scheme evaluated in this research. Three different modeling techniques plus three different ensemble methods are proposed. The last ensemble method is RF, which has its own training stage and does not consider the output of any other method.

(TN), False Positive (FP) and False Negative (FN), $\sum = TP + TN + FP + FN$. With these counters, the following calculations for the Accuracy and the Kappa Factor can be obtained.

$$Accuracy = \frac{TP + TN}{\sum} \tag{2}$$

$$\hat{I} = \frac{TP + TN}{\sum} \times \frac{TP + FP}{\sum} + \frac{FP + TN}{\sum} \times \frac{FN + TN}{\sum} \tag{3}$$

$$k = \frac{Accuracy - \hat{I}}{1 - \hat{I}} \tag{4}$$

Clearly, the Accuracy measures the percentage of times a model agrees with the labeled data. On the other hand, the Kappa Factor measures the degree of agreement after the agreement due to chance is removed from consideration; this measurement is highly related with the specificity and sensitivity [41]. Thus higher values of Kappa (higher than 0.5) ensures a good balancing of both the Sensitivity and the Specificity; higher values of the Kappa also guarantees good values in both measurements.

3.4. The experimentation setup

When a model is finally learned, the model is then deployed. Deploying a model means that it will be evaluated with a huge amount of new data; this new data may come from different sources (different sensors, individuals, etc.) and the amount of data will normally be much higher than that used for training. Therefore, the possibilities of finding data that has not been considered during training is not negligible.

In this study, the aim is to reproduce this scenario by involving a single data set for training and several different data sets for validation. As stated before, the UMA Fall, the UNIOVI, the DaLiAC and the TST data sets are included in this experimentation; the former and the latter include simulated falls and ADL, while the second includes ADL and simulated epileptic seizures and the third includes only ADL. Consequently, in this study we have set the UMA Fall for training and testing (the learning stage), while the UNIOVI, TST and DaLiAC are used in validation (the deploying stage). If results are acceptable, then the TST will be used for learning and testing, while the UNIOVI, UMA Fall and DaLiAC will be kept for validation.

It is worth mentioning that all the differences among the data sets have been considered. For instance, the sampling frequency and the sensor span

and range are all parameters of the data sets. Therefore, it is possible to measure the time intervals in the data, so the FSM defined for event detection can be deployed without any problem on the different data sets. Furthermore, knowing the span and range of the sensors in each data set also allows to apply the thresholds defined in the event detection. Finally, given that for each detected peak the set of 8 features is computed and, provided the calculations are performed according to what has been detailed in Section 3.2.2, the produced features would be comparable independently of the data set.

Therefore, for the first experimentation a total of 531 TS will be available for training and testing (including 208 TS containing fall events), while 786 TS will be available for validation (including 132 TS containing fall events). Provided the obtained results are acceptable, then the second experimentation will incorporate 264 TS (including 132 TS containing fall events) for training and testing, while 1053 TS will be kept for validation (208 of them containing fall events).

The TS belonging to the training and testing data set have been evaluated through the peak detection; for each detected peak the set of 8 features described in Section 3.2.2 is calculated. These 8 features from all the detected peaks conforms the data set to train and test the different models. This data set has been standardized computing its mean μ_8 and standard deviation (σ_8). Afterwards, 10-fold cross validation has been employed in the training and testing stage to obtain the best parameter subset for each of the different models; a grid search strategy has been implemented. The best parameter subset is the best configuration of values for each of the parameters that a modeling technique has; for instance, the KNN method has the value K chosen as an odd number within the interval [1, K_{limit}], being K_{limit} a positive integer. After the completion of the 10-fold cross validation, the best parameter subset is used to learn the the best model using the complete 8 features data set.

In order to use all the TS in the validation stage, a fusion stage is included. As mentioned before, the differences in frequency and sensors' span and ranges are considered to allow a coherent processing. The peak detection method is evaluated on each TS belonging to the validation data set. The sliding windows are adapted to the corresponding sampling frequency, and for each detected peak the 8 features are computed. The 8 features extracted from each peak are standardized using the μ_8 and σ_8. Finally, these standardized 8 features instances are classified by the different models analyzed in this study.

