Automated Detection of Real-World Falls: Modeled From People With Multiple Sclerosis

Clara Mosquera-Lopez, *Member, IEEE*, Eric Wan, Mahesh Shastry, *Member, IEEE*, Jonathon Folsom, Joseph Leitschuh, John Condon, Uma Rajhbeharrysingh, Andrea Hildebrand, Michelle Cameron, and Peter G. Jacobs

***Abstract*—Falls are a major health problem with one in three people over the age of 65 falling each year, oftentimes causing hip fractures, disability, reduced mobility, hospitalization and death. A major limitation in fall detection algorithm development is an absence of real-world falls data. Fall detection algorithms are typically trained on simulated fall data that contain a well-balanced number of examples of falls and activities of daily living. However, real-world falls occur infrequently, making them difficult to capture and causing severe data imbalance. People with multiple sclerosis (MS) fall frequently, and their risk of falling increases with disease progression. Because of their high fall incidence, people with MS provide an ideal model for studying falls. This paper describes the development of a context-aware fall detection system based on inertial sensors and time of flight sensors that is robust to imbalance, which is trained and evaluated on real-world falls in people with MS. The algorithm uses an auto-encoder that detects fall candidates using reconstruction error of accelerometer signals followed by a hyper-ensemble of balanced random forests trained using both acceleration and movement features. On a clinical dataset obtained from 25 people with MS monitored over eight weeks during free-living conditions, 54 falls were observed and our system achieved a sensitivity of 92.14%, and false-positive rate of 0.65 false alarms per day.**

***Index Terms*—Automated fall detection, real-world falls, multiple sclerosis, imbalance-aware classification, auto-encoder, random forest hyper-ensemble.**

Manuscript received March 21, 2020; revised October 13, 2020; accepted November 17, 2020. Date of publication November 27, 2020; date of current version June 4, 2021. This work was supported by the Department of Veterans Affairs, Rehabilitation Research and Development Service (Award RX001831-01A1). Peter G. Jacobs and Eric Wan have a financial interest in MotioSens, a company that may have a commercial interest in the results of this research and technology.
(Corresponding author: Clara Mosquera-Lopez.)

Clara Mosquera-Lopez, Mahesh Shastry, Jonathon Folsom, Joseph Leitschuh, Uma Rajhbeharrysingh, and Peter G. Jacobs are with the Department of Biomedical Engineering, Oregon Health & Science University, Portland 97239, OR USA (e-mail: mosquera@ohsu.edu; mahesh.shastry@gmail.com; folsomj@ohsu.edu; leitschj@ohsu.edu; rsingh.uma@gmail.com; jacobsp@ohsu.edu).

Eric Wan is with the Department of Electrical and Computer Engineering, Portland State University, Portland, OR USA (e-mail: eric.wan@pdx.edu).

John Condon is with the Proto-tech Research, Portland, OR USA (e-mail: jcondon11@yahoo.com).

Andrea Hildebrand and Michelle Cameron are with the Department of Neurology at the Oregon Health & Science University and the Veterans Affairs Portland Health Care System, Portland, OR USA (e-mail: hildeand@ohsu.edu; cameromi@ohsu.edu).
Digital Object Identifier 10.1109/JBHI.2020.3041035

I. Introduction

FALLING is a major public health problem with the cost of falls reaching \$60B per year [1] due to their impact on hip fractures, disability, reduced activity and mobility, hospitalization and death. Seniors are particularly at risk as approximately one in three people over the age of 65 will fall each year [1]. "Long-lie" falls, a particularly dangerous event, when a person remains on the ground for a long period of time after a fall [2], or even just the fear of falling can impact various negative health outcomes including lower mobility, social isolation, nursing home admission, loss of independence, and lower quality of life [2], [3]. The ability to detect a fall in real-time is therefore of critical importance.

With the availability and growing adoption of smart and connected sensing devices, automatic fall detection systems are being rapidly developed for research and commercial purposes. These use a variety of sensing technologies including inertial sensors (e.g., accelerometer and gyroscope) [4]–[12], video cameras [4], [13], depth cameras (especially the Kinect) [5], [7], [14]–[19], doppler radar [20], [21], and radio-frequency sensors (e.g., Wi-Fi) [22]. Commercial fall detection systems are typically based on inertial sensors worn by the person being monitored. Large changes in acceleration are used to automatically detect a fall and notify an emergency response system or care provider [23]. While these commercial systems can provide an improved sense of security, and potentially reduce the fear of falling [24], users of these systems report high false positive rates, and published accuracy measures have all been reported on simulated falls that have been performed by actors in a laboratory [25]–[27].

The development of a highly accurate real-world fall detection system has been elusive for a number of reasons. First, it is difficult to distinguish a fall (defined as an event which results in a person coming to rest inadvertently on the ground, floor or other lower level [28]) from other confounding activities of daily living (ADL) such as lying down, sitting on a chair, or reaching for the floor. For systems optimized for high sensitivity, fall confounders can be a source of multiple false positive alarms that would limit a system's value and usability in practical applications. Second, given the relatively infrequent, accidental, and dangerous nature of actual falls, it is a daunting task for researchers to build datasets with real-world fall data [29] for algorithm development and validation. A day during which a fall occurs happens so infrequently relative to no-fall days that it is very difficult to collect an adequate dataset of falls. To collect just 100 real-world falls in seniors, under the assumption that, on average, one in three seniors have a fall each year, would require monitoring 109,500 days of free-living activities. Therefore, automated fall detectors have only been tested on simulated falls, and generally in healthy adults. Finally, computational approaches for analysis and modeling of large amounts of data, particularly classical machine learning algorithms such as support vector machines (SVM) or artificial neural networks (ANN), tend to generalize poorly when trained on highly unbalanced datasets because they usually predict the minority class (i.e., the class that contains a much smaller number of instances than the other classes), which is often the class of interest, with very low precision [30].

In contrast to seniors, people with multiple sclerosis (MS) tend to fall often. Nearly 50% of people with MS fall in a six-month period and around 30 to 50% fall multiple times [31]–[33]. While falls in people with MS may be different from falls in seniors or healthy adults [34], the MS population provides a potentially important model for developing real-world fall detection algorithms. Approximately one million people live with MS in the United States [35]. MS is a chronic progressive disease that damages the brain, spinal cord and optic nerves impacting a wide range of neurological functions, including sensation, strength, vision, balance and coordination, all of which predispose people with MS to falls [36]–[38]. As is true with seniors, falls negatively impact the health and quality of life of people with MS, causing physical injuries and psychological impacts including fear of falling, which can then result in reduced activities and social isolation [39].

The high incidence of falls in people with MS makes it an ideal model for studying falls and fall prediction algorithms. In MS studies involving fall intervention and rehabilitation, the most commonly employed method for prospective monitoring of falls is to use paper fall calendars [32], [40]. Patients are asked to record the number of times they fall each day and return the calendars every two to four weeks. These calendars might not be completed at the time of a fall, resulting in recall bias and incomplete information. In addition, calendars are easily lost or misplaced, may not be sent in to the investigator, and do not provide real-time notification of falls. Despite this, paper fall calendars are still being used as the "gold standard" in fall monitoring [41]. In this study, paper calendars in conjunction with other fall report methodologies were used to establish ground truth on falls. However, if a more accurate fall monitoring tool were to become available, it would be useful both for real-time monitoring (in the case of senior care) and also for use in research studies, as would be the case in drug and rehabilitation intervention studies in MS.

