Studies in Systems, Decision and Control 273

Hiram Ponce
Lourdes Martínez-Villaseñor
Jorge Brieva
Ernesto Moya-Albor *Editors*

Challenges and Trends in Multimodal Fall Detection for Healthcare

EXTRAS ONLINE

Springer

Studies in Systems, Decision and Control

Volume 273

Series Editor

Janusz Kacprzyk, Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

The series "Studies in Systems, Decision and Control" (SSDC) covers both new developments and advances, as well as the state of the art, in the various areas of broadly perceived systems, decision making and control–quickly, up to date and with a high quality. The intent is to cover the theory, applications, and perspectives on the state of the art and future developments relevant to systems, decision making, control, complex processes and related areas, as embedded in the fields of engineering, computer science, physics, economics, social and life sciences, as well as the paradigms and methodologies behind them. The series contains monographs, textbooks, lecture notes and edited volumes in systems, decision making and control spanning the areas of Cyber-Physical Systems, Autonomous Systems, Sensor Networks, Control Systems, Energy Systems, Automotive Systems, Biological Systems, Vehicular Networking and Connected Vehicles, Aerospace Systems, Automation, Manufacturing, Smart Grids, Nonlinear Systems, Power Systems, Robotics, Social Systems, Economic Systems and other. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution and exposure which enable both a wide and rapid dissemination of research output.

** Indexing: The books of this series are submitted to ISI, SCOPUS, DBLP, Ulrichs, MathSciNet, Current Mathematical Publications, Mathematical Reviews, Zentralblatt Math: MetaPress and Springerlink.

More information about this series at http://www.springer.com/series/13304

Hiram Ponce · Lourdes Martínez-Villaseñor ·
Jorge Brieva · Ernesto Moya-Albor
Editors

Challenges and Trends in Multimodal Fall Detection for Healthcare

Springer

Editors
Hiram Ponce
Facultad de Ingeniería
Universidad Panamericana
Mexico City, Mexico

Lourdes Martínez-Villaseñor
Facultad de Ingeniería
Universidad Panamericana
Mexico City, Mexico

Jorge Brieva
Facultad de Ingeniería
Universidad Panamericana
Mexico City, Mexico

Ernesto Moya-Albor
Facultad de Ingeniería
Universidad Panamericana
Mexico City, Mexico

ISSN 2198-4182 ISSN 2198-4190 (electronic)
Studies in Systems, Decision and Control
ISBN 978-3-030-38747-1 ISBN 978-3-030-38748-8 (eBook)
https://doi.org/10.1007/978-3-030-38748-8

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This book presents challenging issues and current trends for designing human fall detection and classification systems, as well as healthcare technologies, using multimodal approaches. In healthcare, falls are frequent especially among elderly people and it is considered a major health problem worldwide. Recently, fall detection and classification systems have been proposed to address this problem, and to reduce the time, a person fallen receives assistance. For a comprehensive perspective of these healthcare technologies, the book is divided into two parts.

In the first part, human fall detection and classification systems are approached. A holistic view, from the design process to the implementation, is considered in the self-contained chapters presented. Moreover, these contributions mainly correspond to the challenges, methodologies adopted and results of the international competition, namely *Challenge UP—Multimodal Fall Detection* that was held during the International Joint Conference on Neural Networks (IJCNN) in 2019. Throughout this part, many of the chapters include open coding. This gives readers and practitioners the opportunity to be involved in the fall detection and classification problem by hands-on experience. First, chapter "Open Source Implementation for Fall Classification and Fall Detection Systems" presents a public multimodal dataset for fall detection and classification systems, namely UP-Fall detection. This dataset was part of the above-mentioned competition; thus, a concise tutorial on how to manipulate and analyze it, as well as how to train classification models and evaluate those using the dataset, for classification systems, is presented. In chapter "Detecting Human Activities Based on a Multimodal Sensor Data Set Using a Bidirectional Long Short-Term Memory Model: A Case Study," authors propose a deep learning model using bidirectional long short-term memory (Bi-LSTM) to detect five different types of falls using a dataset provided by the *Challenge UP* competition. The work corresponds to authors that won the third place. In contrast, chapter "Intelligent Real-Time Multimodal Fall Detection in Fog Infrastructure Using Ensemble Learning" presents a proposed methodology for conducting human fall detection near real time by reducing the processing latency. This approach considers distributing the fall detection chain over different levels of computing: cloud, fog, edge and mist. In addition, chapter "Wearable Sensors

Data-Fusion and Machine-Learning Method for Fall Detection and Activity Recognition" presents a method for fall detection and classification using the UP-Fall detection dataset. The authors present an interesting approach performing unsupervised similarity search in order to find the most similar users to the ones in test set, helping for parameter tuning. These authors won the first place in the *Challenge UP* competition. In contrast to the above sensor-based approaches, in chapter "Application of Convolutional Neural Networks for Fall Detection Using Multiple Cameras," authors present a fall detection system using a 2D convolutional neural network (CNN) evaluating independent information of two monocular cameras with different viewpoints, using the public UP-Fall detection dataset. The results obtained show that the proposed approach detects human falls with high accuracy, and it has comparable performance to a multimodal approach. Lastly, chapter "Approaching Fall Classification Using the UP-Fall Detection Dataset: Analysis and Results from an International Competition" presents the results of the competition and the lessons learned during this experience. In addition, it discusses trends and issues on human fall detection and classification systems.

On the other hand, the second part comprises a set of review and original contributions in the field of multimodal healthcare. These works present trends on ambient assisted living and health monitoring technologies considering the user-centered approach.

