Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2019) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

A low-cost and unobtrusive system for fall detection

Jesús Fernández-Bermejo Ruiz[a], Javier Dorado Chaparro[a], Cristina Bolaños Peño[a], Henry A. Llumiguano Solano[a], Xavier del Toro García[a], Juan C. López López[a]

[a] *Computer Architecture and Networks Group, University of Castilla-La Mancha, Paseo de la Universidad, 4, Ciudad Real, 13005, Spain*

Abstract

Nowadays, the amount of elder people living alone is increasing, with all the risks that it involves, maybe the most dangerous threat that they face is fall off with nobody around to help them. A fall at an advanced age usually leads to consequences like bones and hip fracture, which in addition to the low mobility that these people present, make to stand up impossible for them. This situation can get worse if after the fall a person loses consciousness, making it impossible to contact to a third party for help by means of a mobile phone or something similar. Different solutions have been developed in order to accomplish this problem, but some of them are not realistic enough, for example some video solutions invade our privacy, and those which are based on mobile phones expect the user to go everywhere with it. In this paper, it is proposed a low cost solution based on an 9-axis *IMU (Inertial Measure Unit)*, which counts with an accelerometer, gyroscope and magnetometer that will give us the needed information to build a fall detector supported by machine learning. This system will include a gateway, which will be responsible of the data collection and the most complex computations.

Keywords: Active aging; Healthcare; Fall detector; IMU sensors; Accelerometers; Gyroscopes; Machine Learning

1. Introduction

During the 21st century, the life expectancy in the population has increased considerably. While in 1950 life expectancy was around 50 years, some studies have already predicted an increase in life expectancy to 76 years in the coming decades [2]. Increased life expectancy implies that by 2050 there will be more than 2 billion people over the age of 60. Older people generally have health problems in addition to limited mobility, so maintaining a good quality of life becomes one of the greatest challenges for our society.

[*] Corresponding author. Tel.: +34-926-295-300 ; fax: +34-926-295-300 ; Ext: 96621.
E-mail address: jesus.fruiz@uclm.es

Technological solutions that facilitate active aging and improve health as we age are gaining attention as a means to help us meet the challenges brought up by an aging population. New technologies have already been used to diagnose and treat all types of diseases, and even to monitor physical health status of patients. Studies as [7] have exposed the importance of exercising and keeping an active lifestyle to minimize physical and psychological issues as we age. At the same time, different works have used technologies to encourage active aging, as for example [8] in which an at home rehabilitation system for rheumatoid arthritis patients is presented. Other researches have focused in detecting and diagnosing physical problems and diseases, in [20] a solution to detect parkinson diseases is presented based on computer vision techniques.

There is one area in which new technologies have great potential in the detection of falls involving older people. According to the WHO (World Health Organisation) [17], falls are the second leading cause of death from unintentional injuries worldwide, a situation that is especially aggravated in adults over 65. Approximately 30% of older people suffer serious injuries after a fall, sometimes also leading to psychological problems. Approximately 28-35% of these people fall each year, a percentage that increases to 30-50% for people living in long-term care institutions. Falls account for 40% of all injury deaths, with a significant increase in older age groups [17].

To reduce the physical and psychological consequences of a fall, real-time fall detection, responding and providing assistance to the patient as soon as possible, is gaining importance. IMUs and image processing based solutions are some of the advanced solutions being developed. In these systems, two algorithms stand out from the rest, the *Threshold Based Algorithm (TBA)*, used in constrained devices to process data at the edge, and Machine Learning algorithms, which are used when data processing occurs in the cloud or in an external node. This study focuses on the use of IMUs for fall detection, more specifically, on the MetaMotionR sensor from Mbientlab, that provides information about the acceleration, rotation and magnetic field that the sensor undergoes thanks to the accelerometers, gyroscopes and magnetometers it has.

2. Related Work

Many papers can be found on fall detection systems using IMUs. These studies use the data collected from accelerometers and gyroscopes to characterize a person's movement in order to determine when this movement is a potential fall. Accelerometers alone are more commonly used in these studies, but both devices have been used over the years. Three different types of systems can be distinguished.

