

Article

# Novel Hierarchical Fall Detection Algorithm Using a Multiphase Fall Model

Chia-Yeh Hsieh, Kai-Chun Liu, Chih-Ning Huang, Woei-Chyn Chu and Chia-Tai Chan \*

Department of Biomedical Engineering, National Yang-Ming University, Taipei 112, Taiwan; kerrhsieh@ym.edu.tw (C.-Y.H.); g30104026@ym.edu.tw (K.-C.L.); g39604001@ym.edu.tw (C.-N.H.); wchu@ym.edu.tw (W.-C.C.)

\* Correspondence: ctchan@ym.edu.tw; Tel.: +886-2-2826-7371

Academic Editors: Subhas Chandra Mukhopadhyay, Hemant Ghayvat and Nagender Kumar Suryadevara

Received: 24 November 2016; Accepted: 3 February 2017; Published: 8 February 2017

**Abstract:** Falls are the primary cause of accidents for the elderly in the living environment. Reducing hazards in the living environment and performing exercises for training balance and muscles are the common strategies for fall prevention. However, falls cannot be avoided completely; fall detection provides an alarm that can decrease injuries or death caused by the lack of rescue. The automatic fall detection system has opportunities to provide real-time emergency alarms for improving the safety and quality of home healthcare services. Two common technical challenges are also tackled in order to provide a reliable fall detection algorithm, including variability and ambiguity. We propose a novel hierarchical fall detection algorithm involving threshold-based and knowledge-based approaches to detect a fall event. The threshold-based approach efficiently supports the detection and identification of fall events from continuous sensor data. A multiphase fall model is utilized, including free fall, impact, and rest phases for the knowledge-based approach, which identifies fall events and has the potential to deal with the aforementioned technical challenges of a fall detection system. Seven kinds of falls and seven types of daily activities arranged in an experiment are used to explore the performance of the proposed fall detection algorithm. The overall performances of the sensitivity, specificity, precision, and accuracy using a knowledge-based algorithm are 99.79%, 98.74%, 99.05% and 99.33%, respectively. The results show that the proposed novel hierarchical fall detection algorithm can cope with the variability and ambiguity of the technical challenges and fulfill the reliability, adaptability, and flexibility requirements of an automatic fall detection system with respect to the individual differences.

**Keywords:** fall detection algorithm; multiphase fall model; wearable sensor

## 1. Introduction

According to the International Database of the U.S. Census Bureau, the average percentage of individuals over 65 years old was 17% of the population in more developed countries, such as the United States, Japan, Germany, etc., in 2015, and this percentage is expected to reach 30% in 2050. Population aging is a common phenomenon in many countries, particular developing and developed countries, due to declining fertility rates and increasing life expectancy [1]. Falls among the elderly are public health issues and threats which can implicate quality of life and cause severe disabilities that impact independent living [2,3]. Fall-related injuries among the fallers found helpless in their homes are more serious than those among people who are helped within 12 h [4]. The association between the mortality rates and waiting time for rescue shows a positive correlation [4,5]. In addition to causing disability or functional impairment, non-fatal injurious falls also have psychological and social effects [6]. Fear of falling again makes the elderly lose self-confidence in their ambulant safety, limiting the function of activities of daily living (ADLs) [7].

In order to improve the quality of home healthcare services, two kinds of strategies are proposed to deal with the occurrence of falls, including fall prevention and fall detection [8]. The fall prevention strategy analyzes risk factors, and then delivers targeted interventions to reduce the incidence of falls [9,10]. The environmental factors include obstacles, poor light, loose carpets, lack of safety equipment, and weather; the most frequently mentioned physiological reasons for falls are losing balance, a history of falls, functional and cognitive impairments, medication use, postural blood pressure, muscle weakness, and visual impairment [7,11]. Existing ways to prevent falls from occurring include muscular strength and body balance training, and a preventative checklist can be created to minimize the risk of falls in inimical environments. Unfortunately, falls cannot be completely prevented, so a fall detection system is essential for the elderly.