This type of experimentation is not new in the context of FD. The research proposed in [30] compared the performance of different classifiers when trained with one data set and validated with a different one. However, there are several differences between these two studies. On the one hand, the peak detection stage of the experimentation in [30] was performed manually, selecting 6 seconds width windows centered in a peak of the acceleration magnitude higher than $1.5 \times g$. On the second hand, the authors did not include any pre-processing method: the models directly tackle the problem taking as input the complete 6 seconds windows. In this case, the solution is more oriented to web services due to the size of the models that could have been gathered, although there is not fixed boundary whether a model is suitable for running in the wearable device or in a remote service.

4. Experimentation and discussion on the results

The experiments were carried out using project R together with the caret, nnet (for the Neural Networks), C5.0 (for the RBS), DMwR (for the kNN) and e1071 (for the SVM and for RF) packages. Table 2 shows the Accuracy and the Kappa Factor obtained for each of the modeling techniques after the training and testing stage. These results represent the performance of the models learned with their best parameter subset and the complete UMA Fall data set. As it can be seen, the performance for the training data set is remarkable good for some of these methods (Neural Networks and Rule Base Systems), and acceptable for the others.

The results obtained in the validation stage are shown in Table 3 and Table 4. Table 3 includes the confusion matrices for each method, while Table 4 shows the Accuracy, Kappa Factor, Sensitivity, Specificity, Precision and Recall obtained for each model.

Method	**Accuracy**	**Kappa Factor**
NN	0.9340	0.8676
RBS	0.9500	0.9000
KNN	0.8722	0.7444
SVM	0.9040	0.8072
RF	0.9000	0.8000

Table 2: Training classification error for each of the analyzed methods.

	NN			**RBS**			**KNN**	
	Reference			Reference			Reference	
Predicted	Fall	NF	Predicted	Fall	NF	Predicted	Fall	NF
Fall	75	95	Fall	98	104	Fall	126	166
NF	57	559	NF	34	550	NF	6	488
	Voting			**Any**			**RF**	
	Reference			Reference			Reference	
Predicted	Fall	NF	Predicted	Fall	NF	Predicted	Fall	NF
Fall	9	63	Fall	59	65	Fall	114	134
NF	123	591	NF	73	589	NF	18	520

Table 3: Results from validation: Confusion Matrices for each of the ML techniques. NF stands for Not_Fall.

Model	**Accuracy**	**Kappa**	**Sensitivity**	**Specificity**	**Precision**	**Recall**
NN	0.8066	0.3793	0.5682	0.8547	0.4412	0.5682
RBS	0.8244	0.4815	0.7424	0.8410	0.4851	0.7424
KNN	0.7812	0.4723	0.9545	0.7462	0.4315	0.9545
Voting	0.8079	0.4899	0.8636	0.7966	0.4615	0.8636
Any	0.7812	0.4723	0.9545	0.7462	0.4315	0.9545
RF	0.8703	0.5902	0.7803	0.8884	0.5853	0.7803

Table 4: Results from validation: statistical measurements for each classifier.

Results shown in Tables 3 and 4 are highly unsatisfactory and cannot be considered good solutions; at most, they can be seen similar to those obtained in commercial products. Although the training results were remarkably good (higher than the 95% of successful), the number of undetected falls is still high when the validation data set grows bigger than the training data; the same happens with the false alarms. All of these results suggest that the obtained models suffered too much with the changes of sensors and/or user, which might be the reason why commercial products rely on the feedback from the user to finally mark a risen fall detection as an alarm.

There are several reasons for this high failure percentage. On the one hand, there are differences in the dynamics according to the fall type. It is not the same when the participant is walking fast, trips over and falls with the

case when the participant is standing still and fades: there are differences in the acceleration magnitude for each type of falls. Several authors suggested taxonomies of the falls; however, there is no agreement in an unique perfectly described fall taxonomy. The problems with the FD methods are not new; actually, the obtained results confirm some reports concerning the performance of deployed FD solutions. There are studies concerned with the dynamics in a fall event with sensors located on the waist [4, 10, 21, 42, 9], establishing the taxonomy and the time periods for each sequence. Interestingly, it has been found that the vast majority of the solutions have been obtained using data gathered from simulated falls. The study published in [11] has already reviewed different methods with a data set of real falls of Parkinson Disease patients and have found that the performances of the published solutions are worse than reported.

