A Reliable Fall Detection System Based on Analyzing the Physical Activities of Older Adults Living in Long-Term Care Facilities

Majd Saleh, Manuel Abbas, Joaquim Prud'Homm, Dominique Somme, and Régine Le Bouquin Jeannès, *Member, IEEE*

***Abstract*— Fall detection systems are designed in view to reduce the serious consequences of falls thanks to the early automatic detection that enables a timely medical intervention. The majority of the state-of-the-art fall detection systems are based on machine learning (ML). For training and performance evaluation, they use some datasets that are collected following predefined simulation protocols i.e. subjects are asked to perform different types of activities and to repeat them several times. Apart from the quality of simulating the activities, protocol-based data collection results in big differences between the distribution of the activities of daily living (ADLs) in these datasets in comparison with the actual distribution in real life. In this work, we first show the effects of this problem on the sensitivity of the ML algorithms and on the interpretability of the reported specificity. Then, we propose a reliable design of an ML-based fall detection system that aims at discriminating falls from the ambiguous ADLs. The latter are extracted from 400 days of recorded activities of older adults experiencing their daily life. The proposed system can be used in neck- and wrist-worn fall detectors. In addition, it is invariant to the rotation of the wearable device. The proposed system shows 100% of sensitivity while it generates an average of one false positive every 25 days for the neck-worn device and an average of one false positive every 3 days for the wrist-worn device.**

***Index Terms*— Fall detection, wearable sensors, machine learning, elderly health care.**

Manuscript received June 9, 2021; revised November 19, 2021; accepted November 30, 2021. Date of publication December 7, 2021; date of current version December 21, 2021. This work was supported in part by the European Union through the European Regional Development Fund (ERDF), in part by the Ministry of Higher Education and Research, in part by the French Region of Brittany and Rennes Métropole, and in part by the French National Research Agency (ANR) in the Context of the ACCORDS under Project ANR-17-CE19-0024-01. *(Corresponding author: Régine Le Bouquin Jeannès.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Rennes CHU Ethics Committee under Approval No. 19.56.

Majd Saleh, Manuel Abbas, and Régine Le Bouquin Jeannès are with Univ Rennes, Inserm, LTSI - UMR 1099, F-35000 Rennes, France (e-mail: majdsaleh84@gmail.com; manuel.abbas@univ-rennes1.fr; regine.le-bouquin-jeannes@univ-rennes1.fr).

Joaquim Prud'Homm is with Univ Rennes, CHU Rennes, Inserm, LTSI - UMR 1099, F-35000 Rennes, France (e-mail: joaquim.prud-homm@inserm.fr).

Dominique Somme is with Univ Rennes, CHU Rennes, CNRS, ARENES - UMR 6051, F-35000 Rennes, France (e-mail: dominique.somme@chu-rennes.fr).

Digital Object Identifier 10.1109/TNSRE.2021.3133616

I. Introduction

FALLS represent a major public health problem since every year more than 37 million falls require medical intervention [1]. In addition, the health system costs from fall related injuries of older adults are substantial. Fall detection systems are designed in view to reduce the serious consequences of falls thanks to the early automatic detection that enables a timely medical intervention. These systems are based on either wearable or ambient sensors. In this paper, we focus on wearable fall detectors that can be used continuously accompanying the user all the time.

Fall detection is still an open research topic to date [2], [3]. Most of fall detection algorithms are based on machine learning (ML) like [4]–[10]. ML-based solutions require relatively large datasets for training and performance evaluation. The majority of the state-of-the-art works use datasets of simulated falls and activities of daily living (ADLs) that are collected following some predefined protocols [11]. Some examples of the datasets used to develop ML-based fall detection systems are: FallAllD [7], SisFall [12], UMA-Fall [13] and UniMiB [14]. To show the drawbacks of the state-of-the-art works that are considerably dependent on such kind of datasets, we first discuss the considered performance criteria.

ML-based fall detection is generally formulated as a binary classification problem that aims at classifying human activities into two classes: falls and ADLs. ML models are iteratively optimized, on a given training set, to maximize the classification accuracy. In other words, they implicitly aim at finding the best trade-off between sensitivity and specificity i.e. they aim at detecting as much falls as possible and avoiding false positives (FPs). In all state-of-the-art datasets that we explored [7], [12]–[15], the ADLs were collected in a protocol-based way i.e. subjects were asked to do a set of activities for a while and to repeat them several times. Hence, the distribution of ADL types in these datasets is considerably different from the actual distribution in real life. For instance, in the FallAllD dataset [7], the number of samples representing jumping is almost equal to the number of samples representing walking. This balance between the number of samples that belong to different ADL types in a given dataset is beneficial to conduct some statistical analysis. However, the aforementioned difference in the ADLs distribution between the simulated datasets and real life leads to two major issues.

The first is that maximizing the accuracy with the presence of a large percentage of highly ambiguous ADLs (e.g. jumping, stumbling without falling, …) leads to augmenting the system specificity at the expense of sensitivity. Since the ultimate objective of fall detection systems is the safety of older adults, the sensitivity raises as the most important performance criterion and decreasing it is a clear drawback. Moreover, the gain in the specificity achieved from the sensitivity/specificity trade-off might not result in a significant reduction in FPs in real life conditions since some ambiguous ADLs like jumping are rarely met in the daily life of the elderly people. **The second** drawback of the above-mentioned difference in the ADLs distribution is that the reported specificity in most of the state-of-the-art works does not reflect the performance in real world conditions. More precisely, the reported specificity does not reveal the expected number of false positives during a fixed period in real life. For example, assume we want to embed a given fall detection system that has been evaluated on a simulated dataset and reported a specificity of 99.99% in a wearable fall detector that is designed to make one decision per second (86 400 decision/day). Theoretically, the specificity of 99.99% translates to 8 false positives per day which is relatively high.