First of all, there are the *TBA systems*, which define a threshold on monitored measurements, which, when exceeded, a fall is assumed by the system. Such systems are very common when the algorithm is embedded in a constrained device, such as an ESP8266 microcontroller, without enough resources to perform complex operations. In [5] a general solution is presented for mobile platforms, using IMUs from both wearable sensor and a smartphone. The data gathered from these sensors are fed to a TBA algorithm, specifically designed to perform fall detection. Additionally, [4] presents a similar solution but using only accelerometer data. Other examples include the work in [18], in which raw data from an accelerometer is used in a TBA merged with the calculation of the sensor orientation to determine the position of the user. Furthermore, other studies such as [14] and [15] devise an algorithm that predicts when a fall is about to occur. These systems have the drawback of involving several sensors placed on different parts of the body, which makes it unfeasible to use it in everyday life.

The second type of systems are those based on Machine Leaning, in which a model is trained and defined. A labelled dataset is used for training. The label indicates whether the data represent a fall or not. Usually, a set of features representing the user's movement is extracted from the raw data. Different machine learning approaches have been used over the years, [6] designed a system with a wrist-worn wearable device that has an accelerometer, gyroscope and magnetometer. Some features are extracted from the data collected from these sensors to train different models, the best performing models were *k-NN (k Nearest Neighbors)*, Logistic Regression and *SVM (Support Vector Machines)*. In [11], a similar study was conducted. Different machine learning algorithms were used in this study and evaluated in three different approaches, only with accelerometers, only with gyroscopes and with data extracted from visual processing.

The last system, and the one selected for this paper, is a mix of the two previous ones. These systems use TBA to detect if a movement can be considered a fall, and if it is, the movement data will be transferred to a previously trained machine learning model algorithm, which will decide if the movement is indeed a fall. In [1] a state machine is defined that will monitor the measurements of the user's smartphone's accelerometer. This state machine works like TBA and when a abrupt movement is detected, the data are sent to a neural network to determine whether the movement is a fall. If a fall is wrongly detected, the data are used as feedback, introducing it into the neural network labeled as a non-fall, improving the classification.

It is worth noting that there are solutions for fall detection based on image processing. The work in [13], for example, detects falls using a system based on Microsoft's Kinect. The work in [10] presents a solution based on the fusion of data retrieved from IMUs and Kinect.

In this work, a low-cost solution is presented, which is capable of monitoring the user in real-time and for a long period of time, in an unobstrusive and less invasive way, using only one device. The focus of the paper is the inclusion of orientation at the time to evaluate if a movement is a fall, a very valuable information that can make the difference.

3. Algorithm

The system proposed here is based on a combination of TBA and Machine Learning techniques. Inspired by the work in [1], this paper proposes a variant of the state machine that they originally presented . However, there are some issues that need to be addressed before describing the algorithm proposed here.

First, the location of the sensor has to be decided. This is important because depending on the sensor location, the data could be different and many studies use different locations such as the wrist, the waist, different parts of the leg, etc. It was decided that sensor should be located at the waist, as it is the centre of gravity and therefore it reduces the noise in the data. Furthermore, the waist is the most significant location for detecting imbalances that can cause a fall. Second, the sampling rate has to be defined. According to the research of [16], the human body moves in a range of frequencies between 0 and 15Hz, so the frequency should be at least 30Hz in order to comply with Nyquist's theorem. Moreover, two of the most commonly used frequencies for fall detection were 100Hz, as considered in [15] and [18] and 50Hz in [1]. Thus, while the sensor is configured at 100Hz, the actual frequency ranges between 40Hz and 70Hz, due to losses in the BLE connection leading to a drop in the received readings.

Thus, once it has been decided where to place the sensor and what is its sampling rate, the proposed recognition algorithm can be analysed. Two parts can be clearly distinguished. On the one hand, the TBA, which will be in charge of recognising when a movement is a possible fall. On the other hand, the machine learning part, which will consist of a previously trained model that processes the data that the TBA considers as a possible fall. With these data, the model will be able to confirm or discard a movement as a possible fall. The main reason behind the machine learning component is that in our daily life there are some activities that can cause data resembling a fall, such as sitting on a chair, which can be confusing for the TBA. These activities are known as *Activities of Daily Living (ADLs)*.