Besides, the study published in [32] also found that analyzing the solutions with data gathered from real falls produce a high error rate and rather poor performances. In [23], a comparison between a data set including real falls of elderly people suffering impairment illnesses [11] and the simulated falls and ADL published in [37] -also included in the data set comparison [35]- showed the statistical differences between real falls and simulated ones.

Finally, the dissimilarities between simulated falls and real falls were reported in [13, 42]. It is clear that data from real falls is needed in order to evaluate the solutions and to get a real picture of the different solutions' performance [43]. Furthermore, the relative poor performance of the published FD data set when one is used for training and a different one for validation was reported in [30]. As a conclusion, a general taxonomy and the corresponding set of valid publicly available data sets (ideally, with data from real falls) are still pending to be developed in order to obtain robust and reliable FD solutions.

Additionally, the event detection method can also play an important role in these results. Whenever thresholds are used, the problem of finding their suitable values arises. Up to our knowledge, there is no event detection stage that automatically adapts to the current subject.

On the other hand, there are still techniques that have not been studied in FD. For instance, it is necessary to perform a hybridizing between unsupervised and supervised ML. We do believe that introducing clustering first and classification afterwards would lead to better solutions [24]. However, this hybridization should keep a low computational cost profile; additionally, these techniques need to adapt to the current user [25]. Furthermore, the

use of recurrent networks (similarly as proposed in [33, 44]) can also provide valid means to discriminate the falls. Nevertheless, this type of techniques requires too much computing for a Smartwatch, spending time and draining the battery [4, 34]. The option of sending the TS using wireless links is valid for indoor environments (where WiFi is available), but still requires high communication fares.

Interesting enough to mention, the use of as many public data sets as possible should be combined with real data in normal use condition. That is, the simulated falls and the activity of daily living are not, in many cases, real; they include the data gathered while a participant performed an ADL during certain period of time. Also, the falls start from standing still and letting the participant fall in a mattress. After this experimentation, it seems that fall simulation should, at least, be more similar to that suffered by elderly people. Furthermore, the data gathered from ADL should also come from participants wearing the devices during long periods of time.

5. Conclusion

This study is focused on fall detection, analyzing the current state of the art and performing an experimentation with one of the recently published solutions. The experimentation has mimicked real deployment of solutions: it has been trained with a complete simulated fall data set and validated with several different data sets, some of them including simulated falls as well, while others include ADL or seizure simulations.

After the experimentation, several conclusions were drawn:

- The obtained results confirm some reports concerning the performance of deployed FD solutions. Actually, the validation figures are still far from the desired ones, meaning that there are still problems to be solved in the FD.

- Varying the sensor and the participant is still a challenge, and the models do not provide enough generalization capabilities; relying in the user feedback to avoid the generation of false positive (false alarms).

- There are differences in the dynamics according to the fall type, perhaps due to the lack of a generally accepted fall taxonomy.

- Using simulated data might be an interesting starting point, but gathering data from real falls of healthy participants is clearly needed.

There is more research still pending. Firstly, the development of a well-defined taxonomy of the type of falls is clearly needed. Publicly available data sets including real falls of healthy participants would be useful. However, these data sets are difficult to obtain (and even undesirable, as that involves the fall of an elderly person), alternative solutions should be developed to obtain more reliable and trust-able simulated fall data sets; perhaps, using standard safeguard training mannequins or stuns doubles would help in this task. Secondly, introducing more sophisticated ML schemes might improve the results. Clearly, the solutions based on event detection plus a classification stage might not be enough to cope with this type of problem; perhaps a previous unsupervised learning part would be able to group the main normal behavior, keeping only the classifiers for those cases in the frontier.

Acknowledgements

This research has been funded by the Spanish Ministry of Science and Innovation, under projects MINECO-TIN2014-56967-R and MINECO-TIN2017-84804-R, and by the Grant FCGRUPIN-IDI/2018/000226 project from the Asturias Regional Government.

References

[1] L. Rubenstein, Falls in older people: epidemiology, risk factors and strategies for prevention, Age Ageing 35 (2006) 37–41.