Besides the drawbacks inherited from the aforementioned gap in the ADLs' distribution, simulations in most of the state-of-the-art datasets have been executed by young subjects. On the one hand, simulating falls by young people is justified because collecting a large number of real falls is a tough task that requires long time and extensive resources. On the other hand, simulating ADLs by young subjects is not well justified since it is possible to collect ADLs from older adults. In fact, the SisFall dataset [12] that was collected using waist-worn devices tackled this problem where the ADLs were collected from 15 subjects aged over 60 years. However, like in most of the state-of-the-art datasets, ADLs were collected following a predefined protocol. Therefore, developing ML-based systems using these datasets suffer from the aforementioned problems related to the distribution of ADL types. It is worth mentioning that collecting fall events data in real-world conditions has received little attention in the literature. Besides the FARSEEING real-world fall repository [16], where data have been recorded using inertial sensors, the majority of datasets on the public domain are lab-based as previously discussed.

Our current work tackles all the above-mentioned problems. The first step in the proposed strategy is collecting a large amount of data that represents the activities of older adults in their daily life. Then, these real-world data are analyzed and a preprocessing system is proposed to extract the ambiguous ADLs (ADLs that are suspected to be falls) from the recorded data. A hybrid dataset that consists of the aforementioned ambiguous ADLs and a wide variety of simulated falls is constructed. A reliable ML-based fall detection system is then proposed where, contrary to the classical works, the objective is to discriminate falls from the real-life ambiguous ADLs. The objectives of the proposed fall detection system are summarized as follows:

- fall detection sensitivity has the highest priority. Nonetheless, the specificity is to be maximized (but not at the expense of sensitivity), since frequent false alarms may cause additional burdens on users;
- the same fall detection system could be used in neck- or wrist-worn devices;
- the fall detection system is rotation-invariant, i.e. rotating the device does not affect the system behavior;
- the ML model is to be trained on real-world ADLs of older adults collected for a long time period. The falls used for training represent a wide variety of fall types simulated by healthy young subjects;
- the fall detection algorithm is to be evaluated for a long period in order to give an accurate estimate of the number of false positives in real-world conditions.

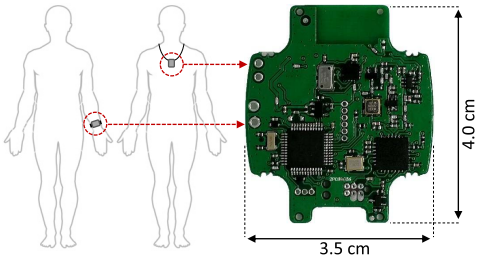

Fig. 1. The considered positions of wearable devices and the printed circuit.

The paper is organized as follows: we start by exploring the data acquisition system in Section II. Then, 400 days of real-life activities of older adults are analyzed in Section III in order to develop an efficient preprocessing system for fall detectors. The proposed ML-based fall detection algorithms are then explained in Section IV. The performance of the proposed algorithms is evaluated in Section V. Some practical considerations are discussed in Section VI before concluding the paper is Section VII.

II. Data Acquisition

We assume that motion data could be well represented by acceleration and barometric pressure signals. To this end, our partner company, RF-Track, designed and implemented the required data-loggers for collecting the motion data by means of accelerometers and barometers. In order to satisfy the acceptability requirements of the subjects, we considered two types of wearable devices: wrist-worn and neck-worn devices, as shown in Fig. 1. Although a waist-mounted device might be one of the best in terms of reliability [7], it is hardly worn under the shower (one of the most important causes of falls) or in bed. Hence, it cannot accompany the elderly everywhere to ensure continuous monitoring and thus provide timely intervention in case of a fall. The printed circuit of a data-logger is also illustrated in Fig. 1. Data-loggers were equipped with:

- the accelerometer LIS3DH which is configured to a sampling frequency of $f_s^a = 50$ Hz and a measurement range of ± 8 g;
- the barometric pressure sensor MS5611 which is configured to a sampling frequency $f_s^b = 10$ Hz;
- the micro-controller STM32L431 made by STMicroelectronics.

The employed accelerometer measures 3-D acceleration i.e. $\mathbf{a} = [a_x, a_y, a_z]$. The objective of measuring the barometric

pressure is to estimate the changes in the user's altitude ΔH which is helpful to increase the specificity of fall detectors as will be discussed in the next section. In fact, the employed barometer measures both the barometric pressure and the temperature. In order to convert the measured pressure into the corresponding altitude, the measured pressure signal is first denoised using a slope limit filter. Then, the following formula is applied [17]:

$$h = \frac{T}{0.0065} \times (1 - (\frac{P}{P_0})^{1/5.257}) \quad (1)$$

where h is the height in meters, T represents the measured temperature in kelvin, P represents the measured pressure and P_0 represents the pressure at sea level in normal conditions.

Using the above described data-loggers, motion data have been collected from 16 subjects (25% male and 75% female) aged 80 years or older and living in long-term care facilities. The height and weight of those subjects range between [142, 176] cm and [44, 86.3] kg respectively. An ethical approval that allows the collection of the elderly activity data was obtained from the Rennes CHU Ethics Committee (approval number 19.56, date of approval: 11/05/2019). In addition, the subjects have given their consent to participate in this study and to use their data for research purposes. Wrist-worn devices have been used by 9 subjects while the remaining 7 subjects used neck-worn devices. The total amount of recorded data was 400 days (214 with wrist-worn +186 with neck-worn devices). During this period, two real falls occurred. One fall has been captured by a wrist-worn device and the other one by a neck-worn device. This real world dataset will be named *RealAct* throughout this paper.