3.1. Threshold Based Algorithm

The role of the TBA is to detect when a movement is likely to be a fall. For this purpose, acceleration is a good indicator of a sudden movement. In this sense, the acceleration module of the vector (to be denoted as *AM*) can be calculated using the acceleration of each axis (see equation 1).

$$AM = \sqrt{A_x^2 + A_y^2 + A_z^2}. \tag{1}$$

The implementation of the algorithm is based on a *Finite State Machine (FSM)*. The FSM uses the AM value to determine when a movement entails a fall. In the development of the FSM, [1] has been taken as a reference, so for further details please refer to this paper. The main difference between the FSM presented here and the one selected as a reference is the threshold used to decide whether a movement can be a fall. In [1], AM should be higher than 3G, but based on our experience, when a person falls, he/she tries to stop the fall, which can cause a smooth fall that reduces the accelaration peaks, so 2.5G seems to be a better choice. Another important difference is that in post-peak state, the state machine does not return to the sampling state if another peak is detected, because a fall could involve more than one peak. The FSM can be found in Figure 1. Each state represents the following:

- Sampling: Raw data from the sensor is being received.

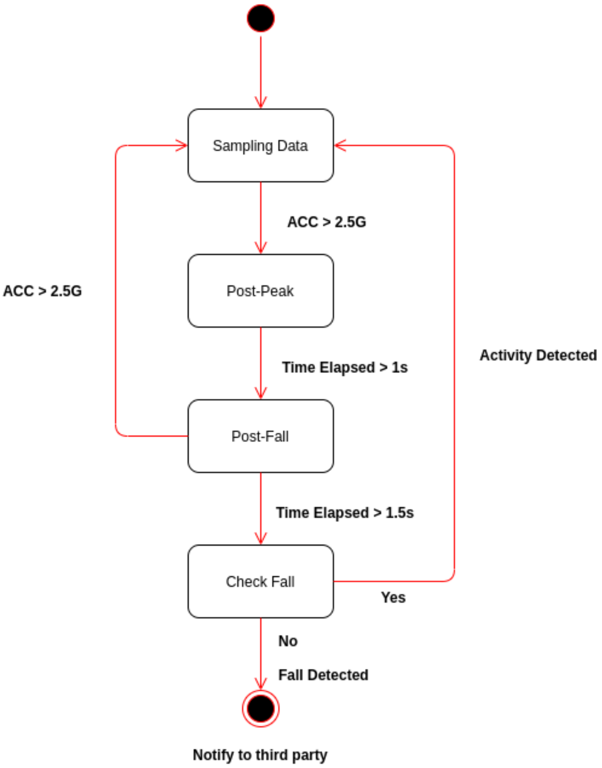

Fig. 1. Finite State Machine from the system

- Post-peak: After detecting an AM higher than 2.5 G, all data are saved in a one-second window, which is expected to be the duration of a fall.
- Post-fall: A possible fall has been registrated, it is necessary to wait a second and a half and if a peak is detected it means that the user has not fallen, the window is then cleaned and the system returns to the post-peak stage.
- Check-fall: A possible fall is detected after one and a half seconds with no relevant activity. The window data is sent to the machine learning component to determine whether the data actually represent a fall.

3.2. Machine Learning

The second part of the algorithm involves a machine learning model to recognise when a potential fall is indeed one and not an ADL. Essentially two components are required for the machine learning engine: a dataset with labelled data, which contains different movements each indicating whether it is a fall or an ADL, and a model trained from these data. Regarding the dataset, there are many different ones available from previous studies. A survey of previously used public datasets for fall detection is presented in [3]. A major issue encountered when selecting a dataset for fall detection is the adaptation of the dataset to the case study. A dataset was required that would capture the data from the available sensors on the MetaMotionR, and with the device placed on the user's waist. Eventually a dataset with these requirements could not be found, so it was decided to construct a dataset to meet these requirements (see Section 4).

The next step in building the machine learning algorithm is to determine the best training approach. Different studies discuss the best method for fall detection, and two algorithms stand out from the rest, k-NN and SVM. According to [19], SVM is the one yielding best results, which is why this was the selected method. The main objective of SVM is to create the hyperplane that best separates the classes considered by the model (fall or ADL). Furthermore, the SVM model involves a set of hyperparameters that help the algorithm to improve its results while avoiding overfitting and underfitting. Among all these parameters, the three most important ones were analysed in depth:

- C: Indicate a penalty for each misclassified point.
- Gamma: Indicate the curvature of the hyperplane that separates the data.
- Kernel: Represents the distribution of the classes.