This paper presents the design and evaluation of a fall detection system that is trained on real-world falls in people with MS. The system consists of both a body-worn tri-axial accelerometer and a context-aware movement monitoring system that uses time-of-flight (ToF) indoor wireless beacons positioned around a home to track a person's movement. The contributions of this paper are threefold. First, our work is the first system developed to detect and record fall events in people with MS who, due to the gait and balance problems associated with the disease, have movement patterns that differ greatly from those of healthy adults. Second, this work describes several approaches to design and validate a robust classifier ensemble that appropriately handles the high imbalance of a real-world dataset collected from people with MS during an 8-week observational study. Third, in addition to the commonly used inertial features derived from accelerometry data, new measures obtained from movement tracking sensors were employed to demonstrate the impact of indirect context features on the accuracy of fall detection.

The rest of this paper is organized as follows: Section II presents a brief literature review that discusses related work on fall detection. Section III presents our modeling approach including the specifics of the data collection and processing, as well as the architecture of the fall detection system. Section IV discusses the performance of our method on simulated and clinical data, and Section V concludes the paper with recommendations for future research directions.

II. Related Work

In general, fall detection systems use high acceleration portions of the signal to identify features that are good predictors of the occurrence of fall events. The impact stage of a fall is characterized by abrupt changes in body motion parameters such as acceleration, orientation, or inclination [42]. Over the past years, many fall detection approaches have been proposed. Most previously proposed methods use a body-worn inertial measurement unit (IMU) containing a tri-axial accelerometer [6], [11], [12], [43]. Some systems include complementary sensors to improve fall detection accuracy. For example, Wang *et al.* [6] developed a two-modal approach to fall detection based on the magnitude of the acceleration during the impact stage of the fall as well as the heart rate and the orientation of the person's trunk for a few seconds after impact. Acceleration and heart rate were measured using an IMU unit and pulse pressure sensor, respectively. Three different thresholds were applied to acceleration and heart rate features (magnitude of the acceleration, change in the heart rate measured after the impact with respect to the pre-impact value possibly indicating stressful or dangerous situation, and user's trunk orientation) in chronological order to alert about the occurrence of an accidental fall. The authors reported high fall detection accuracy on simulated falls close to 97.5% with sensitivity 96.8% and specificity 98.1%. During practical tests monitoring people of ages 5 to 70 years old, the high specificity of the system was confirmed (i.e., there were no false positive alarms), but there were also no falls; therefore, the sensitivity of the system is yet to be evaluated.

Kau *et al.* [8] proposed a smart-phone-based fall detection and corresponding wide area rescue system. The authors proposed a cascaded classifier that sequentially evaluates features from a tri-axial accelerometer signal generated by the IMU embedded in the smart phone. The first feature, evaluated during the pre-impact stage of the fall, is the acceleration magnitude during the weightlessness-like period preceding the impact. The second feature is the amplitude of the acceleration magnitude during the impact stage. These first two conditions for the fall event recognition can be regarded as necessary, but not sufficient because they are also present during other activities such as running, jumping, or going down or up stairs. The next feature relates to the variability of the acceleration after impact due to the motionless status after the fall event. Next, the orientation of the device is evaluated to detect changes in the person's position. Finally, wavelets are used to extract features from the high frequency components of the acceleration and detect accidental falls using a SVM classifier. If a fall event is detected, the user's position is acquired by the global positioning system (GPS) or the assisted GPS, and sent to an specified rescue center via the cellular communication network. The authors reported a sensitivity of 92% and specificity of 99.75% on experiments conducted in a laboratory environment.

Aziz *et al.* [43] validated an SVM-based fall detection system trained on simulated falls performed by young healthy adults. The system used statistics of the acceleration magnitude to detect falls, and then tested the model on three groups of older adults who where continuously monitored during a six-hour period. The sensitivity of the algorithm was 80% (8 out of 10 real-world falls were successfully detected) with associated false positive rates ranging from 0.05 to 0.15 false alarms per hour, which is equivalent to 1.2-3.6 false alarms per 24-hour period.

Sucerquia *et al.* [12] also provided insights into the accuracy, specifically the false positive rate, of a real-time fall detection system during a real-life experiment involving three adults. They used a simple threshold on a non-linear combination of acceleration features including magnitude and variability, and reduced false positive alarms by checking the periodicity of the filtered acceleration signal during the post-impact stage of each fall event candidate. Although they achieved a fall detection accuracy of 99.3% (specificity and sensitivity greater than 99%) when performing 10-fold cross-validation on a dataset of simulated falls, the overall false positive rate was 0.1 false alarms per hour, equivalent to 2.4 false alarms per day, when tested on people in real-life environments. This false positive rate is still limiting the usability of this system.

As summarized previously, most systems trained on simulated falls produce more than one false positive each day. Furthermore, Bagala *et al.*'s [26] evaluation of 13 threshold-based fall detection methods based on accelerometer features from 29 real-world falls in older adults found considerably lower sensitivity and specificity than the values reported for the simulated falls. The average sensitivity of the evaluated methods was $57.0 \pm 27.3\%$, and there were 3 to 85 false alarms per day of monitoring.

Lipsitz *et al.* [44] evaluated an investigational fall detection device produced by Philips (Amsterdam, Netherlands) in a real-world setting. The pendant fall detection sensor worn around a person's neck was tested on older adults living in nursing homes with a documented history of at least one fall within 12 months prior to the start of the study. During a six-month trial, a total of 89 falls occurred among the older adults as reported by the nursing staff, and the device detected only 17 of them. Moreover, of the 128 instances the device labeled as falls, 111 were false positives.

More recently, Apple Inc. (Cupertino, CA) added a fall detection capability to the Apple Watch Series 4 and later, which is designed to detect hard falls and help connect users to emergency services if a detected fall is followed by a period of immobility of about one minute. To the best of our knowledge, the accuracy of this feature on controlled tests hasn't been reported in the literature.

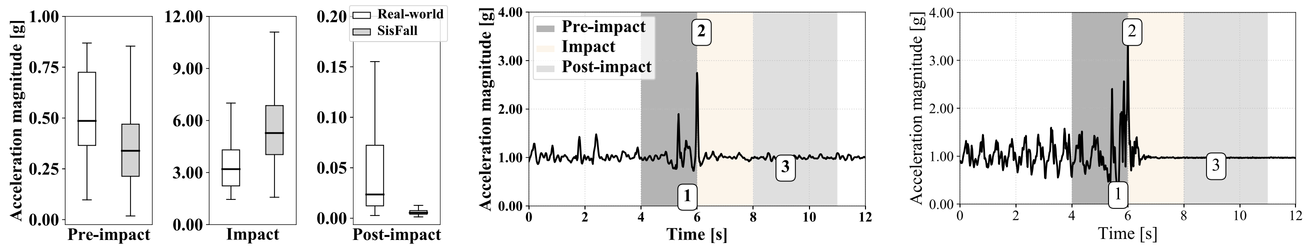
The main limitation of training a system on simulated falls by healthy adults in a laboratory setting is that their movement patterns may not accurately reflect those patterns that are typical of real-world falls in the target population. As an example, Fig. 1 shows key differences between the acceleration signals of real-world falls vs. simulated falls. Therefore, these systems trained on simulated falls only can detect falls but, to date, they have only moderate specificity and their sensitivity has not been demonstrated in real-world environments [26], [29], [44].