In the remainder of the paper, the lab-based dataset FallAllD [7] and the real world dataset RealAct are both considered to develop and to evaluate the performance of the proposed fall detection approach. The falls of FallAllD were simulated by relatively young subjects who wore body protection during these simulations. Although a comprehensive protocol has been set to represent all possible falls, the human reflex will prevent the subjects unconsciously from falling strongly, which leads to an attenuated impact compared to real falls. Furthermore, younger subjects are more dynamic during the pre-impact phase, and the ADLs are simulated with higher intensities compared to the movements of elderly. Therefore, RealAct is better suited for performance evaluation in free-living conditions. Nonetheless, seeing the lack of falls in this dataset, FallAllD will be used as a complementary dataset to train the machine learning models and to evaluate their sensitivity. The ADLs as well as the falls in the RealAct dataset are analyzed in the next section.

III. An Efficient Preprocessing System for Fall Detectors

In this section, the RealAct dataset described in Section II is analyzed where the objective is to propose a pre-processing system for fall detectors with minimal assumptions about the mechanism of falling. This pre-processing system is aimed to achieve 100% of sensitivity with an initial reasonable specificity. The idea is inspired by the generic definition of a fall given by WHO [1]: "*A fall is defined as an event which results in a person coming to rest inadvertently on the ground or floor or other lower level*". Following this definition, we define three conditions $\{C_1, C_2, C_3\}$ that are described in Table I. The condition C_1 reflects the inadvertent movement of a person coming to rest which could be characterized by the presence of an impact shock in the acceleration signal. Particularly, the sum vector magnitude of acceleration $||\mathbf{a}|| = \sqrt{a_x^2 + a_y^2 + a_z^2}$ is compared with a predefined threshold 2.25 g. It is worth mentioning that this threshold is quite low since human could generate such level of acceleration with soft and safe movements like e.g. laying down on a bed. The condition C_3 inspects whether the movement happened in the downward direction as mentioned in the previous definition. Particularly, ΔH is compared with a predefined threshold to verify whether the subject moved down for more than 25 cm. While falling from an altitude of only 25 cm seems unlikely, we set this low threshold for ensuring the sensitivity of the fall detection system. Intuitively, after falling down, old subjects would not be able to recover rapidly. The condition C_2 monitors whether the subject remained inactive for more than 3.2 seconds ($T_{inac} \geq 3.2$ s). This period is long enough to express inactivity while it is not too long to affect the sensitivity. It is worth mentioning that, for industrial constraints, we might adjust these parameters (particularly raise their values) to increase the specificity of the system. Nevertheless, the number of false negatives (*i.e.* the number of missed falls) might also increase. We assume that the three conditions are met in all falls. Particularly, we expect first the presence of an impact shock (C_1 is met) followed by an inactivity period (C_2 is met) and the difference in altitude ΔH between the inactivity period and the pre-fall phase is greater than the predefined threshold (C_3 is met). To understand the impact of the aforementioned conditions on the specificity of a fall detector in a meaningful way, the following combinations of conditions are analyzed:

- the presence of an impact shock: C_1
- the presence of an impact shock followed with an inactivity period, denoted as $C_1\&C_2$
- the presence of an impact shock followed with an inactivity period with a downward movement for more than 25 cm, denoted as $C_1\&C_2\&C_3$.

To justify the election of the aforementioned three thresholds, our team compared the proposed method to a reference fall detector on the market. The latter device detects heavy falls (high impact shock) and is based on two conditions, namely (i) loss of altitude (a downward fall of 60 cm) and (ii) an inactivity period of atleast 15 seconds. As a result, the sensitivity of this device is very low compared to our proposed

TABLE I
THE CONSIDERED CONDITIONS IN THE PREPROCESSING SYSTEM

Condition	formula	description
C_1	$\|\|\mathbf{a}\|\| \geq 2.25$ g	This condition represents the presence of an impact shock
C_2	$T_{inac} \geq 3200$ ms	The user remained inactive for 3200 ms at least
C_3	$\Delta H \leq -25cm$	The user moved downwards 25 cm at least

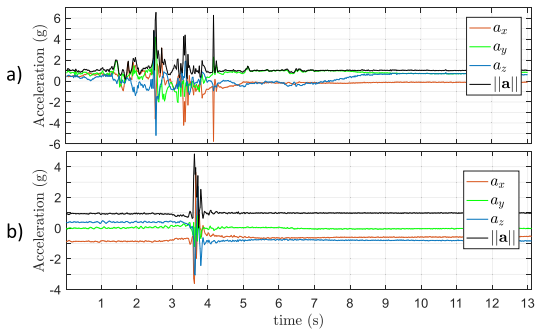

Fig. 2. 3D acceleration signals of two real falls of older adults: a) a fall captured using a wrist-worn device. b) a fall captured using a neck-worn device.

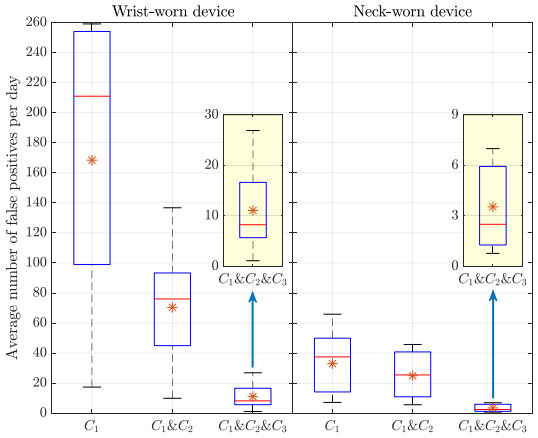

Fig. 3. The average number of false positives per day for wrist- and neck-worn devices represented in box plots. Results are first averaged per subject and then averaged per day.

method. For instance, falls from bed are hardly detectable by this reference fall detector.