A cross-validation technique can be used to optimise these hyperparameters. In this technique, all combinations of different values for the hyperparameters are explored. The values yielding the best classification results will be chosen as the optimal values for these hyperparameters.

Now that both the data and the model to be used have been defined, the following steps describe the machine learning algorithm proposed here:

1. Feature Extraction: The first step is to extract features from the raw data collected from a movement captured in a 1-second window. This is mainly intended to reduce the dimensionality of the input data by rendering abstractions that represent the movement with fewer data. Some examples of features used in this study are the mean and standard deviation of the original data. The work in [9] demonstrates that time domain features, such as the aforementioned ones, are the best choice to recognise human activity. This step will be developed with further detail in Section 5.
2. Scaling: Once the features have been extracted, because each feature will have different magnitudes, it is necessary to apply a scaler to avoid some features having more weight than others. A standard sklearn scaler was used.
3. Feature Selection: Despite having used a scaler, some features will be more relevant than others when detecting a fall. Finding these features means that the less relevant ones can be eliminated and the noise can be minimised. To do so, some methods can be used such as ANOVA f-value, Pearson correlation, Lasso regression and recursive feature elimination.
4. Cross validation: Train the model with selected features and use cross validation to obtain the model with the best hyperparameters.

4. Building dataset

To train the machine learning model, it is necessary to create a dataset to distinguish between data that correspond to a fall and data that correspond to an ADL. To do so, a set of exercises are defined, consisting of falls and ADLs, performed by different subjects under a controlled environment. During the performance of the exercises, when the AM exceeds the threshold, a one-second window will be stored in a csv file. The following exercises have been considered:

ADLs:

- Hit the sensor
- Jump
- Run and stop
- Sit on a chair
- Pull the sensor

Falls:

- Forward fall
- Fall on the right side
- Fall on the left side
- Backward fall

There are some activities considered under the ADLs set that, at first, could be difficult to be understood as proper ADLs, e.g. hitting and pulling the sensor. However, these involve sensor disturbances but not in the context of a fall. For example, the user may inadvertently hit the sensor, and if the system is not prepared for such a situation, this may result in a false positive. In addition, during data collection, the sensor sometimes experiences some unexpected peaks that result in a fall trigger, e.g. when a subject adjusts his or her trousers. These types of activities, in which the sensor may undergo some variations that are not a straight consequence of a user's movement, may be represented by the sensor pull activity.

Data were collected at the *ITSI (Institute of Information Technologies and Systems)* building, from the University of Castilla-La Mancha. 17 different individuals performed the defined set of exercises. One of the issues that were discussed was whether it was appropriate to use a mat to collect the data, as its use may lead to smoothing the acceleration peaks. Otherwise, the individual could perform the activities in fear of injury, resulting in data that do not represent a real fall. Finally, the decision was made to collect all data with a mat, as it was considered that it is preferable to have data that approximate a real fall with a slightly smoothed peak. The subjects who performed the activities to create the dataset were 4 females and 13 males, with an average age 30 ± 8.02 years, average height 174.18 ± 7.85 cm, average weight 74.35 ± 9.71 kg. The labelled dataset is available on the website of the ARCO research group[1].

5. Feature Extractor

When an abrupt movement of the user is detected, the raw data from the sensors are saved in a window of 1 second. These raw data usually do not represent a movement in the best way because includes some noise data, for this reason, is needed to extract features that will filter it and generate abstractions in order to describe the movement. Once a sudden movement is detected, sensor raw data are stored in a 1-second window.

Most studies on fall detection systems use accelerometers as the most commonly used sensors, and in some cases gyroscope data are also added. Nevertheless, this study intends to use all the features offered by the MetaMotionR sensor. This sensor has a 9-axis accelerometer, gyroscope and magnetometer, which provide information about acceleration, rotation and magnetic field in each axis. Furthermore, this sensor has a mode called Sensor Fusion, which offers the possibility to fuse all data coming from the three aforementioned sensors, and get information about the absolute and relative orientation. The absolute orientation could be very valuable, as it can reveal whether the user is lying on the ground, thus being a strong indicator that a fall might occurred.

Given all the sensor features, the raw data to be processed are the acceleration (in G) and rotation (in °/s) in each axis, and the absolute orientation (in euler angles) in pitch, roll and yaw. All these data are collected and processed in a one-second window after the first peak exceeding 2.5 G is detected.