III. Materials and Methods

This section first describes the observational study we conducted to collect real-world fall data from people with MS and the generalities of the datasets used for model development and validation. It also describes the publicly available dataset on simulated falls that was used to augment the training of the algorithm. Then, a detailed discussion of the methods and algorithms of our automated fall detection system is presented.

A. Data Acquisition System

MotioWear (MotioSens, Portland, OR) is an advanced sensing system used for real-world movement data collection (see Fig. 2) [45]. MotioWear is designed to measure acceleration signals that can be used to detect and count falls, as well as to track a person's location, both indoors and outdoors. MotioWear uses ToF beacon technology to track a body-worn tag indoors, and GPS technology for outdoor localization. The tag also includes an IMU for capturing abrupt changes in movement that may be caused by a fall. The MotioWear system goes beyond typical IMUs by capturing both IMU data and also indoor/outdoor location of fall events which can be used to infer context. Wireless MotioWear beacons are mounted in electrical outlets throughout the home. The ToF radio transceiver in the body-worn MotioWear tag communicates with these beacons and the system localizes the tag using the ToF of the wireless data packet sent between the tag and the beacons to triangulate the tag's position using a sigma point Kalman filter [46]. MotioWear learns the specific regions of interest within a home through a simple calibration routine that is designed to take about 15-20 minutes. During this home calibration routine, the person carries the MotioWear tag in their pocket while using the MotioWear smart-phone app to identify regions of interest (e.g., bedroom, home office, bathroom, shower, toilet, etc.). After the home calibration routine, the system can infer the person's location and infer activities. The MotioWear system provides access to the raw IMU data as well as the location data, thus allowing design



(a) Distribution of acceleration magnitude (b) Real-world fall (c) Simulated fall

Fig. 1. Acceleration magnitude differences between real-world falls vs. simulated falls. (a) Distributions of acceleration magnitudes during the pre-impact stage (weightlessness-like period), impact stage, and post-impact stage. Acceleration magnitude of (b) a real-world fall of a person with MS vs. (c) a simulated fall from the *SisFall* dataset. In general, real-world falls have 1) higher acceleration magnitude during the weightlessness-like pre-impact stage, 2) lower impact acceleration, and 3) larger acceleration variability during the post-impact period.

Fig. 2. Data acquisition system MotioWear (MotioSens, Portland, OR). MotioWear tag-based indoor/outdoor tracking and context awareness hardware components (left), Waist-worn MotioWear Tag (right).

of the fall detection algorithms studied in this paper. In this study, the MotioWear system acquired inertial measurements with only the IMU tri-axial accelerometer and did not use the IMU gyroscope due to power consumption constraints.

B. Datasets

1) Real-World Fall Data: People with a confirmed diagnosis of MS of any type who reported having fallen at least twice during the two months prior to the start of the study were recruited from the Veterans Affairs Portland Health Care System (VAPORHCS) and the Oregon Health & Science University (OHSU) Multiple Sclerosis Center clinics to participate in the Free From Falls (FFF) study. FFF is a comprehensive fall prevention program that includes lectures and exercise sessions designed to enhance motor and sensory balance strategies and awareness of fall risks, and to identify and develop fall prevention strategies. Additional inclusion criteria for the study included no MS relapse in the previous month, age over 18 years or over, ability to walk at least 100 meters with or without unilateral assistance (Expanded Disability Severity Scale (EDSS) step ≤ 6.0), community dwelling, willingness and intellectual ability to understand and to sign an informed consent and to adhere to protocol requirements, and sufficient motor function to complete a written daily record of falls for eight months. Exclusion criteria included serious conditions that would preclude reliable study participation or increase risk of injury during the program per investigator's discretion and inability to follow direction in English. The FFF study was approved by the VAPORHCS and OHSU Institutional Review Boards (IRB) and registered on clinicaltrials.gov (NCT02583386).

TABLE I
BASELINE DEMOGRAPHICS, FALL HISTORY AND MOBILITY PERFORMANCE OF PARTICIPANTS, WHOSE DATA WERE USED FOR THE DEVELOPMENT AND VALIDATION OF THE FALL DETECTOR (N = 25)

Demographics		
Age, years	Range	33 - 76
	Average	54
Sex, %	Female	68
	Male	32
MS subtype, %	Relapsing-Remitting MS (RRMS)	44
	Secondary-Progressive MS (SPMS)	32
	Primary-Progressive MS (PPMS)	24
EDSS	Range	4.0 - 6.0
Fall history		
Falls in the past year, n	Range	1 - 700
	Median	6

A total of 96 participants were recruited and a subgroup of 34 of these participants were assigned to use the MotioWear sensing system for 8 weeks. These 34 participants were further split into four groups monitored in a sequential manner. Data from the first four participants were used as preliminary data to refine the fall detector system for use with the remaining participants. For instance, an important change made after the first four participants was that the accelerometer range was increased from ± 2.0 g to ± 8.0 g. Of the thirty remaining participants, one dropped out of the study, and datasets collected from four participants were excluded from the analysis due to compliance or technical problems (i.e., poor connectivity among indoor beacons making tracking data not suitable for context analysis, transmission of corrupted or noisy measurements, and connectivity problems between the hub and our data server leading to insufficient data); thus, data from 25 participants (73.5%) were included in our analysis. Table I shows the profile of the participants included in our analysis.

The participants of the study were instructed to wear the MotioWear wearable tag on their trunk or in their pockets. The MotioWear system collected movement-related data using a tri-axial accelerometer embedded in the tag that measured acceleration signals both indoors and outdoors at a sampling rate of 50 Hz. Contextual movement data was collected using the ToF sensor in the tag that measured the distance from the tag to wireless beacons positioned in outlets around the participants' homes at a sampling frequency of 4 Hz. ToF data was further processed using an Unscented Kalman filter to generate accurate estimates of participants's indoor position and velocity [46]. The data associated with this paper is available trough IEEE dataport https://ieee-dataport.org/open-access/real-world-falls-multiple-sclerosis-ms and also upon request on www.ohsu.edu/school-of-medicine/jacobs-lab/software-data.

Given the low likelihood of a fall in short periods of time, we are faced with highly imbalanced data, which means that the falls target class contains a much smaller number of instances than the ADLs class. The size of the imbalance in our real-world falls dataset can best be understood by considering that out of 25 people who were monitored for 8 weeks, 54 falls were observed, each of which lasted no longer than five seconds. This yields an imbalance of more than two million seconds of total data compared with only 270 seconds of fall data.