Chapter "Classification of Daily Life Activities for Human Fall Detection: A Systematic Review of the Techniques and Approaches" reviews the techniques and approaches employed to device systems to detect unintentional falls. The techniques are classified based on the approaches employed and the used sensors and noninvasive vision-based devices. In chapter "An Interpretable Machine Learning Model for Human Fall Detection Systems Using Hybrid Intelligent Models," authors propose a fall detection system based on intelligent techniques using feature selection techniques and fuzzy neural networks. The authors highlight the importance of feature selection techniques to improve the performance of hybrid models. The main goal was to extract knowledge through fuzzy rules to assist in the fall detection process. In chapter "Multi-sensor System, Gamification, and Artificial Intelligence for Benefit Elderly People," authors present a multi-sensory system into a smart home environment and gamification to improve the quality life of elderly people, i.e., avoiding social isolation and increasing physical activity. The proposal comprises a vision camera and a voice device, and artificial intelligence is used in the data fusion. Lastly, chapter "A Novel Approach for Human Fall Detection and Fall Risk Assessment" proposes a noninvasive fall detection system based on the height, velocity, statistical analysis, fall risk factors and position of the subject from depth information through cameras. The system is then adaptable to the physical conditions of the user.

We consider this book useful for anyone who is interested in developing human fall detection and classification systems and related healthcare technologies using multimodal approaches. Scientists, researchers, professionals and students will gain understanding on the challenges and trends on the field. Moreover, this book is also attractive to any person interested in solving signal recognition, vision and machine learning challenging problems given that the multimodal approach opens many experimental possibilities in those fields.

Lastly, the editors want to thank Universidad Panamericana for all the support given to this publication and the related research project that includes the organization of the international competition and the creation of the public dataset. The editors also want to thank Editor Thomas Ditzinger (Springer) for his valuable feedback and recognition to this work.

Mexico City, Mexico
November 2019

Hiram Ponce
Lourdes Martínez-Villaseñor
Jorge Brieva
Ernesto Moya-Albor

Acknowledgement This edited book has been funded by Universidad Panamericana through the grant “Fomento a la Investigación UP 2018,” under project code UP-CI-2018-ING-MX-04.

Contents

Reviews and Trends on Multimodal Healthcare

Wearable Sensors Data-Fusion and Machine-Learning Method for Fall Detection and Activity Recognition

Hristijan Gjoreski, Simon Stankoski, Ivana Kiprijanovska, Anastasija Nikolovska, Natasha Mladenovska, Marija Trajanoska, Bojana Velichkovska, Martin Gjoreski, Mitja Luštrek and Matjaž Gams

Abstract Human falls are common source of injury among the elderly, because often the elderly person is injured and cannot call for help. In the literature this is addressed by various fall-detection systems, of which most common are the ones that use wearable sensors. This paper describes the winning method developed for the Challenge Up: Multimodal Fall Detection competition. It is a multi-sensor data-fusion machine-learning method that recognizes human activities and falls using 5 wearable inertial sensors: accelerometers and gyroscopes. The method was evaluated on a dataset collected by 12 subjects of which 3 were used as a test-data for the challenge. In order to optimally adapt the method to the 3 test subjects, we performed an unsupervised similarity search—that finds the three most similar users to the three users in the test data. This helped us to tune the method and its parameters to the 3 most similar users as the ones used for the test. The internal evaluation on the 9 users showed that with this optimized configuration the method achieves 98% accuracy. During the final evaluation for the challenge, our method achieved the highest results (82.5% F1-score, and 98% accuracy) and won the competition.

Keywords Activity recognition · Fall detection · Accelerometers · Machine learning · Wearable sensors

Electronic Supplementary Material The online version of this chapter (https://doi.org/10.1007/978-3-030-38748-8_4) contains supplementary material, which is available to authorized users.

H. Gjoreski (✉) · S. Stankoski · I. Kiprijanovska · A. Nikolovska · N. Mladenovska · M. Trajanoska · B. Velichkovska
Faculty of Electrical Engineering and Information Technologies, Ss Cyril and Methodius University, Skopje, North Macedonia
e-mail: hristijang@feit.ukim.edu.mk

M. Gjoreski · M. Luštrek · M. Gams
Jožef Stefan Institute, Ljubljana, Slovenia

H. Ponce et al. (eds.), *Challenges and Trends in Multimodal Fall Detection for Healthcare*, Studies in Systems, Decision and Control 273,
https://doi.org/10.1007/978-3-030-38748-8_4

1 Introduction

Human falls are critical health-related problems for the elderly [14], and the statistics show that approximately 30% of people over the age of 65 fall each year, and this proportion increases to 40% in those aged more than 70 (Gillespie et al. [10]. According to Worth Health Organization [32] about 20% of the elderly who fall require medical attention. Furthermore, falls and the fear of falling are important reasons for nursing-home admission [28]. Falls are particularly critical when the elderly person is injured and cannot call for help. These reasons, combined with the increasing accessibility and miniaturization of sensors and microprocessors, are driving the development of fall-detection (FD) systems.

Fall detection has received significant attention in recent years, however it still represents a challenging task [13]. The Challenge UP: Multimodal Fall Detection competition[1] presents a great opportunity the activity-recognition community to test and compare their approaches. The goal of the challenge is to recognize, as accurately as possible, 12 activities, including 5 falls.

This paper describes the method that we developed for the competition.[2] It is a multi-sensor data-fusion machine-learning method that recognizes human activities and falls using 5 accelerometers and 5 gyroscopes. It includes several steps: data preprocessing, data segmentation, sensor orientation correction, feature extraction, feature selection, hyperparameter optimization, and training a machine learning model.

The evaluation was performed on a dataset provided by the organizers of the competition. It consists of wearable sensors data collected by 12 subjects of which 3 were used as a test data for the challenge. Our method was ranked first, achieving highest recognition performance: 82.5% F1-score, and 98% accuracy.

The rest of the paper is organized as follows. The dataset is explained in Sect. 2, whereas section three is dedicated to explaining the methodology of our system. In the description of the methodology we discuss the preprocessing applied to our data, the sensor orientation correction, as well as the feature extraction and feature selection procedures. In Sect. 4 we focus on the evaluation methods for the pipeline, and in Sect. 5 we conclude the paper.