Selected raw data provide enough information to characterise the user's movement. The first step is therefore to identify which useful features can be extracted from these raw data. Some studies have previously focused on which features are best suited to perform human activity recognition. In [9] it is discussed that the most commonly used features are those in the time domain and the frequency domain. In the time domain, we can observe some statistical operations such as mean and standard deviation that will help us filter and reduce the original data . On the other hand, in the frequency domain, *DTF (discrete fast Fourier transforms coefficient)* stands out as it enables the decomposition of a user's movement signal and obtains the power and frequency of all the signals that comprise the movement. This work is based in the following selected features.

- Time-Domain
  - Acceleration mean: Mean of all the AM saved in the window.
  - Acceleration stdev: Standard deviation of all the AM saved in the window.
  - Acceleration Y mean: Mean of all acceleration in the window on Y axis.
  - Acceleration Z mean: Mean of all acceleration in the window on Z axis.
  - Acceleration Y stdev: Standard deviation of all acceleration in the window on Y axis.
  - Acceleration Z stdev: Standard deviation of all acceleration in the window on Z axis.
  - Rotation Y mean: Mean of all rotations in the window on Y axis.
  - Rotation Z mean: Mean of all rotations in the window on Z axis.
  - Rotation Y stdev: Standard deviation of all rotations in the window on Y axis.
  - Rotation Z stdev: Standard deviation of all rotations in the window on Y axis.
  - Fall time: Time elapsed between the AM that triggers the fall and the last acceleration greater than 1.5G.
  - Orientation (Pitch, Roll and Yaw): Orientation of the sensor once the movement is finished (2.5 seconds after the peak that trigger the movement)
- Frequency-Domain

[1] https://arcoresearch.com/2021/04/16/dataset-for-fall-detection/

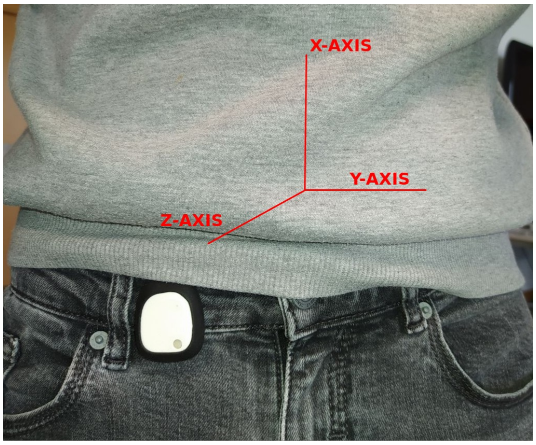

Fig. 2. Axis from the sensor

- Max Fourier coefficient in acceleration: Max amplitude on signals that compound the acceleration
- Max Fourier coefficient in rotation: Max amplitude on signals that compound the rotation

There are some features that focus on the Y and Z axis but none on the X axis (local axis of the IMU), as can be noted from the aforementioned features. The reason for this is that when the sensor is placed on the user's waist, the X axis is perpendicular to the ground, so movement in this axis will rarely involve a fall (the axis of the sensor can be observed in Figure 2). It was therefore concluded that to avoid the inclusion of noise to the data, the X-axis data should be minimised.

6. Proposed Architecture

The proposed architecture consists of two main components, the sensor that will monitor the user's movements, and a gateway device that will collect and process all the data to decide whether a movement is a fall or an ADL. When a movement is classified as a fall, the gateway will be responsible for notifying that situation.

The sensor used to monitor users' movements is the MetaMotionR sensor from the Mbientlab manufacturer[2]. It has two recording modes, the streaming mode for receiving data in real time and the logging mode, for storing the data on the device itself for later downloading. The logging mode only has memory for 500,000 measures, which may not be enough to monitor a user over a long period of time. On the other hand, the streaming mode guarantees, according to the manufacturer's specifications, 24 hours of continuous monitoring.

While streaming mode is more suitable for our purpose, which is to detect falls in real time, it is necessary to confirm whether the theoretical measuring time is actually the time a sensor can be working before running out battery. To check the manufacturer's specification, the sensor was left in streaming mode, collecting data for almost 32 hours non-stop, confirming that it has an autonomy of 32 hours, slightly longer than the time indicated by the manufacturer. This is enough time to monitor a user for more than a day, so the sensor is suitable for a real-world environment. Because the aim of this study is to create the least intrusive solution possible, only one sensor will be used, which will be placed on the user's waist. This choice was selected mainly because the waist is the centre of gravity of a person, being less sensitive to ADL movements.