2) In-lab Simulated falls by Healthy Adults: While our real-world fall dataset is, to the best of our knowledge, the largest known dataset from people with MS with real-world falls, we decided to augment this dataset with a publicly available simulated falls and activities of daily living dataset called *SisFall* [11] to determine if doing so would improve the performance of the fall detection algorithm. The *SisFall* dataset includes both simulated fall events and ADLs that can confound a fall detection algorithm. *SisFall* contains inertial data recorded while healthy adult volunteers simulated falls and ADLs wearing a custom-built waist-worn wearable device composed of two types of accelerometer and one gyroscope. This dataset consists of 19 ADLs and 15 fall types performed by 23 young adults, 15 ADL types performed by 14 healthy and independent participants over 62 years old, and data from one 60-year-old participant who performed all ADLs and falls (19 ADLs and 15 fall types). Inertial measurements were recorded with a sampling frequency of 200 Hz.

C. Data Processing and Feature Extraction

The acceleration signals from our real-world dataset and the *SisFall* dataset were processed by generating 12-second signal segments centered at points with acceleration magnitude greater than 1.8 g. This threshold was selected based on the work presented in [47], where it was determined that for any position of the acceleration sensor (hip, chest, or head) without jumping and running, the maximum normal acceleration from ADLs is 1.75 g, while the minimum normal fall acceleration is 2.9 g; and the maximum tangential acceleration from ADLs is 1.6 g, while minimum tangential fall acceleration is 1.9 g. Thus, the threshold for detecting falls can be assumed to be 1.9 g or lower if the system is optimized for higher sensitivity. In this work, segments whose maximum acceleration magnitude was less than or equal to 1.8 g were not considered for algorithm development and were automatically categorized as periods of inactivity or ADLs. Therefore, our classification method aims at accurately distinguishing between high acceleration ADLs and falls.

Features were extracted from acceleration signals and context data obtained from ToF wireless sensors. The acceleration vector $\vec{a} = (a_1, a_2, a_3)$ is made up of the three acceleration measurements taken at a given point in time. Acceleration features included statistics of the magnitude of the acceleration (minimum, maximum, range, mean, median, standard deviation, skewness, and kurtosis) and estimated acceleration-based displacement for given time intervals: five-second fall candidate (centered at the acceleration peak) as well as pre-impact (with duration of two seconds before the impact), impact (with duration of two seconds immediately after the impact), and post-impact (with duration of three seconds counted after two seconds of the impact) stages.

Also, the orientation of the acceleration sensor was estimated using gravity as presented in [48]. Following this approach, the vertical acceleration vector $\vec{g} = (g_1, g_2, g_3)$, corresponding to gravity, is estimated by first averaging all the measurements on the respective acceleration axes for a sampling interval of a few seconds. In this work, we empirically chose a two-second window to estimate the gravity vector. Next, the dynamic acceleration due to person's activity is calculated as $\vec{a}_d = (a_1 - g_1, a_2 - g_2, a_3 - g_3)$. Finally, the vertical component of the dynamic acceleration vector at a given time a_d^v is calculated using vector dot products as presented in Equation (1).

$$\vec{a}_d^v = \left(\frac{\vec{a}_d \bullet \vec{g}}{|\vec{g}|} \right) \frac{\vec{g}}{|\vec{g}|} \tag{1}$$

Using the estimated gravity vector $\vec{g}$, the angle between the pre-impact gravity vector $\vec{g}_{pre-impact}$ and the post-impact gravity vector $\vec{g}_{post-impact}$ was calculated to quantify the orientation change of the acceleration sensor possibly due to a fall event (see (2)). Additional statistical features are calculated from the vertical velocity and accelerometer-based displacement during the various stages of a fall candidate.

$$\theta = \cos^{-1} \left(\frac{\vec{g}_{pre-impact} \bullet \vec{g}_{post-impact}}{|\vec{g}_{pre-impact}||\vec{g}_{post-impact}|} \right) \tag{2}$$

In addition to acceleration-based features, we consider context features including the time of day of the activity, day of week, as well as movement before and after a high acceleration peak. Accurate displacement features were derived from ToF sensor data, which were used to track the indoor position and velocity of a person using advanced Kalman filtering techniques [46]. Relevant features were selected using the mutual information criterion, and highly correlated features with correlation coefficient greater than 0.85 were removed.

A complete list of the most relevant acceleration features for MS fall detection is presented in Table II. Table II also shows the mutual information value and the relevance ranking of the features that were selected as important for classifying simulated falls. Note that the three best features for predicting real-world falls are not the same as the features for predicting simulated falls. Table III shows the most relevant features obtained from context data, which is available only in the real-world MS falls dataset. From the list of selected features, it can be seen that

TABLE II
MOST RELEVANT ACCELERATION FEATURES FOR MS FALL DETECTION RANKED BY MUTUAL INFORMATION CRITERION

Feature	Mutual information			Signal pre-processing		Segment			
	MS	SisFall	SisFall rank	Raw	Low-pass-filtered (fc = 5.0 Hz)	5-second fall candidate	Pre-impact	Impact	Post-impact
Max(vertical velocity)	0.501	-	-	✓					✓
Acceleration-based displacement	0.491	-	-	✓					✓
Skew(acceleration magnitude)	0.471	0.390	3	✓				✓	
Estimated sensor orientation change	0.453	0.407	1	✓			✓		✓
Periodicity of vertical acceleration signal	0.432	0.196	16		✓				✓
Range(vertical velocity range)	0.427	0.398	2	✓		✓			
Skew(acceleration magnitude)	0.364	-	-	✓		✓			
Kurtosis(acceleration magnitude)	0.360	-	-		✓	✓			
Kurtosis(acceleration magnitude)	0.345	-	-		✓			✓	
Standard deviation magnitude	0.344	-	-	✓					✓
Acceleration standard deviation magnitude	0.333	-	-		✓	✓			
Max(acceleration magnitude)	0.329	-	-	✓					✓
Range(vertical velocity)	0.312	-	-	✓				✓	
Kurtosis(vertical velocity)	0.301	0.316	5	✓		✓			
Acceleration standard deviation magnitude	0.294	0.266	8		✓				✓
Std(vertical velocity)	0.294	-	-	✓					✓
Skew(acceleration magnitude)	0.292	-	-		✓			✓	
Min(acceleration magnitude)	0.266	0.246	10	✓					✓
Mean(acceleration magnitude)	0.260	-	-	✓					✓
Kurtosis(acceleration magnitude)	0.250	0.291	7		✓		✓		
Max(acceleration magnitude)	0.241	-	-	✓				✓	
Min(vertical velocity)	0.211	-	-	✓			✓		
Median(acceleration magnitude)	0.206	-	-	✓		✓			

TABLE III
MOST RELEVANT CONTEXT FEATURES FOR MS FALL DETECTION RANKED BY MUTUAL INFORMATION CRITERION

Feature	Mutual information
Movement 15 s post-impact	0.186
Hour of day	0.071
Day of week	0.010

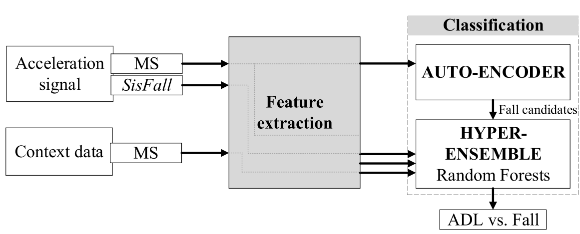


Fig. 3. Flow diagram of the method for automated fall detection.

features extracted using data from the post-impact stage are the most relevant features for distinguishing between fall events and high acceleration ADLs.