2 Related Work

Activity recognition (AR) and fall detection (FD) approaches can be divided into those that use wearable and non-wearable sensors, respectively. The most common

[1]The Challenge Up Multimodal Competition, available at: https://sites.google.com/up.edu.mx/challenge-up-2019/overview.

[2]The code developed for the challenge is available at: https://github.com/challengeupwinner/challengeupcode.

non-wearable approach is based on cameras [34]. Video-based human activity recognition is a hot research area in computer vision to help people with special needs. Miguel et al. [23] developed a computer-vision based system to recognize abnormal activity in daily life in a supportive home environment. The system tracked activity of subjects and summarized frequent active regions to learn a model of normal activity. It detected falling events as abnormal activity, which is very important in-patient monitoring systems. Although this approach is physically less intrusive for the user compared to one based on wearable sensors, it suffers from problems such as target occlusion, time-consuming processing and privacy concerns.

The most mature approach to both AR and FD is probably using wearable accelerometers, [4, 15, 17, 27]. The most common accelerometer-based AR approach uses machine learning. Typically, a sliding window passes over the stream of sensor data, and data in each window are classified with one of the known classification methods, such as decision trees (DTs) and support vector machines (SVM). The most frequent AR task is classifying activities in relation to movement, e.g., walking, running, standing still and cycling [17, 27].

An alternative approach to accelerometer-based AR is based on manually created rules [20]. These rules are usually based on features that are calculated from sensor orientations and accelerations. Bourbia et al. [4] presented an approach in which decision rules are used to recognize activities. Another implementation of such rules was presented by Lai et al. [21], who used six accelerometers, placed on the neck, waist, both wrists and both thighs and reported accuracy of 99.5%.

Fall detection has also been addressed in related studies [22]. Some of the first studies include Williams et al. [31] and Doughty et al. [7]. In this approaches the fall is detected by detecting a change in body orientation from upright to lying immediately after a large negative acceleration. Later, this algorithm was upgraded and fine-tuned by Aziz et al. [2] and Putra et al. [24, 25].

Degen et al. [6] presented a fall detector worn on the wrist that incorporates a multi-stage fall detection algorithm. The first condition is the detection of a high velocity towards the ground. Next, an impact needs to be detected within 3 s. After impact, the activity is observed for 60 s, and if at least 40 s of inactivity are recorded, an alarm is activated. The results show no false alarms, but large percentage of backwards and sideways falls were not detected.

The most common approaches to FD are rules that use thresholds applied to accelerations and features derived from them. Ren et al. [26] proposed personalized and adaptive threshold model and showed that accuracy increases for 1–3% compared to other threshold models. Wu et al. [33] developed a fall detection system based on a single, triaxial, accelerometer, which results showed lower sensitivity and specificity compared to multi-sensor approaches. Hardjianto et al. [18] used an accelerometer on smartphone, with six variations of the placement of the device. The method used for fall detection is threshold method applying only one parameter, the value of resultant acceleration. It resulted in 98.1% of accuracy, 96.9% of sensitivity, and 100 specificity.

In recent years there are also approaches that use machine learning instead of threshold-based algorithms for FD. Putra et al. [24, 25] proposed an event-triggered machine learning approach that aligns each fall stage so that the characteristic features of the fall stages are more easily recognized. The F1-score reached by the chest-worn sensor is 98% and 92% for the waist-worn sensor.

3 Dataset

The dataset provided for the competition includes 12 activities, performed by 12 subjects. The data from 9 of the subjects were released for training the models, the data from the remaining 3 subjects were used for final evaluation of the competitors. The subjects performed 7 simple human daily activities (walking, standing, sitting, picking up an object, jumping, laying, on knees) and 5 types of falls (falling forward using hands, falling forward using knees, falling backwards, falling sideward, falling sitting in empty chair). The distribution of the data according to the activities is shown in Fig. 1.

The dataset was recorded using multiple types of sensors, i.e. wearable sensors, ambient sensors and vision devices. The wearable sensors were located in the left wrist, under the neck, at right pocket of pants, at the middle of waist (in the belt), and in the left ankle. Each of these sensors contains 3-axis accelerometer, 3-axis gyroscope and a sensor for ambient light. Also, one electroencephalograph, located at the forehead, was used for measuring the brainwave signals. The ambient sensors include six infrared sensors placed above the floor of the room, and all of them reported changes in interruption of the optical devices. Lastly, the dataset was enriched with images from two cameras, which captured the subjects while doing the activities. The sampling rate of the sensors used in the dataset is 20 Hz.

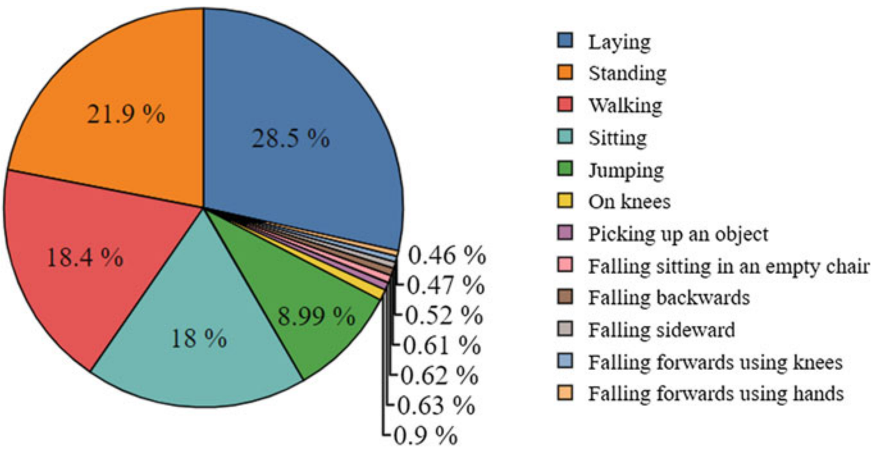

Fig. 1 The distribution of the data according to the activities

4 Method

The method that we developed for this study is shown in Fig. 2. It includes data preprocessing (filters, data segmentation—sliding window), feature engineering and extraction, feature selection, and finally a classification model to recognize the activity. Each of the steps are described in the following subsections.