[2] R. Igual, C. Medrano, I. Plaza, Challenges, issues and trends in fall detection systems, BioMedical Engineering OnLine 12 (2013). URL: `http://www.biomedical-engineering-online.com/content/12/1/66`.

[3] S. Chaudhuri, H. Thompson, G. Demiris, Fall detection devices and their use with older adults: A systematic review, J Geriatr Phys Ther 37 (2014) 178–196.

[4] Y. S. Delahoz, M. A. Labrador, Survey on fall detection and fall prevention using wearable and external sensors, Sensors 14 (2014) 19806–19842. URL: `http://www.mdpi.com/1424-8220/14/10/19806/htm`. doi:`doi:10.3390/s141019806`.

[5] S. S. Khan, JesseHoey, Review of fall detection techniques: A data availability perspective, Medical Engineering and Physics 39 (2017) 12–22. URL: http://www.sciencedirect.com/science/article/pii/S1350453316302600. doi:https://doi.org/10.1016/j.medengphy.2016.10.014.

[6] E. Casilari-Pérez, F. García-Lagos, A comprehensive study on the use of artificial neural networks in wearable fall detection systems, Expert Systems with Applications 17 (2017) 198. doi:http://doi.org/10.3390/s17010198.

[7] T. Zhang, J. Wang, L. Xu, P. Liu, Fall detection by wearable sensor and one-class svm algorithm, in: I. G. Huang DS., Li K. (Ed.), Intelligent Computing in Signal Processing and Pattern Recognition, volume 345 of *Lecture Notes in Control and Information Systems*, Springer Berlin Heidelberg, 2006, pp. 858–863. URL: https://link.springer.com/chapter/10.1007%2F978-3-540-37258-5_104?LI=true#citeas. doi:https://doi.org/10.1007/978-3-540-37258-5_104.

[8] A. Bourke, J. O'Brien, G. Lyons, Evaluation of a threshold-based triaxial accelerometer fall detection algorithm, Gait and Posture 26 (2007) 194–199.

[9] A. Bourke, P. van de Ven, M. Gamble, R. O'Connor, K. Murphy, E. Bogan, E. McQuade, P. Finucane, G. Olaighin, J. Nelson, Evaluation of waist-mounted tri-axial accelerometer based fall-detection algorithms during scripted and continuous unscripted activities, Journal of Biomechanics 43 (2010) 3051–3057.

[10] A. K. Bourke, J. Klenk, L. Schwickert, K. Aminian, E. A. F. Ihlen, S. Mellone, J. L. Helbostad, L. Chiari, C. Becker, Fall detection algorithms for real-world falls harvested from lumbar sensors in the elderly population: A machine learning approach, in: 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2016, pp. 3712–3715. doi:10.1109/EMBC.2016.7591534.

[11] F. Bagala, C. Becker, A. Cappello, L. Chiari, K. Aminian, J. M. Hausdorff, W. Zijlstra, J. Klenk, Evaluation of accelerometer-based fall de-

tection algorithms on real-world falls, PLoS ONE 7 (2012) e37062. doi:https://doi.org/10.1371/journal.pone.0037062.

[12] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, T. Jämsaä, Comparison of low-complexity fall detection algorithms for body attached accelerometers, Gait and Posture 28 (2008) 285–291.

[13] M. Kangas, I. Vikman, L. Nyberg, R. Korpelainen, J. Lindblom, T. Jamsa, Comparison of real-life accidental falls in older people with experimental falls in middle-aged test subjects, Gait and Posture 35 (2012) 500–505.

[14] F. Bianchi, S. J. Redmond, M. R. Narayanan, S. Cerutti, N. H. Lovell, Barometric pressure and triaxial accelerometry-based falls event detection, IEEE Transactions on Neural Systems and Rehabilitation Engineering 18 (2010) 619–627. doi:10.1109/TNSRE.2010.2070807.

[15] A. Sorvala, E. Alasaarela, H. Sorvoja, R. Myllyla, A two-threshold fall detection algorithm for reducing false alarms, in: Proceedings of 2012 6th International Symposium on Medical Information and Communication Technology (ISMICT), 2012.