D. Machine-Learning-Based Fall Detection

This section discusses our proposed fall detection algorithm, which is a two-stage classification method composed of 1) an auto-encoder for initial detection of fall candidates based on acceleration data, followed by 2) a hyper-ensemble of balanced random forests trained using both acceleration and movement features. The auto-encoder was tuned to capture all falls within its fall candidate list while the hyper-ensemble random forest was designed to eliminate the false-positives within the fall candidate list (see Fig. 3).

1) Detection of Fall Candidates by Neural Network Auto-Encoder: An auto-encoder is an unsupervised learning algorithm that approximates the function $y = f(x) = x$ [49]. The output of the auto-encoder is expected to yield a close estimate of the input. We developed an auto-encoder to reconstruct the acceleration input data and detect fall candidates when the input acceleration signal was reconstructed with a root-mean-square error (RMSE) higher than 0.057. The selection of the cutoff RMSE was based on the minimum reconstruction error calculated for the training fall segments. To ensure that nearly all falls were detected, the cutoff RMSE was set such that all falls were detected in the training set (i.e., sensitivity in the training set was set to 100%).

2) Final Label Assignment by Hyper-Ensemble of Random Forests: The hyper-ensemble of random forests was designed to reduce the high number of false positives from the fall candidate list output by the auto-encoder. We designed the hyper-ensemble of random forests (RF) similar to the approach presented in [50] to handle the high level of data imbalance present in our real-world fall data. In order to address this imbalance and to make use of all the available data, which mainly provide valuable information about ADLs (majority class), we performed the following steps: i) data augmentation, ii) sampling of data corresponding to ADLs, and iii) hyper-ensemble of RFs trained on balanced subsets. An illustration of the proposed hyper-ensemble classification method is shown in Fig. 4.

Notice in Fig. 4 on the left panel, the real-world MS falls (black), the simulated *SisFall* falls (dark gray), and the non-falls / ADLS (light gray) are shown together as rectangles within the entire dataset (X). The real-world dataset is referred to as D and the simulated data set as S. A positive example (i.e. a fall event) is referred to as P while a negative example (i.e. non-fall event) is referred to as N. In this way, the real-world dataset can be defined as a union of both positive and negative examples

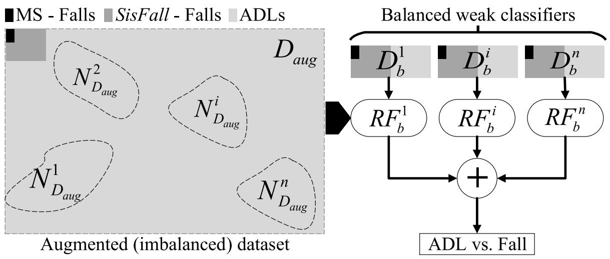


Fig. 4. The hyper-ensemble of random forest classifiers.

$D = P_D \cup N_D = \{x|x \in P_D \vee x \in N_D\}$. The simulated data is also a union of positive and negative examples $S = P_S \cup N_S = \{x|x \in P_S \vee x \in N_S\}$.

1) ***Data Augmentation:*** This involves augmenting the real-world fall events with simulated fall events. This type of over-sampling is comparable with other methods of oversampling such as the Synthetic Minority Over-sampling Technique (SMOTE) [51]. SMOTE overcomes imbalance by generating additional observation examples from an existing set based on statistical sampling of the existing set. Unlike SMOTE, we used simulated falls to generate additional observations. A total of 3,592 examples were added to the training dataset, corresponding to 15 fall types performed five times each by 24 people and recorded using two accelerometers; eight examples were excluded that couldn't be read.. Simulated falls are combined with the real-world falls to form the augmented dataset (D_{aug}) whereby $D_{aug} = P_{D_{aug}} \cup N_{D_{aug}} = \{x|x \in P_D \vee x \in P_S\} \cup \{x|x \in N_D \vee x \in N_S\}$.
2) ***ADL Sampling:*** Following data augmentation, multiple balanced datasets were generated, each one comprising an equal number of fall and non-fall events. The number of balanced sets (n) was calculated by dividing the total number of non-fall events by the total number of fall events ($n = |N_{D_{aug}}|/|P_{D_{aug}}|$) in the augmented data set. Each balanced set (D_b^i) comprises all of the augmented fall events ($P_{D_{aug}}$) and a sample (i) of the augmented non-fall events ($N_{D_{aug}}^i$) such that $D_b^i = P_{D_{aug}} \cup N_{D_{aug}}^i$. Each balanced dataset (D_b^i) is considered a weak representation, because it does not ensure sufficient coverage of all of the non-fall events.
3) ***Hyper-Ensemble of Random Forests:*** After data augmentation and generation of the balanced data sets, we trained n RFs, one for each balanced D_b^i. An RF is a robust classifier consisting of a collection of decision trees $\{h(x, \Theta_k), k \in \{1, 2, \ldots, K\}\}$ where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class given an input feature vector x [52]. Since each balanced D_b^i does not cover all the available ADLs data, all of the weak representations are combined using a hyper-ensemble of multiple RFs, whereby each forest is trained on a separate balanced data set. By combining the predictions of multiple RFs two goals can be achieved: full coverage of the dataset and more diverse base classifiers that can eventually improve the overall ensemble's accuracy.

IV. Results

Various general-purpose prediction performance metrics were used to assess the accuracy of our proposed fall detection system, including the area under the receiver operating curve (AUC), sensitivity, specificity, false alarms per day, and overall accuracy computed as the average of the sensitivity and specificity as defined in [12]. This section first shows the performance of our prediction algorithm on real-world falls and compares our algorithm with two SVM classifiers trained and tested on the same real-world dataset. Next, the performance of the algorithm is evaluated just on the simulated falls, since a direct performance comparison can be done with another published algorithm by Sucerquia et. al in [12] who also evaluated their algorithm on the *SisFall* dataset.

A. Cross-Validation Approach

For real-world and simulated falls, the accuracy of our fall detection method is reported based on the k-fold cross-validation approach [53]. In k-fold cross-validation, the original sample is randomly partitioned into k equally sized sub-samples. At every iteration, one of the k sub-samples is held out for model validation while the remaining k-1 sub-samples are used as training data. For the real-world accuracy assessment, we performed participant-level cross-validation to evaluate the performance of our method on a real-world scenario. In this setting, k = 10 training/validation folds were generated ensuring that each testing fold had at least one fall. This was done to estimate the overall AUC of the algorithm. The cross-validation process was then repeated k times, with each of the k sub-samples used exactly once as the validation data. The results from the k training-validation runs are averaged (or otherwise combined) to produce a single performance estimation. The averaged sensitivity values were obtained for the participants who fell during the study, and the amount of false positive alarms per day were calculated across all participants.

For simulated falls, we also chose k = 10 to directly compare our results with those presented by Sucerquia *et al.* in [12].