4.1 Data Preprocessing

Signal segmentation is very important step in the activity recognition process. Therefore, we segmented the sensor signals using a sliding window size of 0.5 s with a 0.25 s overlap. This way the model recognizes activity every 0.25 s. The window size and the sliding factor are important in data processing and have to be tuned correctly for the task at hand. Longer windows naturally contain more data and are expected to enable greater classifying accuracy, especially for more complex activities. Shorter windows, on the other hand, make it possible to detect activity changes faster. Considering the fact that we aim to achieve accurate fall detection and falls last shorter than one second on average in this dataset, the window size had to be chosen so that it is small enough in relation to average fall duration. The optimal window size in our experiments was determined empirically.

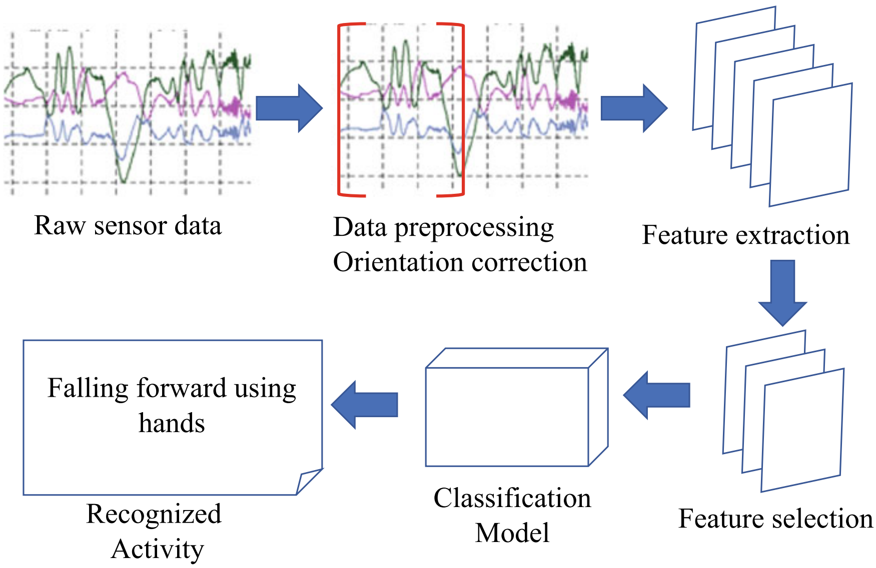

Fig. 2 Activity recognition and fall detection pipeline

Beside the raw sensor signals (x, y and z axis) we additionally extracted the magnitude of the acceleration vector. It was calculated for the accelerometer, as well as the gyroscope and is shown in (1).

$$m = \sqrt{x^2 + y^2 + z^2} \tag{1}$$

4.2 Orientation Correction

After analyzing the data, we noted that the orientation of the sensors varies between users, and even more between different trials. Therefore, we have developed a method that corrects the orientation of the sensors, i.e., it uses rotation matrices to correct (rotate) the accelerometers data. The method corrects accelerometer axes orientation by applying a rotation transformation to the device's raw data [16]. To calculate the angle between the actual acceleration (e.g. the Earth's gravity (g) for static activities) and some of the axis (e.g., x-axis) we used the formula shown in (2)—where the values a_x, a_y and a_z represent the actual acceleration vector.

$$\varphi_x = \arccos\left(\frac{a_x}{\sqrt{a_x^2 + a_y^2 + a_z^2}}\right) \tag{2}$$

The coordinate system is rotated using trigonometry and rotational matrices, in such a way that it corrects the data. In order to do this, one should calculate the difference between the expected angle (φ_x) and the rotated angle (φ_{xr}). This difference gives the angle by which the coordinate system should be rotated in order to correct the orientation of the sensor. The rotation is performed by a rotation matrix, which describes a rotation of a coordinate system with respect to another orientation. An acceleration vector in the initial reference frame can be transformed into a vector in a rotated frame by multiplication of the initial vector with the rotation matrix [9]. In three dimensions, rotations are possible around the three principal axes (x, y and z).

To achieve this, we used the *standing* activity as a reference in order to compute the current angle φ_{xr}. This kind of reference angles (orientation vectors) are defined for all accelerometers (neck, wrist, belt, right pocket and ankle). As a reference angle (φ_x) we used the angle under which the sensors of our referent subject are placed. The method then calculates the rotation angle for every other subject in the dataset with respect to the referent subject. Once it is calculated, all raw accelerometer data thereafter are multiplied by the rotation matrix to achieve the corrected orientation.

4.3 Feature Extraction and Selection

In order to extract as much features as possible, we used the TSFRESH library. It performs time series feature extraction and selection, which we exploited in generating approximately 12,000 features. In the next step we performed feature selection in order to reduce the number of features and keep only the most relevant ones. We focused on removing those features which did not contribute to the accuracy of our model and/or increased the odds for overfitting [19].

In the first step, we discarded the features containing missing and Not-a-Number (NaN) values, which resulted in leaving 7700 features. Then, we estimated the mutual information between each feature and the label (activity). We sorted the features in descending order according to this value, as the higher the mutual information, the higher the dependency of the label from the corresponding feature. In the next step, we divided the features in groups of 100. We began with the first group of features, where we calculated the Pearson correlation coefficient for every pair of features. If the correlation between a pair exceeded a threshold of 0.8, out of the two we removed the feature with the lower mutual information. To the remaining features of the group we appended the following group of 100 features. The process was repeated until all the initial groups of 100 were iterated.