[16] E. Casilari, M. A. Oviedo-Jiménez, Automatic fall detection system based on the combined use of a smartphone and a smartwatch, PLOS ONE 10 (2015) 1–11. URL: https://doi.org/10.1371/journal.pone.0140929. doi:10.1371/journal.pone.0140929.

[17] Q. T. Huynh, U. D. Nguyen, L. B. Irazabal, N. Ghassemian, B. Q. Tran, Optimization of an accelerometer and gyroscope-based fall detection algorithm, Journal of Sensors 2015, Article ID 452078 (2015) 8 pages. doi:http://dx.doi.org/10.1155/2015/452078.

[18] P. Jatesiktat, W. T. Ang, An elderly fall detection using a wrist-worn accelerometer and barometer, in: 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2017, pp. 125–130. doi:10.1109/EMBC.2017.8036778.

[19] A. M. Sabatini, G. Ligorio, A. Mannini, V. Genovese, L. Pinna, Prior-to- and post-impact fall detection using inertial and barometric altimeter measurements, IEEE Transactions on Neural Systems and Rehabilitation Engineering 24 (2016) 774–783. URL: http://ieeexplore.

ieee.org/abstract/document/7173441/. doi:10.1109/TNSRE.2015.2460373.

[20] S. Abbate, M. Avvenuti, F. Bonatesta, G. Cola, P. Corsini, AlessioVecchio, A smartphone-based fall detection system, Pervasive and Mobile Computing 8 (2012) 883–899.

[21] S. Abbate, M. Avvenuti, P. Corsini, J. Light, A. Vecchio, Wireless Sensor Networks: Application - Centric Design, Intech, 2010, p. 22. doi:10.5772/13802.

[22] P. Tsinganos, A. Skodras, A smartphone-based fall detection system for the elderly, in: Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis, 2017.

[23] S. B. Khojasteh, J. R. Villar, C. Chira, V. M. González, E. de la Cal, Improving fall detection using an on-wrist wearable accelerometer, Sensors 18 (2018) 1350. doi:https://doi.org/10.3390/s18051350.

[24] J. R. V. Mirko Fáñez, E. de la Cal, V. M. González, J. Sedano, Feature clustering to improve fall detection: A preliminary study, in: F. M. Álvarez, A. T. Lora, J. A. S. Muñoz, H. Quintián, E. Corchado (Eds.), Proceedings of the 14th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2019). Advances in Intelligent Systems and Computing, volume 950, Springer, 2019, pp. 219–228.

[25] J. R. Villar, E. de la Cal, M. Fañez, V. M. González, J. Sedano, User-centered fall detection using supervised, on-line learning and transfer learning, Progress in Artificial Intelligence 2019 (2019) 1–22.

[26] P. Kostopoulos, T. Nunes, K. Salvi, M. Deriaz, J. Torrent, F2d: A fall detection system tested with real data from daily life of elderly people, in: 2015 17th International Conference on E-health Networking, Application Services (HealthCom), 2015, pp. 397–403.

[27] I. P. E. S. Putra, J. Brusey, E. Gaura, R. Vesilo, An event-triggered machine learning approach for accelerometer-based fall detection, Sensors 18 (2018) 2034. doi:https://doi:10.3390/s18010020.

[28] H. Gjoreski, J. Bizjak, M. Gams, Using smartwatch as telecare and fall detection device, in: 2016 12th International Conference on Intelligent Environments (IE), 2016, pp. 242–245. doi:10.1109/IE.2016.55.

[29] A. Hakim, M. S. Huq, S. Shanta, B. Ibrahim, Smartphone based data mining for fall detection: Analysis and design, Procedia Computer Science 105 (2017) 46–51. URL: http://www.sciencedirect.com/science/article/pii/S1877050917302065. doi:https://doi.org/10.1016/j.procs.2017.01.188.

[30] R. Igual, C. Medrano, I. Plaza, A comparison of public datasets for acceleration-based fall detection, Medical Engineering and Physics 37 (2015) 870–878. URL: http://www.sciencedirect.com/science/article/pii/S1350453315001575. doi:https://doi.org/10.1016/j.medengphy.2015.06.009.

[31] M. Deutsch, H. Burgsteiner, Health Informatics Meets eHealth, volume 223 of *Studies in Health Technology and Informatics*, IOS Press, 2016, pp. 259–266. doi:10.3233/978-1-61499-645-3-259.