B. Fall Detector Performance

1) Performance of the Algorithm on Real-World Data: The algorithm performed well on the real-world fall detection with a sensitivity of 92.14 ± 13.65, specificity of 98.48 ± 1.17,and a false positive rate of 0.65 ± 0.55 false alarms per day as given in Table IV. Also shown in Table IV is the trade-off in sensitivity in terms of reduced false alarms per day that can be achieved by incorporating the hyper-ensemble random forest. Our goal was to achieve substantially fewer than one false positive per day. Incorporating contextual information enabled us to reduce the false positive rate by 48.4%, but it was at the cost of slightly lower sensitivity (94.11% vs. 92.14%).

TABLE IV
PERFORMANCE OF FALL DETECTION METHODS ON REAL-WORLD FALLS COLLECTED FROM PATIENTS WITH MS

Performance metric	Mean ± Std		
	Auto-encoder	Hyper RF (acceleration features)	Hyper RF (acceleration + context features)
AUC	0.914 ± 0.065	0.989 ± 0.005	0.993 ± 0.004
Sensitivity, %	98.33 ± 4.41	94.11 ± 12.34	92.14 ± 13.65
Specificity, %	61.86 ± 22.59	98.21 ± 1.31	98.48 ± 1.17
Overall accuracy, %	80.10 ± 10.70	96.16 ± 5.83	95.31 ± 6.52
False alarms per day	22.71 ± 13.05	1.26 ± 0.55	0.65 ± 0.55

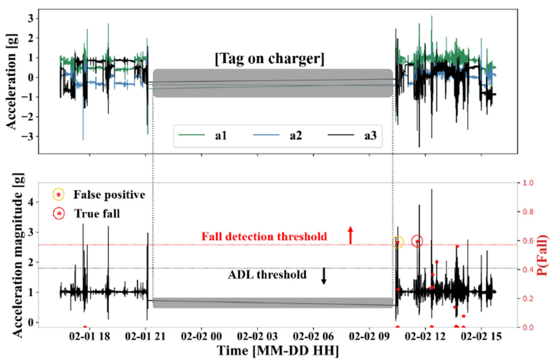


Fig. 5. Example of acceleration data recorded from a study participant during 24 hours (top) and fall detection results (bottom).

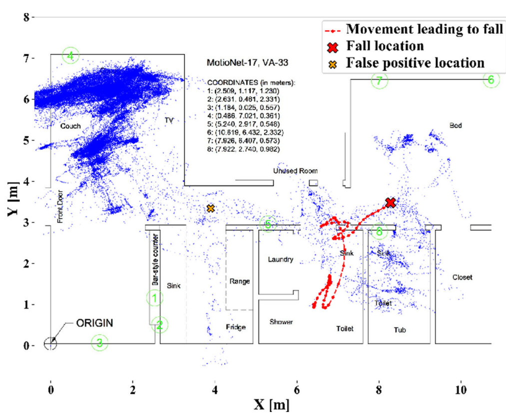


Fig. 6. Example of movement tracking and fall localization during a 24-hour period.

Figure 5 shows an example of tri-axial acceleration data recorded from a study participant during 24 hours. The probability of a fall event is calculated based on the proportion of trees in the hyper-ensemble that classified a given fall candidate event as a fall. In Fig. 5, there are two falls detected by our model, which correspond to one true fall and a false positive alarm. Figure 6 shows an example of a participant's indoor movement tracking and the location of detected fall events during the same 24-hour period shown in Figure 5.

In order to compare the performance of our proposed imbalance-aware fall detection algorithm with previously published work on our real-world falls dataset, we built a linear SVM model and a SVM model with a radial basis function (RBF) kernel, which were trained using acceleration and context features. We selected the SVM approach since it yielded the highest accuracy in fall detection in a comprehensive comparative review of threshold and machine-learning-based classification models published in [54]. The linear SVM classifier achieved AUC of 0.92 ± 0.13, sensitivity of 92.67 ± 15.93%, specificity of 94.76 ± 6.05%, and 12.5 ± 29.62 false positive per day. On the other hand, the SVM model with RBF kernel achieved AUC of 0.97 ± 0.04, sensitivity of 96.67 ± 10.54%, specificity of 96.25 ± 4.61%, and 4.0 ± 3.46 false positive per day. If the sensitivity of the autoencoder plus random forest fall detector classifier described in the current paper is set to 97.22 ± 4.37%, it would result in a specificity of 98.29 ± 1.08% and 2.43 ± 2.1 false positive alarms a day. In summary, both SVM models performed comparatively with our proposed model in terms of sensitivity, but with at least 64.6% higher false positive per day rates.

2) Analysis of the Effect of Context on Classification Accuracy: The improvements in the false positive rate when adding relevant context features might be mainly due to the accuracy in the estimation of the post-impact displacement descriptor using ToF tracking data. Note that the total post-impact displacement estimated from acceleration data is also ranked high among inertial features. However, it is known that acceleration-based displacement estimation is often inaccurate due to the noisy nature of acceleration signals. Intuitively, this feature should be highly relevant as it might be expected that a person who has fallen would not move very much immediately after a fall. Our data showed that during the 15 seconds following a true fall event, study participants with MS tended to move only an average of 0.76 m compared with 2.59 m when they were performing an ADL that may have looked like a fall. Data also showed that most falls in people with MS tend to occur during periods of high activity between noon and 6:00 p.m., making it challenging to distinguish falls from confounding ADLs.

3) Performance of the Algorithm on Simulated Data: To compare our methods with those of others who have published on the *SisFall* dataset, we trained an RF classifier with data from the *SisFall* dataset using our machine-learning pipeline (i.e., features extraction, feature selection, and RF classification) and compared the performance with a threshold based fall detector [12]. For the simulated *SisFall* dataset, the hyper-ensemble of random forest was not necessary because the *SisFall* dataset is not highly imbalanced. The most relevant features to distinguish between falls and ADLs in the *SisFall* dataset by the mutual information criterion (mutual information greater than 0.2) were used to train the classifier after removing highly correlated features (correlation coefficient greater than 0.85). The mean and standard deviation of the performance indicators

TABLE V
PERFORMANCE OF OUR MACHINE-LEARNING FALL DETECTION METHODS ON THE *SisFall* DATASET (SIMULATED FALLS)

Performance metric	Mean ± Std
AUC	1.00 ± 0.00
Sensitivity, %	99.81 ± 0.22
Specificity, %	99.99 ± 0.02
Overall accuracy, %	99.90 ± 0.11

obtained trough 10-fold cross-validation are shown in Table V. The overall accuracy achieved by our method was 99.90 ± 0.11% compared with 99.39 ± 0.36% achieved by Sucerquia et. al in [12] using the same dataset. This demonstrated the robustness of our machine-leaning-based approach in accurately detecting simulated falls. It also emphasizes that higher accuracy can be expected in simulated falls compared with real-world falls.