Finally, we selected the definite set of features using a wrapper feature selection algorithm. Here, the first step was to utilize the best scoring feature in regard to the value of its mutual information with the label and train a classification model to estimate the macro F1-score. Then, in every following step, the next feature of the uncorrelated features was added to the previously kept features. Once the feature was added, the model was retrained and a new F1 score obtained. If, at each step, the score decreased not more than 1%, the newly added feature was kept. Otherwise, the feature was dismissed, making the feature list before and after said step unchanged. This measure, repeated for all the remaining features, allowed us to take into consideration every wearable sensor and at the same time prevented us from overfitting our model. The final feature selection resulted in 152 relevant, uncorrelated, class-defining features.

4.4 Classification

We compared three machine learning algorithms: *Decision Tree*, *XGBoost*, and *Random Forest*. After thorough evaluation and comparison (see the results in Sect. 5.3), we have chosen the best performing one, i.e., *Random Forest*. This algorithm showed more robust performance when tested in different scenarios and different users. In the following paragraphs each of the algorithms is described in relation to our activity recognition task.

Decision Tree [29] is an algorithm that learns a model in a form of a tree structure. In particular, it divides the dataset into smaller subsets while at the same time the decision tree is incrementally learned. The final result is a tree with decision nodes with two or more branches, each representing values for the feature tested, and leaf nodes which represent a decision on the activity. In our case, all of the features are numeric (this means the same feature can be used multiple times), which resulted in very large trees.

XGBoost [5] is efficient implementation of the gradient boosted trees algorithm. It is a supervised learning algorithm, which predicts the activity by combining the estimates of a set of simpler, weaker models—in our case decision trees models. It uses a gradient descent algorithm to minimize the loss when adding new models. This way, it minimizes an objective function that combines a convex loss function and a penalty term for model complexity. The training proceeds iteratively, adding new trees that predict the errors of prior trees that are then combined with previous trees to make the final prediction of the activity.

Random Forest [11] is ensemble of decision tree classifiers. During training, the Random Forest algorithm creates multiple decision trees each trained on a bootstrapped sample of the original training data and searches only across a randomly selected subset of the input variables to determine a split (for each node). For classification, each tree in the Random Forest predicts the activity, and the final output of the classifier is determined by a majority vote of the trees. This way, the activity that is predicted by most of the decision trees will be chosen as final.

4.5 Hyperparameter Optimization

In the final step, we performed a hyperparameter optimization for each of the 3 algorithms explained in the previous subsection. Hyperparameter optimization is a process of finding a set of optimal hyperparameters for a learning algorithm, where a hyperparameter is a parameter whose value is used to control the learning process. Finally, this optimization finds a tuple of hyperparameters that yields an optimal model which minimizes the error function (maximizes the accuracy) given a dataset.

There are different methods for optimizing hyperparameters: Grid Search; Random Search, Bayesian optimization, Gradient-based optimization, etc. We chose Random Search method as it is one of the most commonly used methods for hyperparameter optimization in time-series data and showed to be more robust compared to the other techniques [3]. Random Search replaces the exhaustive enumeration of all combinations by selecting them randomly. This can be simply applied to the discrete setting, but also generalizes to continuous and mixed spaces. It usually outperforms Grid search, especially when only a small number of hyperparameters affects the final performance of the machine learning algorithm—which was the case in our study. Additionally, Random search is more efficient compared to Grid search, which spends too much time evaluating unpromising regions of the hyperparameter search space because it has to evaluate every single combination in the grid. Random search in contrast, does a better job of exploring the search space and therefore can usually find a good combination of hyperparameters in far fewer iterations [3].

The following hyperparameters were optimized:

- Decision Tree: Maximum number of levels in tree; Minimum number of samples required to split a node; Minimum number of samples required at each leaf node;
- Random Forest: Number of trees in random forest; Number of features to consider at every split; Maximum number of levels in tree; Minimum number of samples required to split a node; Minimum number of samples required at each leaf node;
- XGBoost: The learning rate; Minimum child weight; Number of estimators; Minimum number of samples required to split a node; Minimum number of samples required at each leaf node; Maximum depth.

5 Evaluation

5.1 Dataset Split

In order to optimally adapt the method to the 3 test users, we have performed an unsupervised similarity search—that finds the three most similar users to the three users in the test data. This helped us to tune the method and its parameters to the 3 most similar users as the ones used for the test.

The method uses each user's data individually and performs a K-means clustering, where K is the number of classes/activities, i.e., we used 6 (all the falls are similar and therefore we grouped them). After performing the clustering, then we calculated the centroid for each cluster, which resulted in 6 centroids per user. Then, we calculated a distance matrix that contained the distances between the 6 clusters of the train user, and the 6 centroids from the test user. We calculated this matrix for each pair of users, i.e., we calculated 27 distance matrices (the 9 users in train vs the 3 users in test). For each matrix we have calculated the *distance between the pair of users*, i.e., we calculated the minimum sum of the distances that covers all the 6 clusters. This way we were able to find the 3 most similar (minimal distance) users to the ones used for the test.

The most similar subjects to the test users: 15, 16 and 17, are: 4, 3 and 13 respectively.