Now, activity signals are scanned using overlapped sliding windows. The length of a sliding window is set to $T_w = 13$ s since this length is sufficient to capture all the phases of falling i.e. 3 s for the pre-fall phase, 1 s for the critical phase and 9 s for the post-fall phase. More detailed description of fall phases is provided in Section IV-B. Since a small decision-making period, T_d, increases the sensitivity of the fall detection system, we set $T_d = 200$ ms. Thus, the number of decisions made per day is 432 000.

Fig. 2-a illustrates the acceleration signal of a real fall captured by a wrist-worn device. The subject was walking using a walker and he/she stumbled and fell down on the ground. As shown in Fig. 2-a, the impact shock of acceleration is $||\mathbf{a}|| = 6.54$ g. The change in altitude between the inactivity period and the pre-fall phase is around 60 cm downwards. Fig. 2-b illustrates the acceleration signal of a real fall captured by a neck-worn device. The subject was sitting on the w.c. seat then he/she fell down ending half-sitting on the ground. The acceleration impact shock is 4.83 g while the change in the altitude is only 29 cm downwards.

Now, we discuss the expected number of false positives, in terms of the above-mentioned conditions. Particularly, we are interested in analyzing the average number of FPs per day denoted as $\tilde{FP}/day$. Fig. 3 shows $\tilde{FP}/day$ for wrist- and neck-worn devices where the results are first averaged per subject and then averaged per day. From this figure, we first

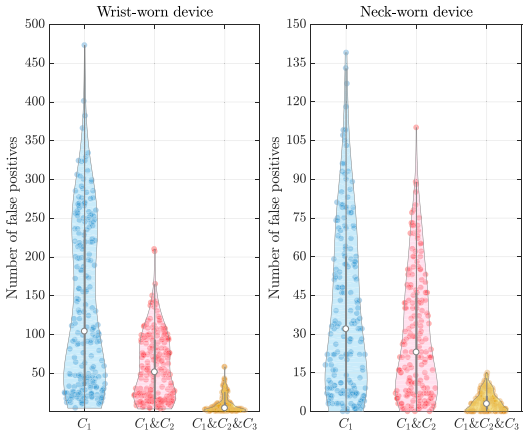

Fig. 4. The average number of false positives per day for wrist- and neck-worn devices represented in violin plots. Each point represents one day and the results are not averaged per subject.

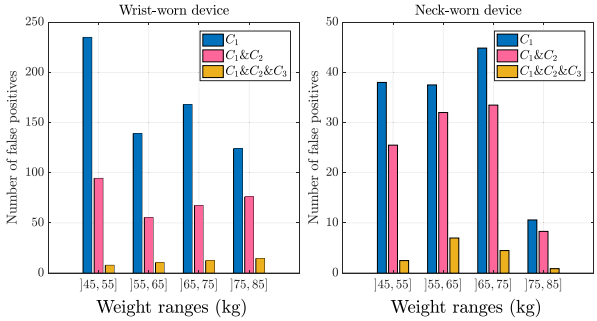

Fig. 5. The number of positives as a function of the subjects' weight.

note that $\tilde{FP}/day$ is considerably higher for wrist-worn devices than for neck-worn ones. It is an expected result since the arm can move in higher degrees of freedom than the chest leading to higher probabilities of meeting the three defined conditions with normal ADLs. The condition C_1 is met on average 168 times per day with wrist-worn devices and 33 times per day with neck-worn devices (averages are represented by red stars in Fig. 3). This shows that the predefined threshold (2.25 g) could be easily reached with normal ADLs of older adults and is highly unlikely to affect the sensitivity. When considering $C_1\&C_2$, we note that $\tilde{FP}/day$ decreases to 70 and 25 using wrist- and neck-worn devices, respectively. This emphasizes the importance of inspecting the presence of a short inactivity period after the impact shock. When combining the three conditions, $\tilde{FP}/day$ considerably decreases to 11 and 3.5 for wrist- and neck-worn devices, respectively. Clearly, employing the barometer to check the difference in altitude is helpful to reduce the number of FPs while it is not expected to affect the sensitivity since falling for less than 25 cm is highly unlikely. Fig. 4 shows the complete distribution of the numbers of FPs per day where each point in this figure represents one day. Since the results in this figure are not averaged over subjects, they are more useful when analysing the worst case of the proposed preprocessing system i.e. the maximum number of FPs per day. When considering the three conditions, the maximum number of FPs is 58 and 15 for wrist- and neck-worn devices, respectively.

Fig. 5 illustrates the effect of the subject weight on $\tilde{FP}/day$. We note that there is no monotonic relationship

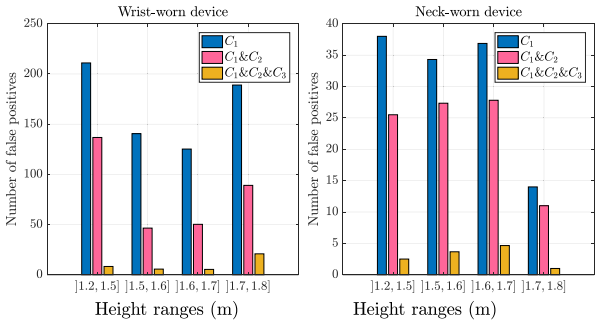

Fig. 6. The number of positives as a function of the subjects' height.

between the weight and $\tilde{FP}/day$ neither for wrist- nor for neck-worn devices. One clear observation with neck-worn devices is that the subjects whose weight is over 75 kg generate the minimal number of FPs in terms of the three considered combinations of conditions. Fig. 6 shows the effect of the subject height on $\tilde{FP}/day$. There is also no monotonic relationship between the height and $\tilde{FP}/day$. The only obvious point is that the users of neck-worn devices whose height is greater than 170 cm show the minimal number of FPs in terms of the three considered combinations of conditions. We conclude from Fig. 6 and Fig. 5 that $\tilde{FP}/day$ is almost independent from the weight and height while it is clearly subject dependent.