V. CONCLUSION

This paper presented an imbalance-aware machine-learning-based classification algorithm to automatically detect falls in people with MS. This is the first detector to be trained and retrospectively tested on real-world falls. The developed fall detection system used inertial data from a tri-axial accelerometer and contextual movement data from time-of-flight indoor wireless transceivers. Our system achieved a sensitivity of 92.14% on real-world falls, which is associated with a false positive rate of 0.65 false alarms per day. Incorporating contextual features was helpful in substantially reducing the false positive rate. While our method achieved a false positive rate of less than one false alarm per day, the usability of the system would be improved by further reducing the number of false alarms, if this could be done without compromising the system's sensitivity. The optimal balance between sensitivity and specificity, and thus the risks associated with missing falls compared to those associated with false alarms, is likely to vary by application and setting. We expect that, in most settings, high sensitivity and thus not missing falls is of high value while false alarms are less problematic as long as they are not so frequent as to cause "alarm fatigue". We propose that less than one false alarm per day is probably acceptable in most settings and that less than one false alarm per week would be ideal.

Our proposed method belongs to the category of multimodal machine-learning-based systems (i.e., it uses more than one class of sensors). Unlike many of the systems discussed in Section II, which make decisions using heuristics or hard rules based on the distribution of acceleration magnitude features during the different fall stages, our method combines semi-supervised and supervised learning into a two-stage data-driven model for anomaly detection and falls vs. ADLs classification. Although our algorithm is more computationally expensive than threshold-based algorithms during both training and prediction phases, our system has a good balance between sensitivity and specificity in real-world scenarios, which has not been demonstrated by previously developed threshold-based algorithms. For example, the algorithms evaluated in Bagala *et al.* [26] achieved an average sensitivity of 57% with false alarm rates of 3-85 per day.

Moreover, previously developed machine-learning-based fall detectors reported sensitivities higher than 80% with an associated false alarm rate higher than one false alarm per 24 hour period when tested on real-world scenarios. The reason for the decline in the accuracy of the detector (lower sensitivity and specificity) is that the training phase is usually done using simulated falls and data from the target population is not dynamically learned during the validation of the system.

This work demonstrated the value of developing fall detection methods using real-world fall data collected from the target population in improving the performance and usability of automated fall detection systems. We noted that the accuracy of our system is slightly lower for real-world falls than for simulated falls, but is still better than other published algorithms tested on older adults under free-living conditions.

It is important to view this work within the context that the algorithm presented here was trained and evaluated on a cohort of people with MS. There has not yet been a study comparing how wearable sensor measurements differ in capturing real-world falls in people with MS compared to healthy or senior populations. However, prior studies have shown that the reasons for falling can be different in healthy adults compared with people with MS [34], and this may impact the fall information captured by the sensors. Healthy people may respond more rapidly to a fall event than a person with MS as people with MS tend to have a delayed response to postural change events.

For our future work, we will explore how this algorithm works on new data sets from different cohorts of people. And we plan to leverage the collected context data to generate an algorithm for assessing fall risk based on objective metrics derived from peoples' activity levels and movement patterns to further reduce the false-positive alarm rate.

Conflict: This potential conflict of interest has been reviewed and managed by OHSU.

REFERENCES

[1] U.S. Department of Housing and Urban Development. (2017) Overcoming obstacles to policies for preventing falls by the elderly. [Online]. Available: https://www.hud.gov/sites/dfiles/HH/documents/OvercomingObstaclesFalls.pdf
[2] D. Wild, U. S. Nayak, and B. Isaacs, "How dangerous are falls in old people at home?," *Brit. Med. J. (Clin. Res. Ed.)*, vol. 282, pp. 266–268, 1981.
[3] M. Y. Wang *et al.*, "Activity limitation due to a fear of falling in older adults with eye disease," *Invest. Ophthalmol. Vis. Sci.*, vol. 53, no. 13, pp. 7967–7972, 2012.
[4] J. C. Castillo, D. Carneiro, J. Serrano-Cuerda, P. Novais, A. Fernández-Caballero, and J. Neves, "A multi-modal approach for activity classification and fall detection," *Int. J. Syst. Sci.*, vol. 45, no. 4, pp. 810–824, 2014.
[5] B. Kwolek and M. Kepski, "Human fall detection on embedded platform using depth maps and wireless accelerometer," *Sensors (Basel)*, vol. 117, no. 3, pp. 489–501, 2014.
[6] J. Wang, Z. Zhang, B. Li, S. Lee, and R. S. Sherratt, "An enhanced fall detection system for elderly person monitoring using consumer home networks," *IEEE Trans. Consum. Electron.*, vol. 60, no. 1, pp. 23–29, Feb. 2014.
[7] B. Kwolek and M. Kepski, "Improving fall detection by the use of depth sensor and accelerometer," *Neurocomputing*, vol. 168, no. 30, pp. 637–645, 2015.