A fall detection system is considered an assistive system aiming to send an alert when a fall event happens. There are two types of fall detection systems, including user-manual and automatic systems. User-manual fall detection systems are designed to send emergency messages by user manipulation. However, such systems cannot provide first aid for the faller during loss of consciousness. Alternatively, automatic fall detection systems are proposed to detect falls without any user manipulation when the faller loses consciousness. Automatic fall detection systems can be classified into two main categories according to their sensor type: one is an ambient device-based fall detection system, while the other is a wearable-based fall detection system [12,13]. Ambient devices are arranged in the smart environment, and they include cameras [14], infrared sensors, acoustic sensors, vibration sensors, and pressure sensors [15]. These ambient devices have good performance in confined environments, such as living rooms, bathrooms, or laboratories. However, these devices are not feasible in uncontrolled environments. Wearable sensor-based fall detection systems, by contrast, can detect falling at any time, anywhere, while the user wears the sensor(s). Lin et al. presented a wearable micro-mercury switch integrated with an optical sensor to enhance the fall detection rate of the wearable device [16]. Chen et al. used a tri-axial accelerometer attached to the waist to detect falls and a network of fixed nodes to locate the victim [17]. Bourke and Lyons used only one bi-axial gyroscope mounted on the trunk to detect falls [18]. In this work, the proposed fall detection algorithm aims to automatically detect a critical fall, where fall is defined as “an event in which a person inadvertently comes to rest on the ground, floor, or lower level as a consequence of the following: sustaining a violent blow, loss of consciousness, sudden onset of paralysis, as in a stroke or an epileptic seizure”, given the person cannot call for help by himself/herself [11,19].

A fall detection algorithm plays an important role in an automatic fall detection system. There are two common technical issues and challenges that should be tackled in order to provide a reliable fall detection algorithm. The first is variability, as falls can happen involuntarily and suddenly in various forms and directions in daily living. Falls might happen when you are walking, standing, and they frequently happen in activity transitions, such as during bed and chair exits [20]. The second is ambiguity, as some features of falls are similar to the features of ADLs, which might confuse the fall detection system. For example, serious falling events can occur with strong impact and energy, which are similar to jumping or running in daily living. Additionally, light falling events occur with a smaller impact and energy than those of serious falling events, but the light falling event might be similar to lying down in daily living. These technical issues and challenges may lead to most fall detection algorithms being insufficient for automatic fall detection systems when only the specific event is considered, such as impact, posture after a fall, and change of velocity during a fall. Several well-known studies have proposed a multiphase fall model to provide more fine-grained observation of the fall event for automatic fall detection systems, where the fall can be divided into different phases, such as four phases (the pre-fall, critical, post-fall, and recovery phases) [21], and five phases (the pre-fall, falling, impact, resting, and recovery phases) [22]. For instance, the post-fall and pre-fall activity might highly influence the impact signal. Therefore, a multiphase fall model has the potential to deal with the aforementioned technical challenges and to provide a more fine-grained level of information for a fall detection system.

This study aims to accurately detect falls during the execution of activities of daily living using wearable sensors. We propose a novel hierarchical fall detection algorithm including threshold-based and knowledge-based approaches to detect a fall event. Firstly, the sets of the thresholds are determined to identify the absolute falls and ADLs in a threshold-based approach. The advantages of a threshold-based approach are the low computing complexity and ease of implementation. However, it is difficult to set an appropriate threshold value since there is an overlap of peak acceleration values generated by falls and ADLs. Therefore, the ambiguous problem is solved by the knowledge-based approach. Relying on a multiphase fall model, including free fall, impact, and rest phases, knowledge-based approaches can improve the performance matrices of the simple threshold-based fall detection algorithms.

The rest of this work is organized as follows: In Section 2, we briefly expand on the related work of fall detection, such as threshold-based and machine learning-based fall detection algorithms. The proposed hierarchical fall detection algorithm, including threshold-based and knowledge-based approaches, and the performance evaluation are introduced in Section 3. The results analysis and discussion are described in detail in Section 4. Finally, we summarize the conclusions of the proposed hierarchical fall detection algorithm in Section 5.

## 2. Related Work

Fall detection systems using wearable sensors are still an open research area that involves a sequence of signal processing, pattern recognition, and machine learning techniques. Generally, the framework of automatic fall detection systems is sensor data collection, a fall detection algorithm, and an emergency alarm [23]. This work is dedicated to developing a fall detection algorithm; therefore, this section especially focuses on reviewing important topics of existing fall detection algorithms.

### 2.1. Fall Detection Algorithm

Sensor placement is a critical issue for the development of wearable sensor-based fall detection algorithms. The most common wearing positions are the waist, wrist, trunk, thigh, back, ankle, foot, neck, and head [23]. The waist and trunk are near the center of mass of the human body and the neck will maintain the head's balance when humans perform ADLs, so the sensor attached to the waist, trunk, or head can detect larger accelerations when the body hits the ground.