The combination of conditions $C_1\&C_2\&C_3$ can be used as an efficient preprocessing system for fall detectors for the following reasons:

- it reduces the number of windows that should be further inspected for falls from tens of thousands per day (e.g. 432 000/day in our configuration) to an average of 11/day for wrist-worn devices and 3.5/day for neck-worn devices;
- the considerable reduction in the number of suspected windows is achieved at a low computational cost and it is not gained at the expense of sensitivity;
- it makes minimal assumptions about the mechanism of falling;
- it is invariant to the position of the fall detector (wrist/neck), invariant to the orientation of the fall detector and almost invariant to the weight and the height of the user.

The proposed preprocessing system is followed by an ML-based fall detection algorithm which is explored in the next section.

IV. The Proposed Fall Detection Algorithms

In this section, we propose several machine learning algorithms for fall detection. We first describe the datasets used for training and evaluating the performance of the proposed algorithms. Then, we explain how an activity window is dynamically segmented into several time phases from which the features are extracted. Feature engineering is then explored where the objective is to derive a set of rotation-invariant features that could be used for representing acceleration signals in both wrist- and neck-worn fall detectors. The extracted features are then used by a multitude of machine learning-based classifiers that are introduced at the end of this section.

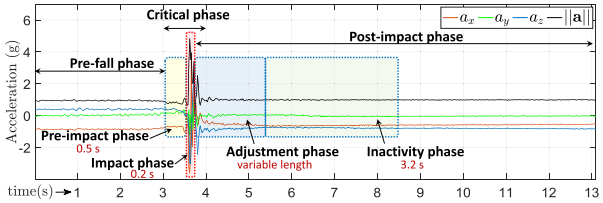

Fig. 7. Dynamic segmentation of a sliding window into different phases.

A. Datasets

In order to train the proposed models, two data sources are employed: 1) the RealAct dataset that has been introduced in Section II, and 2) the public dataset, FallAllD [7], that has been recently published by our team as an open access dataset. Particularly, RealAct is used as a source of real world ADLs (+ two real falls) and FallAllD is used as a source of simulated falls.

While both acceleration and barometric pressure signals are used in the preprocessing system, only acceleration signals are considered in the machine learning models. The preprocessing system keeps only the acceleration signals that are suspected to represent falls and introduces them to the machine learning model. To this end, we extract from the RealAct dataset only the acceleration signals where the conditions C_1, C_2 and C_3 are satisfied. Clearly, only such kind of signals will arrive to the machine learning model and it is not appropriate to train the model on conditions that could never be seen. In other words, we train the proposed models to discriminate between falls and the ambiguous ADLs. The total number of the considered data samples is 3665. These data are divided into a training set of 1259 samples and a test set of 2406 samples. Particularly, a balanced training set has been built such that the following samples are used for training: 1) ambiguous ADLs from 97 days (621 samples) extracted from the RealAct dataset, and 2) around 75% of falls from the FallAllD dataset (638 samples). On the other hand, the test set contains: ambiguous ADLs from 303 days (2170 samples) of the RealAct dataset, 2 real falls from the same dataset and 25% of falls from the FallAllD dataset. Training data are extracted from a set of 21 subjects while test data are extracted from a set of 17 different subjects i.e. to ensure a fair evaluation of the generalization capability of the trained models, the test set does not contain data from any subject who contributed to the training set.

B. Dynamic Segmentation of an Activity Window

As we introduced in Section III, activity signals are scanned using overlapped sliding windows whose length is $T_s = 13$ s. The period between two consecutive overlapped windows is $T_s = 200$ ms. Some methods in the literature [5] extracted global features from the whole window. Other methods [8], [9] extracted local features from fixed positions in the window. Contrary to both types of methods, our strategy is to dynamically segment the sliding window in order to extract features from particular phases of an activity that is suspected to be a fall. Fig. 7 illustrates the proposed dynamic segmentation strategy. We start by looking for an impact peak in the $||\mathbf{a}||$ signal between the 3^{rd} and 4^{th} seconds of the sliding window, period denoted as "critical phase". If an impact peak is found,

a sub-window of a duration of 0.2 s centered around the peak is segmented, period denoted as "impact phase". The 0.5 s-length period that precedes the impact phase is defined as "pre-impact phase" while the period that precedes the pre-impact phase is defined as "pre-fall phase". The period of the pre-fall phase is between 2.4 and 3.4 s according to the position of the impact peak in the critical phase. The period after the critical phase is then scanned in order to find any inactivity period that lasts more than $T_{inac} = 3.2$ s. Particularly, this period is divided into sub-windows of 200 ms. The variance of these 200-ms acceleration signals are calculated. If 16 contiguous sub-windows possess a relatively low variance (below 0.005625), then this period of 3.2 s is segmented and defined as "inactivity phase". It is worth mentioning that this threshold is somewhat tolerant for detecting inactivity, which means that even if the subject is slightly moving his hand, the inactivity phase would still be detected. The period between the impact and inactivity phases is defined as the "adjustment phase". Its length varies from 0 to 6.7 s depending on the positions of the impact and inactivity phases. For example, suppose that the impact phase lies between 3.6 s and 3.8 s, while the inactivity phase lies between 5.3 s and 8.5 s, then the adjustment phase is the period between 3.8 s and 5.3 s (with a length of 1.5 s). Now, when it comes to falls including a series of successive peaks, the same logic is followed. In this case, the impact phase will be centered around the highest peak, the pre-fall and the pre-impact phases will include the peaks which precede the highest peak, and the adjustment phase, which is of variable length, will include the remaining peaks which follow the highest one. Consequently, the inactivity period of 3.2 s, if found, would occur after these consecutive peaks. Features are extracted from the dynamically segmented window as explained in the next section.

C. Feature Engineering

In this section, three rotation-invariant features are proposed to be used in wrist- and neck-worn fall detection systems.