A positive feature of the MetaMotionR sensor is its configurability. The accelerometer offers a sampling rate ranging from 0.001Hz to 100Hz, with a choice of ± 2, ± 4, ± 8 and ± 16 G. The gyroscope has similar characteristics, with the same sampling rate and a range of ± 125, ± 250, ± 500, ± 1000 and ± 2000°/s. The magnetometer sampling rate is limited to a maximum of 25Hz and a range of ±1300 µT (in x,y-axis), ±2500 µT (in z-axis). This sensor also

[2] https://mbientlab.com/metamotionr/

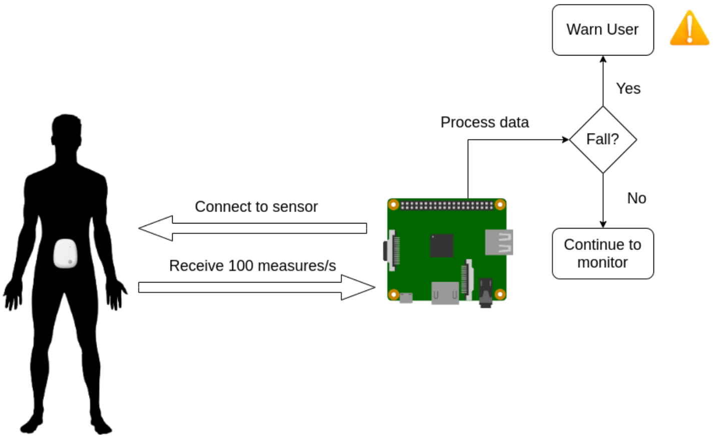

Fig. 3. Fall Detector system architecture

has a Fusion mode in which data from these three sensors are combined using algorithms implemented in the firmware to extract data such as absolute orientation (in Euler Angles or Quaternions).

Most recent studies involve fall detection solutions with IMUs embedded in smartphones. However, this comes with some drawbacks that make the system unusable. First, this is not the primary purpose of a cell phone, as real-time data transmission reduces the device's battery life. Second, it is not realistic to depend on the user carrying the cell phone everywhere, these systems gain relevance when the user is alone, a situation that usually occurs at home, where hardly ever the user will have the smartphone with him or her. Finally, the target audience group for the system presented here are the older people. This group of people do not normally use or even have cell phones. In conclusion, the use of wearables seems to be more convenient and more useful for detecting falls at home.

There is also a second component, the gateway. This gateway will not only collect all sensor data, but will also act as a processing node, being responsible for the implementation of the entire algorithm. This algorithm will be implemented as a fall detection service on the gateway. The user will wear the sensor on the waist, and this service will connect to it and keep the connection alive to monitor the state of the user, receiving around 100 measurements per second. Whenever an event occurs (which in this case could be a fall), the gateway will analyse the user's activity, and if it concludes that a fall is detected, it will communicate with a third party to alert this party. In the first version of the system, a Raspberry Pi 4 is proposed as a gateway. The architecture described before can be observed on Figure 3.

7. Results and evaluation

This section presents the results of the assessment and performance evaluation of the proposed system. When analysing a movement, 4 different situations can occur: a true positive (TP) when a fall event has been correctly classified; a true negative (TN) when a non-fall event has been correctly identified; a false positive (FP) when a fall event has been misidentified and a false negative (FN) when a non-fall event has been misclassified. With these four possible outcomes the performance of the system is evaluated with the following features:

- Sensitivity: Measures the amount of positives that are correctly identified.
- Specificity: Measures the amount of negatives that are correctly identified.
- Accuracy: Measures the amount of events that are correctly identified.
- F1: Is the harmonic mean of the precision and recall, and is extensively used to measure the quality and performance of a model.

In order to test the algorithm, part of the data previously collected need to be used. Of the 765 measures retrieved from the 17 subject, the 80% of the data were used to train the system, and the rest were used to test it. The results of the evaluation conclude in a Sensitivity, Specificity, Accuracy and F1 of 100%, classifying all test data in the correct way. It can be observed that all events are correctly classified, which proves that the system is a good and reliable option for detecting falls in real time and in an unobtrusive way, but it should be noted that these results, despite being very good, were obtained using exercise data in a stable and controlled environment.