Fall detection algorithms of wearable sensor-based fall detection systems are primarily classified into two parts: threshold-based algorithms and machine learning-based fall detection algorithms [12,24]. Some studies have aimed at evaluating the effectiveness of different wearing positions and detection algorithms, as shown in Table 1.

#### 2.1.1. Threshold-Based Fall Detection Algorithms

The threshold-based fall detection methods discriminate between falls and ADLs when the peak values are below or above the threshold [25,26]. The advantages of threshold-based techniques are the low computing complexity and easy implementation in wearable sensors. However, threshold-based techniques are not suitable to detect different types of falls, because thresholds are designed according to the body's experience during falling and fixed thresholds cannot fulfill all kinds of individual habits of activities in daily living.

Bourke et al. [25] used a single threshold, which is an upper fall threshold related to the peak impact force during falling, or a lower fall threshold related to the acceleration before contact with the ground, to detect falls by three accelerometer sensors mounted on the trunk and thigh. The results showed that the 3.52g ( $1g = 9.81 \text{ m/s}^2$ ) upper fall threshold for the trunk had the highest specificity, and suggested that the trunk was the optimal wearing position for a fall sensor. Kangas et al. [27] used a single threshold-based fall detection algorithm with posture detection after the fall with a tri-axial accelerometer attached at the waist, wrist, or head to investigate the placement of a fall detection sensor. The results showed that the head-worn accelerometer provided perfect results, and the authors

suggested that the head was a reasonable wearing position for fall detection. Designing more complex algorithms than the single threshold-based fall detection algorithms, Kangas et al. [28] evaluated different low-complexity fall detection algorithms using accelerometers attached at the waist, wrist, and head. The results ultimately indicated that the effective sensor placements were the waist and head. The sensor at the head level had the highest accuracy, but the usability and user's acceptance should be considered in more detail, i.e., ergonomics. In conclusion, an accelerometer worn on the waist might be an optimal choice for a wearable sensor-based fall detection algorithm.

### 2.1.2. Machine Learning-Based Fall Detection Algorithms

Machine learning-based techniques may overcome the disadvantages of threshold-based techniques. Machine learning is the science of providing the ability of a computer to explore the data construction. Machine learning builds a model using training data to predict or solve the given problem. For fall detection, supervised learning trains the classifier through labeled input data, fall or ADL, during a training period and then identifies individual falls during a classification period [29]. Unsupervised learning algorithms cluster the original data automatically using a clustering algorithm instead of artificially labeling data before training [30]. The accuracy of supervised learning is higher than that of unsupervised learning, but unsupervised learning saves time in the labeling of data. Obviously, high accuracy is needed to fulfill the requirement of a fall detection system. The popular machine learning techniques are naïve Bayesian (NB), support vector machine (SVM), k-nearest neighbors (kNN), radial basis function (RBF), and decision trees. Comparing each machine learning technique in fall detection, Albert et al. [29] performed fall detection by placing a tri-axial accelerometer on the higher part of the subject's trunk, and the results showed that SVM was able to identify a fall with 98% accuracy, and Özdemir et al. [31] claimed that the kNN classifier gave the best accuracy compared to five classifiers, including the least squares method (LSM), SVM, Bayesian decision-making (BDM), dynamic time warping (DTW), and artificial neural network (ANN). Choi et al. [32] detected falls via two experiments, one with single-node analysis and the other with double-node analysis, which used naïve Bayesian classification to classify the data. The results of the single-node analysis and double-node analysis achieved 99.4% and 99.8% accuracy, respectively.

## 3. Materials and Methods

This study used a tri-axial accelerometer that can collect the acceleration generated by body movement, gravitational acceleration, and noise at the same time. The accelerations are generated by a tri-axial accelerometer (OPAL, published by APDM, Portland, USA [33], the range is  $\pm 6g$ ). Since the duration of the fall event is very short, and to avoid losing important data and/or signal, the fixed sampling rate of the accelerometer is set at 128 Hz. The sensor was attached to the waist, around the lower back, to record the body acceleration. The attached position and definition of coordinate system are shown in Figure 1a. We recruited eight healthy volunteers (males,  $22 \pm 1.309$  years old) into this study. We fitted one sensor to the subject's waist, and subjects wore a helmet, waist, knee, and elbow guards as show in Figure 1b while performing the falls on a soft mattress to prevent injuries. The subject performed seven types of falls and ADLs, each repeated two or three times in falls and three or five times in ADLs, depending on physical condition of the subject. Based on the previous work [23], the study summarized the common types of falls and ADLs. The types of falls and ADL events which we adopted are presented in Table 2. There are seven kinds of falls, including 25 falling postures, and seven types of ADLs, including 12 postures, that were performed in the experiment. The falls were executed in four falling directions, such as forward, backward, and right- and left-lateral.