Intuitively, falls are expected to show higher level of randomness than ADLs. Since entropy E is usually used as an expression of randomness, it could be used as a helpful feature when calculated in the impact and adjustment phases. To this end, the first feature F_1, is the sum of distribution entropy E_x, E_y and E_z of the three acceleration components a_x, a_y and a_z respectively, over the impact and adjustment phases.

In addition, falls usually result in a change of the body orientation. Such changes of orientation could be captured by comparing the mean values of the acceleration signals in the pre-fall and inactivity phases, denoted respectively as $\tilde{a}_\alpha^{pr}$ and $\tilde{a}_\alpha^{inac}$ $\forall \alpha \in \{x, y, z\}$. This constitutes the second feature F_2.

The third and last feature is based on the weightlessness: the fact that the sum vector magnitude of acceleration $||\mathbf{a}||$ goes to zero when the body experiences a free fall. The weightlessness feature has been used in several fall detection systems in the literature like [18]. However, weightlessness is unlikely to happen in complex falls in which the body shocks some object before reaching the ground. To this end, it is better to monitor the sudden jumps in the acceleration signal in the impact and adjustment phases. The idea is to count the number of times where the difference between two consecutive acceleration samples exceeds a predefined threshold, $\epsilon = 0.7$ g. The total count is then normalized, i.e. it is divided by the number of samples in the impact and adjustment phases. In fact, this feature is inspired by a widely used measure of heart rate variability: the pNNx that is described in detail in [19]. The pNN0.7 is applied to a_x, a_y and a_z. Expressly, these three features are calculated using the following formulas:

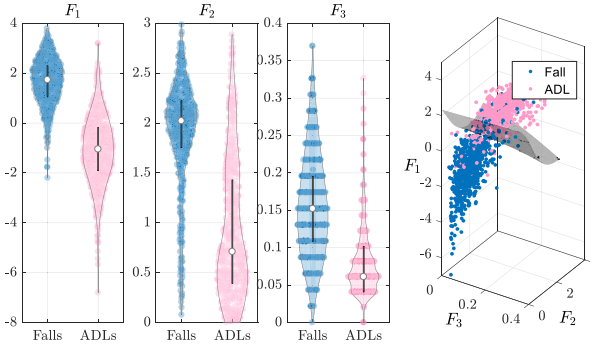

Fig. 8. The three considered features in the fall detection system extracted from falls and ADLs. Left: the individual features represented in violin plots. Most right: the three features are represented together in a 3D space with a separating hyperplane generated using an SVM classifier with a 3^{rd} order-polynomial kernel.

$$\begin{cases} F_1 = E_x + E_y + E_z \\ F_2 = \sum_{\alpha \in \{x,y,z\}} |\tilde{a}_\alpha^{inac} - \tilde{a}_\alpha^{pr}| \\ F_3 = \max_{\alpha \in \{x,y,z\}} \{\text{pNN0.7}(a_\alpha)\} \end{cases} \quad (2)$$

Fig. 8 illustrates, in violin plots, the distribution of the aforementioned features over falls and ADLs. Clearly, none of these individual features is sufficient to discriminate between falls from ADLs. However, when they are considered together as an input to a machine learning-based classifier, it is expected to achieve much better classification results. To illustrate this, the most right sub-figure in Fig. 8 shows the 3 considered features in a 3-D space with a separating hyperplane generated using a support vector machine (SVM) classifier with a 3^{rd} order-polynomial kernel. Visually, this classifier shows good classification performance. In the next section, a multitude of machine learning-based classifiers are introduced to be applied to the proposed features.

D. Machine Learning-Based Classifiers

Falls and ADLs, represented by the proposed features, are to be classified using nine machine learning-based classifiers. We use the *scikit-learn* Python library in which all the considered classifiers are implemented. To avoid listing all the tested configurations, only the configurations that have shown an interesting change in the results are reported. Unless explicitly mentioned otherwise, all the configurations are set to the defaults of the scikit-learn.

The first family of the considered classifiers covers three ensemble methods: gradient boosting (GB) [20], [21], random forest (RF) [21], [22] and AdaBoost [23] classifiers. The GB classifier is configured such that the learning rate is 0.4, the maximum tree depth is 2, and the number of estimators is

TABLE II
EXPERIMENTAL RESULTS

Device →	Neck-worn				Wrist-worn			
Criteria → Methods ↓	Sensitivity (%)	Specificity. (%)	Accuracy (%)	Number of FPs Per day	Sensitivity (%)	Specificity (%)	Accuracy (%)	Number of FPs Per day
$GB_{\{n_{est}=100\}}$	**100.00**	**98.52**	**98.95**	**0.04**	**100.00**	**96.73**	**96.89**	**0.32**
$GB_{\{n_{est}=50\}}$	97.84	96.44	96.85	0.10	97.94	93.73	93.94	0.62
$RF_{\{md=10\}}$	94.24	93.77	93.91	0.18	94.85	87.73	88.08	1.22
$RF_{\{md=2\}}$	92.09	88.43	89.50	0.33	94.85	84.12	84.66	1.57
$AdaBoost_{\{n_{est}=100\}}$	97.12	93.18	94.33	0.19	96.91	88.87	89.27	1.10
$AdaBoost_{\{n_{est}=50\}}$	93.53	91.39	92.02	0.25	97.94	86.42	86.99	1.35
SVM_{Linear}	91.37	91.10	91.18	0.25	96.91	83.25	83.94	1.66
$SVM_{Quadratic}$	90.65	94.07	93.07	0.17	93.81	86.91	87.25	1.30
SVM_{Poly-3}	88.49	94.66	92.86	0.15	92.78	90.02	90.16	0.99
SVM_{RBF}	91.37	91.69	91.60	0.24	95.88	83.69	84.30	1.62
$MLP_{(8,2)}$	91.37	93.18	92.65	0.19	93.81	85.00	85.44	1.49
$MLP_{(20,7,2)}$	92.81	91.69	92.02	0.24	95.88	82.38	83.06	1.75
$DT_{\{md=5\}}$	90.65	91.99	91.60	0.23	93.81	82.65	83.21	1.72
$DT_{\{md=10\}}$	92.09	92.58	92.44	0.21	92.78	84.89	85.28	1.50
$KNN_{\{k=3\}}$	92.81	89.02	90.13	0.31	95.88	81.56	82.28	1.83
$KNN_{\{k=5\}}$	92.81	91.39	91.81	0.25	94.85	83.09	83.68	1.68
LDA	93.53	90.50	91.39	0.27	96.91	80.80	81.61	1.90
QDA	92.81	90.80	91.39	0.26	96.91	80.36	81.19	1.95
NB	95.88	83.20	83.83	1.66	93.64	84.19	85.12	1.13