[32] C. Medrano, I. Plaza, R. Igual, Á. Sánchez, M. Castro, The effect of personalization on smartphone-based fall detectors, Sensors 16, Article ID 117 (2016) 117. URL: http://www.mdpi.com/1424-8220/16/1/117/htm. doi:10.3390/s16010117.

[33] T. Mauldin, M. Canby, V. Metsis, A. Ngu, C. Rivera, Smartfall: A smartwatch-based fall detection system using deep learning, Sensors 18 (2018) 3363. URL: http://dx.doi.org/10.3390/s18103363. doi:10.3390/s18103363.

[34] P. M. Vergara, E. de la Cal, J. R. Villar, V. M. González, J. Sedano, An iot platform for epilepsy monitoring and supervising, Journal of Sensors 2017, Article ID 6043069 (2017) 18 pages. doi:10.1155/2017/6043069.

[35] E. Casilari, J.-A. Santoyo-Ramón, J.-M. Cano-García, Analysis of public datasets for wearable fall detection systems, Sensors 17 (2017) 4324 – 4338. URL: http://www.mdpi.com/1424-8220/17/7/1513. doi:https://10.3390/s17071513.

[36] A. Sucerquia, J. D. López, J. F. Vargas-Bonilla, Sisfall: A fall and movement dataset, Sensors 17 (2017) 198. doi:http://doi.org/10.3390/s17010198.

[37] E. Casilari, J. A. Santoyo-Ramón, J. M. Cano-García, Umafall: A multisensor dataset for the research on automatic fall detection, Procedia Computer Science 110 (2017) 32 – 39. URL: http://www.sciencedirect.com/science/article/pii/S1877050917312899. doi:https://doi.org/10.1016/j.procs.2017.06.110, 14th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2017) / 12th International Conference on Future Networks and Communications (FNC 2017) / Affiliated Workshops.

[38] S. Gasparrini, E. Cippitelli, E. Gambi, S. Spinsante, J. Wahslen, I. Orhan, T. Lindh, Proposal and experimental evaluation of fall detection solution based on wearable and depth data fusion, in: ICT Innovations 2015, Advances in Intelligent Systems and Computing, volume 399, Springer, 2016, pp. 99–108. URL: http://www.tlc.dii.univpm.it/blog/databases4kinectandhttps://ieee-dataport.org/documents/tst-fall-detection-dataset-v2. doi:10.1007/978-3-319-25733-4_11.

[39] H. Leutheuser, D. Schuldhaus, B. M. Eskofier, Hierarchical, multi-sensor based classification of daily life activities: comparison with state-of-the-art algorithms using a benchmark dataset, PLoS ONE 8 (2013). doi:https://doi.org/10.1371/journal.pone.0075196.

[40] J. R. Villar, P. Vergara, M. Menéndez, E. de la Cal, V. M. González, J. Sedano, Generalized models for the classification of abnormal movements in daily life and its applicability to epilepsy convulsion recognition, International Journal of Neural Systems 26 (2016). doi:10.1142/S0129065716500374.

[41] M. Feuerman, A. Miller, Relationships between statistical measures of agreement: sensitivity, specificity and kappa, Journal of Evaluation in Clinical Practice 14 (2008) 930–933.

[42] M. Kangas, I. Vikman, J. Wiklander, P. Lindgren, L. Nyberg, T. Jamsa, Sensitivity and specificity of fall detection in people aged 40 years and over, Gait and Posture 29 (2009) 571–574.

[43] T. Vilarinho, B. Farshchian, D. G. Bajer, O. H. Dahl, I. Egge, S. S. Hegdal, A. Lønes, J. N. Slettevold, S. M. Weggersen, A combined smartphone and smartwatch fall detection system, in: 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, 2015, pp. 1443–1448. doi:10.1109/CIT/IUCC/DASC/PICOM.2015.216.

[44] J. C. Núñez, R. Cabido, J. J. Pantrigo, A. S. Montemayor, J. F. Vélez, Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition, Pattern Recognition 76 (2018) 80–94. doi:https://doi.org/10.1016/j.patcog.2017.10.033.