In Table 1 is developed a comparison between the work of this paper and other studies, using Specificity, Sensitivity and Accuracy.

Table 1. Comparison between systems

Study	Sensitivity	Specificity	Accuracy
J. Fernández-Bermejo et al.	100%	100%	100%
J. Fernández-Bermejo et al. (Without Orientation)	100%	97.70%	98.69%
De Cillisy et al.[5]	100%	100%	100%
Wu et al.[18]	97.1%	98.3%	97.78%
Tsinganos et al.[16]	97.53%	94.89%	95.55%
DeQuadros et al.[6]	95.8%	86.5%	91.1%
Abbate et al.[1]	100%	100%	100%
Kwolek et al.[10]	100%	96.67%	98.33%

It was expected that orientations could make the difference on fall classification, and as it can be observed in the results, the inclusion of orientations as features increase the Specificity and Accuracy from 97.70% and 98.69% respectively to 100%, recognising all the activities properly.

As it can be observed, some studies as [5] and [1] produce the same performance as the study here presented, with the difference that these system do not use the orientation, only accelerometer and gyroscope data. In [5] it is used what they named as heading, which is a relative orientation of the user from a vertical plane (but at the end an orientation which is calculated from raw data), explaining the great performance of the system taking into account it is a TBA algorithm. The work of [1], on the other hand, produces a great performance trusting only in accelerometer data, these results can be explained as the result of a great feature selection, and a great performance of neural networks to classify falls.

It can be concluded that orientations can indeed make a difference. The best performing papers, presented here and in [5] make use of these features. Exceptionally, the work in [1] presents a 100% accuracy solely using accelerometer data, with one main difference, they use a neural network instead of the commonly used machine learning models in fall detection, which seems to perform better in conjunction with the selected features.

8. Conclusions

This paper presents a fall detection system for older adults based on an IMU attached to the user's waist. This system has great potential as it is a low-cost, unobtrusive system with a high level of reliability that can monitor a user for more than 24 hours non-stop. There are however, some drawbacks that need to be addressed, such as the short operating range of the system. Due to the use of Bluetooth to communicate the sensor, the working range of the system is about 5 metres around the gateway. Thus, addressing this issue and increasing the working range is a future work area that needs to be addressed.

The system proposed here has obtained an accuracy of 100%, correctly classifying all movements. However, the testing events were performed in a controlled environment with dictated exercises. The next step involves the deployment of this solution in a real environment as it is a nursing home. The system robustness and accuracy will be evaluated in a real environment with non-simulated falls. For this purpose, residents of the Nursing Home El Salvador (Córdoba, Spain) will be monitored for a week to register and record data, under the SHAPES Project [12].

Future works, therefore involve upgrading the connection range of the sensor and improve the system usability and robustness. Moreover, the operation of the system will be have to be evaluated in a real environment.

Acknowledgements

This paper is funded by European Union's Horizon 2020 programme under grant agreement ID 857159, project SHAPES (Smart and Healthy Ageing through People Engaging in Supportive Systems) by the Spanish Ministry of Economy and Competitiveness under project PLATINO (TEC2017-86722-C4-4-R) by the Regional Government of Castilla-La Mancha under project SYMBIOT (SBPLY/17/180501/000334) and by Spanish Ministry of Education, Culture and Sport within the framework of the National FPU Program (ref. FPU FPU19/03387).

References

[1] Abbate, S., Avvenuti, M., Bonatesta, F., Cola, G., Corsini, P., Vecchio, A., 2012. A smartphone-based fall detection system. Pervasive and Mobile Computing 8, 883–899. doi:10.1016/j.pmcj.2012.08.003.

[2] Bloom, D. E., Boersch-Supan, A., McGee, P., & Seike, A., 2011. Population aging: facts, challenges, and responses. Benefits and Compensation International, 41(1), 22. Benefits and compensation international 41(1).

[3] Casilari, E., Santoyo-Ramón, J.A., Cano-García, J.M., 2017. Analysis of public datasets for wearable fall detection systems. Sensors (Switzerland) 17. doi:10.3390/s17071513.