$n_{est} \in \{50, 100\}$. The number of estimators in the RF classifier is set to 20 while the maximum depth of the trees is set to two different values $md \in \{10, 2\}$. The number of estimators in the AdaBoost classifier is set to $n_{est} \in \{50, 100\}$. The second family covers SVM classifiers where a linear, quadratic, 3^{rd} order-polynomial and radial-basis-function (RBF) kernels are considered. The third family covers multi-layer perceptron (MLP) neural networks. Two MLPs are designed where the number of hidden layers is 2 and 3, respectively, and the number of neurons in the hidden layers is $\{8, 2\}$ and $\{20, 7, 2\}$, respectively. The last family covers diverse basic classifiers that are: decision tree (DT), k-nearest neighbors (KNN), linear and quadratic discriminant analysis (LDA and QDA) and naive Bayes (NB) classifiers. The maximum depth of the tree in the DT classifier is configured to two different values $md \in \{5, 10\}$. Finally, the number of nearest neighbors in the KNN classifier is set to $k \in \{3, 5\}$. The performance of the aforementioned classifiers is evaluated in the next section.

V. EXPERIMENTAL RESULTS

In this section, the performance of the proposed machine learning-based solutions is evaluated in terms of 4 performance criteria: sensitivity, specificity, accuracy and the average number of false positives per day. Before exploring the detailed results, it is worth mentioning that all classifiers (with all the considered configurations) succeeded to detect the two real falls. Table II shows the experimental results for both wrist- and neck-worn fall detection systems. Despite that the proposed solution is independent from the position of the fall detector (wrist/neck), i.e. the models have been trained using a mixture of activities collected by wrist- and neck-worn devices, we show the results on each position separately to clarify the performance differences. From Table II, we observe that the specificity of all classifiers, except the NB classifier, is higher for the neck-worn devices than the wrist-worn devices. As introduced in Section III, this result is expected since more ambiguous ADLs could be generated by the hand than the chest recalling that the hand can move with more degrees of freedom than the chest. From Table II, we clearly note that the best classifier is the gradient boosting machine that is configured to $n_{est} = 100$ estimators. It shows a perfect sensitivity for both wrist- and neck-worn fall detectors. With $GB_{n_{est}=100}$, neck-worn devices generate an average of one false positive every 25 days while the average number of false positives generated by wrist-worn devices is about one every 3 days. From acceptability point of view, some users could prefer wrist-worn devices. However, if the user has no preferences, neck-worn devices are clearly to be recommended since they generally show better accuracy.

To experimentally show the effect of using simulated ADLs on the system sensitivity, the same best model, $GB_{\{n_{est}=100\}}$, has been trained on the same falls but on simulated ADLs from the FallAllD dataset instead of the RealAct dataset. The sensitivity decreased to 95.43% and 80.12% for neck- and wrist-worn devices, respectively. This emphasizes the importance of the proposed strategy of considering real-life ADLs.

The results also confirm our claims concerning the interpretability of the specificity results. We note that the classifier $GB_{\{n_{est}=100\}}$ shows a specificity of 98.52% and 96.73% on the neck- and wrist-worn devices, respectively. If we have no access to the results of the 4^{th} criterion, the average number of FPs per day, it will be quite difficult to understand the impact of the small difference (1.79%) in the specificity values of the neck- and wrist-worn devices. Thanks to the proposed strategy, we have access to this important performance criterion and thus we are able to know that, despite the small difference in specificity, wrist-worn fall detectors generate 8 times as many false positives as neck-worn devices.

VI. PRACTICAL CONSIDERATIONS

This section provides some perspectives for applying the proposed fall detection system under the hardware limitations

of wearable devices. We first recall that the proposed design consists of a threshold-based preprocessing step followed by a ML algorithm. The computational cost of the preprocessing system is low since it requires only simple comparison operations. As shown in Section III, the preprocessing system filters out the majority of the scanned activities and keeps only few number of suspected windows for further ML-based processing. Particularly, the average number of windows that requires ML-based processing is 11/day for wrist-worn devices and 3.5/day for neck-worn devices. Given this limited number of suspected windows, two possible solutions could be used to implement the ML-based system: 1) sending the raw acceleration data to a remote computer which is equipped with the ML algorithm. This is feasible since the average amount of data to be sent is only 42 K Bytes/day (11 windows$\times$13 s$\times$50 Hz$\times$3 axes$\times$2 Bytes) for wrist-worn devices and 13 K Bytes/day for neck-worn devices. We tested this kind of solution in our previous work [10] in which a remote computer receives the raw acceleration data and executes a complex fall detection algorithm; 2) embedding the algorithm in a wearable fall detector that is not necessarily restricted to give a decision in real-time but in near-real-time instead. For instance, the algorithm could give a decision delayed for few seconds. This small delay allows for employing more complicated algorithms. Selecting between the above two solutions depends on the computational resources of the wearable device and the complexity of the employed ML algorithm.