5.2 Evaluation Metrics

Accuracy is the most commonly used metric that can be calculated from a confusion matrix. Its main drawback is that it hides information on the specific nature of errors (the proportions of FP and FN) [30]. It is calculated as following:

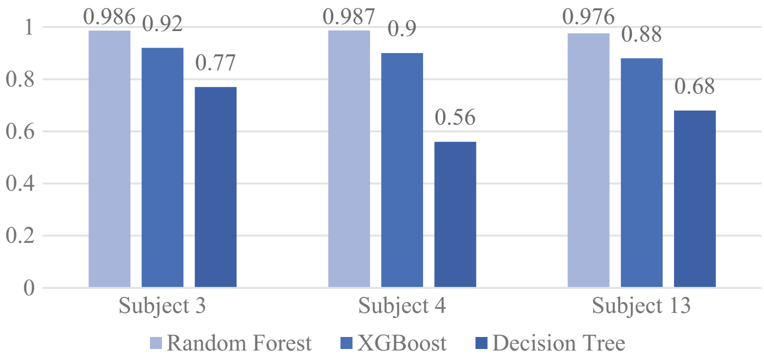

Fig. 3 Accuracy comparison

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \quad (3)$$

We assessed the performance of the model by not only using the accuracy, but also macro F1-score (F1-macro). The F1-macro is the unweighted mean of the F1-scores for the different labels [1]. It can be calculated as harmonic mean between precision and recall, where the average is calculated per label and then averaged across all labels. If Pi and Ri are the precision and recall for each label, then the F1-macro is calculated as in (4):

$$F1\text{-}macro = \frac{1}{Q}\sum_{i=1}^{Q}\frac{2 * Pi * Ri}{Pi + Ri} \quad (4)$$

5.3 Algorithm Comparison

A summary of the results is shown in Figs. 3 and 4, which shows that the system successfully recognized the activities using optimized Random Forest classifier, with high accuracy (97–99%), and F1 macro score (84–90%). The results using other classifiers are significantly worse.

5.4 Confusion Matrices

The following 4 confusion matrices show the performance achieved for the 3 users summarized (Table 2), and each of the users individually (Tables 3, 4 and 5). The

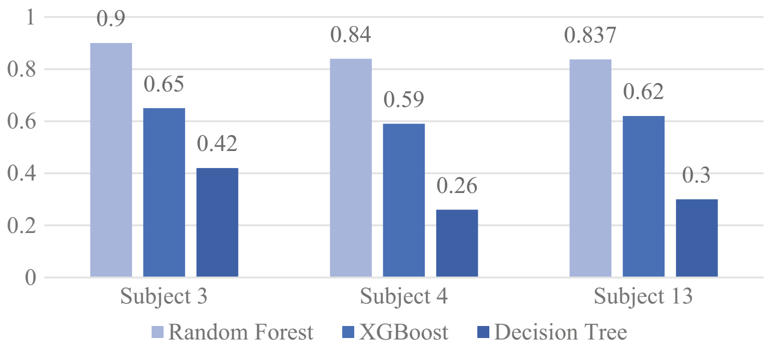

Fig. 4 Comparison macro F1-score

Table 1 Activities and their corresponding IDs

Activity ID	Description
1	Falling forward using hands
2	Falling forward using knees
3	Falling backwards
4	Falling sideward
5	Falling sitting in empty chair
6	Walking
7	Standing
8	Sitting
9	Picking up an object
10	Jumping
11	Laying
12	On knees

IDs of the activities correspond to Table 1. Note that in the training data the *on knees* activity is missing and therefore it is omitted in the results.

The results show that activities 6, 7, 8, 9, 10, 11 (walking, standing, sitting, picking up an object, jumping and lying) are correctly recognized most of the time. Some problem occurs with falling activities, but most likely this is due to the small number of instances and the impossibility of the model to be enough trained on them.

5.5 Challenge UP Competition Final Results

The confusion matrix obtained by the final evaluation during the competition is presented in Table 6. The overall results show that our method achieved 82.5% F1-macro, and 98% accuracy. Although the resulting matrix looks generally satisfying, it

Table 2 Confusion matrix for the 3 users (Subject 3, Subject 4, Subject 13)

Activity	1	2	3	4	5	6	7	8	9	10	11
1	23	1	1	1	0	2	3	0	1	0	0
2	1	23	1	3	0	3	0	0	1	1	1
3	0	1	33	3	3	1	2	0	0	1	3
4	0	0	4	42	0	0	3	0	0	0	1
5	2	0	3	3	27	0	2	0	0	0	0
6	0	0	0	0	0	1909	3	0	7	0	0
7	0	0	0	1	0	34	2357	0	3	2	0
8	0	0	0	0	0	0	0	1902	4	0	29
9	0	0	0	0	2	0	1	0	56	0	3
10	1	0	0	0	0	8	7	0	0	943	0
11	0	0	0	2	1	0	0	0	0	0	1021

Table 3 Confusion matrix for User 3

Activity	1	2	3	4	5	6	7	8	9	10	11
1	9	0	0	0	0	0	2	0	1	0	0
2	1	14	0	2	0	2	0	0	0	1	1
3	0	0	15	0	0	1	2	0	0	0	3
4	0	0	0	15	0	0	3	0	0	0	1
5	0	0	3	0	20	0	0	0	0	0	0
6	0	0	0	0	0	657	0	0	0	0	0
7	0	0	0	1	0	7	759	0	0	2	0
8	0	0	0	0	0	0	0	642	4	0	0
9	0	0	0	0	0	0	0	0	23	0	0
10	0	0	0	0	0	2	4	0	0	309	0
11	0	0	0	2	1	0	0	0	0	0	1021

is noticeable that the biggest issue is the second activity—*falling forward using knees*. Almost half of the instances that belong to this activity are classified as *standing* by our model. We speculate that the reason for this is the lack of instances which represent this activity. Another issue is the imperfection of data labeling. The activity *falling forward using knees* consists of two parts: first standing and then kneeling. It is possible that much of the standing may be labeled as falling due to too little available time.