[8] L. J. Kau and C. S. Chen, "A smart phone-based pocket fall accident detection, positioning, and rescue system," *IEEE J. Biomed. Health Inform.*, vol. 19, no. 1, pp. 44–56, 2015.
[9] P. Pierleoni, A. Belli, L. Palma, M. Pellegrini, L. Pernini, and S. Valenti, "A high reliability wearable device for elderly fall detection," *IEEE Sens. J.*, vol. 15, no. 8, pp. 4544–4553, 2015.
[10] J. K. Lee, S. N. Robinovitch, and E. J. Park, "Inertial sensing-based pre-impact detection of falls involving near-fall scenarios," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 23, no. 2, pp. 258–266, Mar. 2015.
[11] A. Sucerquia, J. D. Lopez, and J. F. Vargas-Bonilla, "Sisfall: A fall and movement dataset," *Sensors (Basel)*, vol. 17, no. 1, p. 198, 2017.
[12] A. Sucerquia, J. D. Lopez, and J. F. Vargas-Bonilla, "Real-life/real-time elderly fall detection with a triaxial accelerometer," *Sensors (Basel)*, vol. 18, no. 4, 2018, Art. no. 1101.
[13] J. L. Chua, Y. C. Chang, and W. K. Lim, "A simple vision-based fall detection technique for indoor video surveillance," *Signal Image Video Process.*, vol. 9, no. 3, pp. 623–633, 2015.
[14] G. Mastorakis and D. Makris, "Depth-based human fall detection via shape features and improved extreme learning machine," *IEEE J. Biomed. Health Inform.*, vol. 18, no. 6, pp. 1915–1922, Nov. 2014.
[15] S. Gasparrini, E. Cippitelli, S. Spinsante, and E. Gambi, "Depth-based fall detection system using a kinect," *Sensors (Basel)*, vol. 14, no. 2, pp. 2756–2775, 2014.
[16] G. Mastorakis and D. Makris, "Fall detection system using kinect's infrared sensor," *J. Real Time Image Process.*, vol. 9, no. 4, pp. 635–646, 2014.
[17] Y. Li, K. C. Ho, and M. Popescu, "Efficient source separation algorithms for acoustic fall detection using a microsoft kinect," *IEEE Trans. Biomed. Eng*, vol. 61, no. 3, pp. 745–755, Mar. 2014.
[18] E. E. Stone and M. Skubic, "Fall detection in homes of older adults using the microsoft kinect," *IEEE J. Biomed. Health Inform.*, vol. 19, no. 1, pp. 290–301, Jan. 2015.
[19] Z. P. Bian, J. Hou, L. P. Chau, and N. Magnenat-Thalmann, "Fall detection based on body part tracking using a depth camera," *IEEE J. Biomed. Health Inform.*, vol. 19, no. 2, pp. 430–439, Mar. 2015.
[20] B. Y. Su, K. C. Ho, M. J. Rantz, and M. Skubic, "Doppler radar fall activity detection using the wavelet transform," *IEEE Trans. Biomed. Eng.*, vol. 62, no. 3, pp. 865–875, Mar. 2015.
[21] M. G. Amin, Y. D. Zhang, F. Ahmad, and K. D. Ho, "Radar signal processing for elderly fall detection: The future for in-home monitoring," *IEEE Signal Process. Mag.*, vol. 33, no. 2, pp. 71–80, Mar. 2016.
[22] Y. Wang, K. Wu, and L. M. Ni, "Wifall: Device-free fall detection by wireless networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 2, pp. 581–594, Feb. 2017.
[23] L. Schwickert *et al.*, "Fall detection with body-worn sensors : A systematic review," *Z. Gerontol. Geriatr.*, vol. 46, no. 8, pp. 706–719, 2013.
[24] B. Aguiar, T. Rocha, J. Silva, and I. Sousa, "Accelerometer-based fall detection for smartphones," in *Proc. IEEE Int. Symp. Med. Meas. Appl.*, 2011, pp. 3461–3469.
[25] "Comparison of acceleration signals of simulated and real-world backward falls," *Med. Eng. Phys.*, vol. 33, no. 3, pp. 368–373, 2011.
[26] F. Bagala *et al.*, "Evaluation of accelerometer-based fall detection algorithms on real-world falls," *PLoS One*, vol. 7, no. 5, 2012, Art. no. e37062.
[27] M. Kangas, I. Vikman, L. Nyberg, R. Korpelainen, J. Lindblom, and T. Jamsa, "Comparison of real-life accidental falls in older people with experimental falls in middle-aged test subjects," *Gait Posture*, vol. 35, no. 3, pp. 500–505, 2012.
[28] World Health Organization. (2018) Falls. [Online]. Available: https://www.who.int/en/news-room/fact-sheets/detail/falls
[29] T. Xu, Y. Zhou, and J. Zhu, "New advances and challenges of fall detection systems: A survey," *Appl. Sci.*, vol. 8, no. 3, p. 418, 2018.
[30] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 4, pp. 463–484, Jul. 2012.
[31] D. Cattaneo, C. De Nuzzo, T. Fascia, M. Macalli, I. Pisoni, and R. Cardini, "Risks of falls in subjects with multiple sclerosis," *Arch. Phys. Med. Rehabil.*, vol. 83, no. 6, pp. 864–867, 2002.
[32] Y. Nilsagard, C. Lundholm, E. Denison, and L. G. Gunnarsson, "Predicting accidental falls in people with multiple sclerosis—A longitudinal study," *Clin. Rehabil.*, vol. 23, no. 3, pp. 259–269, 2009.
[33] P. N. Matsuda, A. Shumway-Cook, A. M. Bamer, S. L. Johnson, D. Amtmann, and G. H. Kraft, "Falls in multiple sclerosis," *PMR*, vol. 3, no. 7, pp. 624–632, 2011.
[34] R. Mazumder, C. Murchison, D. Bourdette, and M. Cameron, "Falls in people with multiple sclerosis compared with falls in healthy controls," *PLoS One*, vol. 9, no. 9, p. e107620, 2014.
[35] M. T. Wallin *et al.*, "The prevalence of MS in the United States: A population-based estimate using health claims data," *Neurology*, vol. 92, no. 10, pp. 1029–1040, 2019.
[36] P. Morell and R. H. Quarles, *Basic Neurochemistry: Molecular, Cellular and Medical Aspects*, 6th ed. Philadelphia, PA: Lippincott-Raven, 1999, ch. Myelin Formation, Structure and Biochemistry.
[37] National Multiple Sclerosis Society. (2018) What is MS? [Online]. Available: https://www.nationalmssociety.org/What-is-MS
[38] M. H. Cameron and Y. Nilsagard, *Handbook of Clinical Neurology*. Amsterdam, The Netherlands: Elsevier, 2018, vol. 159, ch. Balance, gait, and falls in multiple sclerosis, pp. 237–250.
[39] E. W. Peterson, C. C. Cho, and M. L. Finlyson, "Fear of falling and associated activity curtailment among middle aged and older adults with multiple sclerosis," *Mult. Scler.*, vol. 13, pp. 1168–1175, 2007.
[40] S. E. Lamb, E. C. Jorstad-Stein, K. Hauer, C. Becker, and Prevention of Falls Network Europe and Outcomes Consensus Group, "Development of a common outcome data set for fall injury prevention trials: the prevention of falls network Europe consensus," *J. Amer. Geriatr. Soc.*, vol. 53, no. 9, pp. 1618–1622, 2005.
[41] S. Coote, J. J. Sosnoff, and H. Gunn, "Fall incidence as the primary outcome in multiple sclerosis falls-prevention trials: Recommendation from the international MS falls prevention research network.," *Int. J. MS Care*, vol. 16, no. 4, pp. 178–184, 2014.
[42] K. Chaccour, R. Darazi, A. H. El Hassani, and E. Andres, "From fall detection to fall prevention: A generic classification of fall-related systems," *IEEE Sensors J.*, vol. 17, no. 3, pp. 812–822, 2017.
[43] O. Aziz *et al.*, "Validation of accuracy of SVM-based fall detection system using real-world fall and non-fall datasets," *PLoS One*, vol. 12, no. 7, p. e0180318, 2017.
[44] L. A. Lipsitz *et al.*, "Evaluation of an automated falls detection device in nursing home residents," *J. Amer. Geriatr. Soc.*, vol. 64, no. 2, pp. 365–368, 2016.
[45] MotioSense Inc. (2015) Motiowear. [Online]. Available: www.motiosens.com
[46] A. Paul, E. Wan, and P. Jacobs, "Sigma-point kalman smoothing for indoor tracking and auto-calibration using time-of-flight ranging," in *Proc. Inst. Navig. GNSS Conf.*, 2011, pp. 3461–3469.
[47] A. Ibrahim and M. Younis, "Simple fall criteria for mems sensors: Data analysis and sensor concept," *Sensors*, vol. 14, pp. 12 149–12 173, 2014.
[48] D. Mizell, "Using gravity to estimate accelerometer orientation," in *Proc. IEEE Int. Symp. Wearable Comput.*, 2003, pp. 252–253.
[49] P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural Netw.*, vol. 2, pp. 53–58, 1989.
[50] M. Schubach, M. Re, P. N. Robinson, and G. Valentini, "Imbalance-aware machine learning for predicting rare and common disease-associated non-coding variants," *Nature Sci. Rep.*, vol. 7, no. 1, p. 2959, 2017.
[51] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2012.
[52] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[53] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *J. Roy. Statist. Soc.*, vol. B, no. 36, pp. 111–133, 1974.
[54] O. Aziz, M. Musngi, E. Park, G. Mori, and S. N. Robinovitch, "A comparison of accuracy of fall detection algorithms (threshold-based vs. machine learning) using waist-mounted tri-axial accelerometer signals from a comprehensive set of falls and non-fall trials," *Med Biol Eng Comput*, vol. 55, pp. 45–55, 2017.