VII. Conclusion

The majority of the state-of-the-art ML-based fall detection systems are trained on simulated datasets. Typically, the distribution of the ADL types in these datasets is considerably different from the actual distribution in real life. In this work, we have shown that this phenomenon causes two major problems. First, it leads to improving the specificity at the expense of sensitivity. Second, it makes the specificity measure uninterpretable. These problems have been tackled in this work where a reliable design of a machine learning-based fall detection system has been proposed. It is based on analyzing the activities of older adults experiencing their daily life. The proposed system can be used either with neck- or wrist worn fall detectors and it is rotation invariant. The system has shown a sensitivity of 100% on real falls as well as on simulated falls. When evaluated on 303 days of recorded activities of older adults, it has shown a reasonable specificity. Particularly, the system generates only one false positive, on average, every 25 days for the neck-worn device and an average of one false positive every 3 days for the wrist-won device. In future works, the computational complexity and the embeddability issues of the proposed solution will be conducted.

Acknowledgment

The Authors would like to thank Pierre Cherel and Michel Le Mer, with RF-TRACK society, for their valuable contributions to this work.

References

[1] World Health Organization. (Jan. 2018). *Falls*. Accessed: Sep. 21, 2020. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/falls

[2] W. Saadeh, S. A. Butt, and M. A. B. Altaf, "A patient-specific single sensor IoT-based wearable fall prediction and detection system," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 5, pp. 995–1003, May 2019.

[3] S. A. Shah *et al.*, "Sensor fusion for identification of freezing of gait episodes using Wi-Fi and radar imaging," *IEEE Sensors J.*, vol. 20, no. 23, pp. 14410–14422, Dec. 2020.

[4] M. Musci, D. De Martini, N. Blago, T. Facchinetti, and M. Piastra, "Online fall detection using recurrent neural networks on smart wearable devices," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 3, pp. 1276–1289, Jul. 2021, doi: 10.1109/TETC.2020.3027454.

[5] J. Silva, D. Gomes, I. Sousa, and J. S. Cardoso, "Automated development of custom fall detectors: Position, model and rate impact in performance," *IEEE Sensors J.*, vol. 20, no. 10, pp. 5465–5472, May 2020.

[6] F. Bianchi, S. J. Redmond, M. R. Narayanan, S. Cerutti, and N. H. Lovell, "Barometric pressure and triaxial accelerometry-based falls event detection," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 18, no. 6, pp. 619–627, Dec. 2010.

[7] M. Saleh, M. Abbas, and R. Le Bouquin Jeannès, "FallAllD: An open dataset of human falls and activities of daily living for classical and deep learning applications," *IEEE Sensors J.*, vol. 21, no. 2, pp. 1849–1858, Jan. 2021.

[8] M. Saleh and R. Le Bouquin Jeannès, "Elderly fall detection using wearable sensors: A low cost highly accurate algorithm," *IEEE Sensors J.*, vol. 19, no. 8, pp. 3156–3164, Apr. 2019.

[9] M. Saleh and R. Le Bouquin Jeannès, "An efficient machine learning-based fall detection algorithm using local binary features," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Rome, Italy, Sep. 2018, pp. 3–7.

[10] M. Saleh, N. Georgi, M. Abbas, and R. Le Bouquin Jeannès, "A highly reliable wrist-worn acceleration-based fall detector," in *Proc. 27th Eur. Signal Process. Conf. (EUSIPCO)*, A Coruna, Spain, Sep. 2019, pp. 2–6.

[11] N. Noury, P. Rumeau, A. Bourke, G. ÓLaighin, and J. Lundy, "A proposal for the classification and evaluation of fall detectors," *IRBM*, vol. 29, no. 6, pp. 340–349, Dec. 2008.

[12] A. Sucerquia, J. López, and J. Vargas-Bonilla, "SisFall: A fall and movement dataset," *Sensors*, vol. 17, no. 12, p. 198, Jan. 2017.

[13] E. Casilari, J. A. Santoyo-Ramón, and J. M. Cano-García, "Analysis of a smartphone-based architecture with multiple mobility sensors for fall detection," *PLoS ONE*, vol. 11, no. 12, Dec. 2016, Art. no. e0168069.

[14] D. Micucci, M. Mobilio, and P. Napoletano, "UniMiB SHAR: A dataset for human activity recognition using acceleration data from smartphones," *Appl. Sci.*, vol. 7, no. 10, p. 1101, 2017.

[15] A. Wertner, P. Czech, and V. Pammer-Schindler, "An open labelled dataset for mobile phone sensing based fall detection," in *Proc. 12th EAI Int. Conf. Mobile Ubiquitous Syst., Comput., Netw. Services*, Coimbra, Portugal, 2015, pp. 22–24.

[16] J. Klenk *et al.*, "The FARSEEING real-world fall repository: A large-scale collaborative database to collect and share sensor signals from real-world falls," *Eur. Rev. Aging Phys. Activity*, vol. 13, no. 1, pp. 1–7, Dec. 2016.

[17] P. Pierleoni *et al.*, "A wearable fall detector for elderly people based on AHRS and barometric sensor," *IEEE Sensors J.*, vol. 16, no. 17, pp. 6733–6744, Sep. 2016.

[18] Y.-S. Su and S.-H. Twu, "A real time fall detection system using triaxial accelerometer and Clinometer based on smart phones," in *Proc. Int. Conf. Biomed. Health Informat. (ICBHI)*, 2019, pp. 129–137.

[19] J. E. Mietus, "The pNNx files: Re-examining a widely used heart rate variability measure," *Heart*, vol. 88, no. 4, pp. 378–380, Oct. 2002.

[20] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[21] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. New York, NY, USA: Springer-Verlag, 2009.

[22] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[23] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1995.