The rest of the activities are recognized with much higher accuracy. The activities *jumping* and *sitting* are recognized with 100%, which is due to the dissimilarity to any other activity. The other three activities recognized with 100% are: *standing*, *falling backwards* and *falling sideward*, probably because of the orientation correction procedure. We speculate that due to the orientation correction our model was able to

Table 4 Confusion matrix for User 4

Activity	1	2	3	4	5	6	7	8	9	10	11
1	6	0	0	0	0	1	1	0	0	0	0
2	0	4	0	1	0	0	0	0	1	0	0
3	0	0	9	0	2	0	0	0	0	1	0
4	0	0	0	8	0	0	0	0	0	0	0
5	2	0	0	2	4	0	2	0	0	0	0
6	0	0	0	0	0	645	0	0	7	0	0
7	0	0	0	0	0	6	772	0	3	0	0
8	0	0	0	0	0	0	0	640	0	0	0
9	0	0	0	0	1	0	0	0	23	0	0
10	1	0	0	0	0	2	1	0	0	330	0
11	0	1	0	2	0	0	0	0	1	7	1037

Table 5 Confusion matrix for User 13

Activity	1	2	3	4	5	6	7	8	9	10	11
1	8	1	1	1	0	1	0	0	0	0	0
2	0	5	1	0	0	1	0	0	0	0	0
3	0	1	9	3	1	0	0	0	0	0	0
4	0	0	4	19	0	0	0	0	0	0	0
5	0	0	0	1	3	0	0	0	0	0	0
6	0	0	0	0	0	607	3	0	0	0	0
7	0	0	0	0	0	21	826	0	0	0	0
8	0	0	0	0	0	0	0	620	0	0	29
9	0	0	0	0	1	0	1	0	10	0	3
10	0	0	0	0	0	4	2	0	0	304	0
11	1	1	0	1	0	0	0	0	0	0	971

successfully distinguish different falls based on the correct acceleration direction. The final activity *in knees* was poorly recognized, probably due to the short duration (few seconds) and the lack of this activity in the training data.

6 Conclusion

The paper presented the winning ML method of the Challenge Up: Multimodal Fall Detection competition. The method is tuned for robustness and real-time performance by combining multiple wearable inertial sensors: accelerometer and gyroscope, in

Table 6 Confusion matrix—challenge up: multimodal fall detection final results

Activity	1	2	3	4	5	6	7	8	9	10	11	12	Recall
1	10	0	0	0	0	0	0	0	0	0	1	0	0.91
2	0	8	0	0	0	0	6	0	0	0	0	0	0.57
3	0	0	16	0	0	0	0	0	0	0	0	0	1.00
4	0	0	0	15	0	0	0	0	0	0	0	0	1.00
5	0	0	0	0	18	0	0	1	0	0	0	0	0.95
6	0	0	0	0	0	549	0	0	0	0	0	0	1.00
7	2	0	2	1	3	0	659	0	2	0	0	0	0.99
8	0	0	0	0	0	0	0	547	0	0	0	0	1.00
9	0	0	0	0	0	0	2	0	21	0	0	0	0.91
10	0	0	0	0	0	0	0	0	0	279	0	0	1.00
11	4	3	0	0	3	0	1	20	0	0	877	0	0.97
12	0	6	0	0	0	0	0	0	0	0	3	0	0.00
Precision	0.63	0.47	0.89	0.94	0.75	1.00	0.99	0.96	0.91	1.00	1.00	NaN	0.98

Accuracy = 98.03% Precision = 85.77% Recall = 79.42% F1-score = 82.47%

order to recognize activities and detect falls. It includes several steps: data preprocessing, data segmentation, sensor orientation correction, feature extraction, feature selection, hyperparameter optimization, and training a machine learning model.

During the development of the method we have noted that the orientation of the sensors varies between users, and even more between different trials. Therefore, we have developed a method that corrects the orientation of the sensors, i.e., it uses rotation matrices to correct (rotate) the accelerometers data.

We applied extensive feature extraction and selection procedure. It is a three-step procedure that selects an optimal subset of features (152 features) from the 12,000 features initially calculated from the raw sensor data.

Finally, to optimally adapt the method to the 3 test users, we have performed an unsupervised similarity search—that finds the three most similar users to the three users in the test data. This helped us to tune the method and its parameters to the 3 most similar users as the ones used for the test.

The internal evaluation on the 9 users showed that with this optimized configuration the method achieves 98% accuracy. All these steps allowed us to develop accurate fall detection and activity recognition algorithm, that achieved the highest results (82.5% F1-score, and 98% accuracy) at the competition and received the first award.

The method has several limitations. First, it uses 5 wearable sensors, which is impractical for everyday usage by an elderly person. For the future work, we plan to focus more on the practical implementation of the method into a commercial fall detection system. First, we intend to reduce the number of sensors but without losing accuracy. This way the system will be less intrusive and more user-friendly. Another improvement in this direction can be achieved by introducing specially designed clothes, which will include pockets for the sensors. Additionally, the interaction between the user and the system should be introduced by using smartphone, smartwatch, tablet or PC as a medium for showing system's notifications (fall detected, system malfunction, etc.), similar to Gjoreski et al. [12].

Acknowledgements We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. The authors declare that they have no conflict of interest.