[4] Dai, J., Bai, X., Yang, Z., Shen, Z., Xuan, D., 2010. PerFallD: A pervasive fall detection system using mobile phones, in: 2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops, PERCOM Workshops 2010, pp. 292–297. doi:10.1109/PERCOMW.2010.5470652.

[5] De Cillisy, F., De Simioy, F., Guidoy, F., Incalzi, R.A., Setolay, R., 2015. 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, pp. 3727–3730. doi:10.1109/EMBC.2015.7319203.

[6] De Quadros, T., Lazzaretti, A.E., Schneider, F.K., 2018. A Movement Decomposition and Machine Learning-Based Fall Detection System Using Wrist Wearable Device. IEEE Sensors Journal 18, 5082–5089. doi:10.1109/JSEN.2018.2829815.

[7] DE SOUZA, L.N.N., DE CARVALHO, P.H.B., Ferreira, M.E.C., 2018. Quality of life and subjective well-being of physically active elderly people: A systematic review. Journal of Physical Education and Sport 18, 1615–1623. URL: www.efsupit.ro, doi:10.7752/jpes.2018.03237.

[8] Dorado, J., del Toro, X., Santofimia, M.J., Parreño, A., Cantarero, R., Rubio, A., Lopez, J.C., 2019. A computer-vision-based system for at-home rheumatoid arthritis rehabilitation. International Journal of Distributed Sensor Networks 15. URL: https://us.sagepub.com/en-us/nam/open-access-at-sage, doi:10.1177/1550147719875649.

[9] Eyobu, O.S., Han, D.S., 2018. Feature representation and data augmentation for human activity classification based on wearable IMU sensor data using a deep LSTM neural network. Sensors (Switzerland) 18. doi:10.3390/s18092892.

[10] Kwolek, B., Kepski, M., 2014. Human fall detection on embedded platform using depth maps and wireless accelerometer. Computer Methods and Programs in Biomedicine 117, 489–501. URL: https://linkinghub.elsevier.com/retrieve/pii/S0169260714003447, doi:10.1016/j.cmpb.2014.09.005.

[11] Luštrek, M., Kaluža, B., 2009. Fall detection and activity recognition with machine learning. Informatica (Ljubljana) 33, 205–212.

[12] MacLachlan, M., . Shapes:smart and healthy ageing through people engaging in supportive systems. h2020 under grant agreement no. 857159. URL: https://shapes2020.eu/.

[13] Mastorakis, G., Makris, D., 2014. Fall detection system using Kinect's infrared sensor. Journal of Real-Time Image Processing 9, 635–646. doi:10.1007/s11554-012-0246-9.

[14] Steffan, L., Kaul, L., Asfour, T., 2017. Online stability estimation based on inertial sensor data for human and humanoid fall prevention, in: IEEE-RAS International Conference on Humanoid Robots, IEEE. pp. 171–177. doi:10.1109/HUMANOIDS.2017.8239553.

[15] Trkov, M., Chen, K., Yi, J., Liu, T., 2015. Slip detection and prediction in human walking using only wearable inertial measurement units (IMUs), in: IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM, Institute of Electrical and Electronics Engineers Inc.. pp. 854–859. doi:10.1109/AIM.2015.7222645.

[16] Tsinganos, P., Skodras, A., 2017. A smartphone-based fall detection system for the elderly. International Symposium on Image and Signal Processing and Analysis, ISPA , 53–58doi:10.1109/ISPA.2017.8073568.

[17] World Health Organization, . Falls. URL: https://www.who.int/news-room/fact-sheets/detail/falls.

[18] Wu, F., Zhao, H., Zhao, Y., Zhong, H., 2015. Development of a wearable-sensor-based fall detection system. International Journal of Telemedicine and Applications 2015. doi:10.1155/2015/576364.

[19] Zerrouki, N., Harrou, F., Houacine, A., Sun, Y., 2017. Fall detection using supervised machine learning algorithms: A comparative study. Proceedings of 2016 8th International Conference on Modelling, Identification and Control, ICMIC 2016 , 665–670doi:10.1109/ICMIC.2016.7804195.

[20] Ťupa, O., Procházka, A., Vyšata, O., Schätz, M., Mareš, J., Vališ, M., Mařík, V., 2015. Motion tracking and gait feature estimation for recognising Parkinson's disease using MS Kinect. BioMedical Engineering Online 14. doi:10.1186/s12938-015-0092-7.