References

1. Asch, V.V.: Macro- and micro-averaged evaluation measures. Comput. Sci. (2013)
2. Aziz, O., Musungi, M., Park, E.J., Mori, G., Robinovitch, S.N.: A comparison of accuracy of fall detection algorithms (threshold-based vs. machine learning) using waist-mounted tri-axial accelerometer signals from a comprehensive set of falls and non-fall trials. J. Med. Biol. Eng. Comput. **55**(1), 45–55 (2017)
3. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J Mach. Learn. Res. **13**, 281–305 (2012)
4. Bourbia, A.L., Son, H., Shin, B., Kim, T., Lee, D., Hyun, S.J.: Temporal dependency rule learning based group activity recognition in smart spaces (2016)
5. Chen, T., Guestrin, C.: XGBoost: A scalable tree boosting system. In: ACM SIGKDD International Conference, vol. 22, pp. 785–794 (2016)
6. Degen, T., Jaeckel, H., Rufer, M., Wyss, S.: SPEEDY: A fall detector in a wrist watch. In: Proceeding Seventh IEEE International Symposium on Wearable Computing, pp. 184–187 (2003)
7. Doughty, K., Lewis, R., McIntosh, A.: The design of a practical and reliable fall detector for community and institutional telecare. J. Telemed. Telecare **6**, S150–S154 (2000)
8. Friedman, S.M., Munoz, B., West, S.K., Rubin, G.S., Fried, L.P.: Falls and fear of falling: Which comes first? A longitudinal prediction model suggests strategies for primary and secondary prevention. J. Am. Geriatr. Soc. 1329–1335 (2000)
9. Friedman, A., Chehade, N.H., Chien, C., Pottie, G.: Estimation of accelerometer orientation for activity recognition, pp. 2076–2079. Engineering in Medicine and Biology Society (EMBC) (2012)
10. Gillespie, L.D., Robertson, M.C., Gillespie, W.J., Lamb, S.E., Gates, S., Cumming, R.G., Rowe, B.H.: Interventions for preventing falls in older people living in the community (review). The Cochrane Library, 4 (2009)
11. Gislason, P.O., Benediktsson, J.A., Sveinsson, J.R.: Random forests for land cover classification. Pattern Recogn. Lett. **27**(4), 294–300 (2006)
12. Gjoreski, H., Bizjak, J., Gams, M.: Using smartwatch as telecare and fall detection device. In: 12th International Conference on Intelligent Environments (IE), pp. 242–245 (2016a)
13. Gjoreski, M., Gjoreski, H., Luštrek, M., Gams, M.: How accurately can your wrist device recognize daily activities and detect falls? Sensors **16**(6), 800 (2016b)
14. Gjoreskic, H., Gams, M., Luštrek, M.: Context-based fall detection and activity recognition using inertial and location sensors. J. Ambient Intell. Smart Env. **6**(4), 419–433 (2014)
15. Gjoreski, H., Lustrek, M., Gams, M.: Accelerometer placement for posture recognition and fall detection. In: Seventh International Conference on Intelligent Environments, pp. 47–54 (2011)
16. Gjoreski, H., Gams, M., Lutrek, M.: Human activity recognition: from controlled lab experiments to competitive live evaluation. In: IEEE International Conference on Data Mining Workshop (ICDMW), pp. 139–145 (2015)

17. Hammerla, N.Y., Halloran, S., Plotz, T.: Deep, convolutional, and recurrent models for human activity recognition using wearables (2016)
18. Hardjianto, M., Istiyanto, J.E., Putra, A.E.: Fall detection on humans using threshold method based on smartphone accelerometer data (2017)
19. Janko, V., Gjoreski, M., Slapničar, G., Mlakar, M., Reščič, N., Bizjak, J., Drobnič, V., Marinko, M., Mlakar, N., Gams, M., Luštrek, M.: Winning the Sussex-Huawei locomotion-transportation recognition challenge. In: Kawaguchi, N., Nishio, N., Roggen, D., Inoue, S., Pirttikangas, S., Van Laerhoven, K. (eds.) Human Activity Sensing. Springer Series in Adaptive Environments. Springer, Cham (2019)
20. Kozina, S., Gjoreski, H., Gams, M., Luštrek, M.: Three-layer activity recognition combining domain knowledge and meta- classification. JMBE **33**(4), 406–414 (2013)
21. Lai, C., Huang, Y.M., Park, J.H., Chao, H.C.: Adaptive body posture analysis for elderly-falling detection with multisensors. IEEE Intell. Syst. **25**, 2–11 (2010)
22. Lustrek, M., Gjoreski, H., Vega, N.G., Kozina, S., Cvetkovic, B., Mirchevska, V., Gams, M.: Fall detection using location sensors and accelerometers. IEEE Pervas. Comput. **14**(4), 72–79 (2015)
23. Miguel, K., Brunete, A., Hernando, M., Gambao, E.: Home camera-based fall detection system for the elderly. Sensors (2017)
24. Putra, I., Brusey, J., Gaura, E., Vesilo, R.: An event-triggered machine learning approach for accelerometer-based fall detection (2017a)
25. Putra, P., Brusey, J., Gaura, E., Vesilo, R.: An event-triggered machine learning approach for accelerometer-based fall detection (2017b)
26. Ren, L., Shi, W.: Chameleon: Personalised and adaptive fall detection of elderly people in home-based environments (2016)
27. Ronao, C.A., Cho, S.-B.: Human activity recognition with smartphone sensors using deep learning neural networks. Expert Syst. Appl. **59**, 235–244 (2016)
28. Tinetti, M.E., Williams, C.S.: Falls, injuries due to falls, and the risk of admission to a nursing home. The New England J Medicine **337**, 1279–1284 (1997)
29. Wang, Y., Witten, I.H.: Induction of model trees for predicting continuous classes. Eur. Conf. Mach. Learn. **9**, 128–137 (1996)
30. Ward, J.A., Lukowicz, P., Gellersen, H.W.: Performance metrics for activity recognition. ACM transactions on intelligent systems and technology (2011)
31. Williams, G., Doughty, K., Cameron, K., Bradley D.A: A smart fall and activity monitor for telecare applications. In: Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (1998)
32. World Health Organization (WHO): Global brief for World Health Day, Good health adds life to years. (2012) http://whqlibdoc.who.int/hq/2012/WHO_DCO_WHD_2012.2_eng.pdf
33. Wu, F., Zhao, H., Zhao, Y., Zhong, H.: Development of a wearable-sensor-based fall detection system. Int. J. Telemed. Appl. (2015)
34. Zhang, S., Wei, Z., Nie, J., Huang, L., Wang, S., Li1, Z.: A review on human activity recognition using vision-based method. J Healthcare Eng (2017)