**Table 1.** Literature of automatic fall detection algorithms.

Article (Year)	Detection Algorithm (Methods)	Sensor(s)	Placement	Features Used for Fall Detection	Fall and ADL Types	Results
Kangas et al. (2008) [28]	Threshold-based	Tri-axial accelerometer	Waist Wrist Head	Beginning of the fall ( $SV_{TOT}$ ) Falling velocity Fall impact ( $SV_{TOT}$ , $SV_D$ , $SV_{MaxMin}$ , or $Z_2$ ) Posture after impact	Falls: 9 ADLs: –	$Sn^2$ : 97% (Waist) $Sp^2$ : 100% (Waist)
Dinh et al. (2009) [34]	Machine learning-based (NB, RBF, SVM, C4.5 Ripple down rule learner)	Tri-axial accelerometer Dual-axial gyroscope	Thorax	Acceleration (X, Y and Z axis) Gyroscope (X and Y axis)	Falls: 4 ADLs: 3	Naïve Bayesian Acc <sup>2</sup> : 97.3% Radial Basis Function Acc <sup>2</sup> : 95.8%
Chao et al. (2009) [35]	Threshold-based	Tri-axial accelerometer	Chest Waist	Acceleration magnitude Acceleration cross-product	Falls: 8 ADLs: 13	$Sn^2$ : 98.2% (Chest) $Sp^2$ : 92.4% (Chest) $Sn^2$ : 98.2% (Waist) $Sp^2$ : 89.9% (Waist)
Bourke et al. (2010) [36]	Threshold-based	Tri-axial accelerometer	Waist	Upper fall threshold Lower fall threshold Vertical velocity	Falls: 8 ADLs: 4	Velocity + impact + posture $Sn^2$ : 100% $Sp^2$ : 100% Less than 1 false positive a day
Choi et al. (2011) [32]	Machine learning-based (NB)	SNA <sup>1</sup> : Tri-axial accelerometer, dual-axial gyroscope DNA <sup>1</sup> : Tri-axial accelerometer, one-axial gyroscope	SNA <sup>1</sup> : Chest DNA <sup>1</sup> : Chest, Thigh	SNA <sup>1</sup> : Acceleration (X, Y and Z axis) Gyroscope (X and Y axis) DNA <sup>1</sup> : Acceleration (X, Y and Z axis) Gyroscope (X axis)	SNA <sup>1</sup> : Falls: 4 ADLs: 3 DNA <sup>1</sup> : Falls: 4 ADLs: 4	SNA <sup>1</sup> : Acc <sup>2</sup> : 99.4% DNA <sup>1</sup> : Acc <sup>2</sup> : 99.8%
Rescio et al. (2013) [37]	Machine learning-based (SVM)	Tri-axial accelerometer	Waist	The product between the value of the acceleration peak and the change in the CPO	–	$Sn^2$ : 97.7% $Sp^2$ : 94.8%
Özdemir et al. (2014) [31]	Machine learning-based (kNN, LSM, SVM, Bayesian Decision Making, Dynamic Time Warping, ANN)	Tri-axial accelerometer Tri-axial gyroscope Tri-axial magnetometer	Head, Chest, Waist, Wrist, Thigh, Ankle	Minimum, Maximum Mean, Variance Skewness Kurtosis Autocorrelation Discrete Fourier transform	Falls: 20 ADLs: 16	kNN $Sn^2$ : 100% $Sp^2$ : 99.91%
Huynh et al. (2015) [38]	Threshold-based	Tri-axial accelerometer Tri-axial gyroscope	Chest	Upper fall threshold Lower fall threshold	Falls: 4 ADLs: 6	$Sn^2$ : 96.55% $Sp^2$ : 89.50%
Palmerini et al. (2015) [39]	Threshold-based	Tri-axial accelerometer	Lower back	Continuous wavelet transform coefficients Upper peak value Lower peak value	Falls: 5 ADLs: –	Wavelet $Sn^2$ : 90% $Sp^2$ : 89.7%

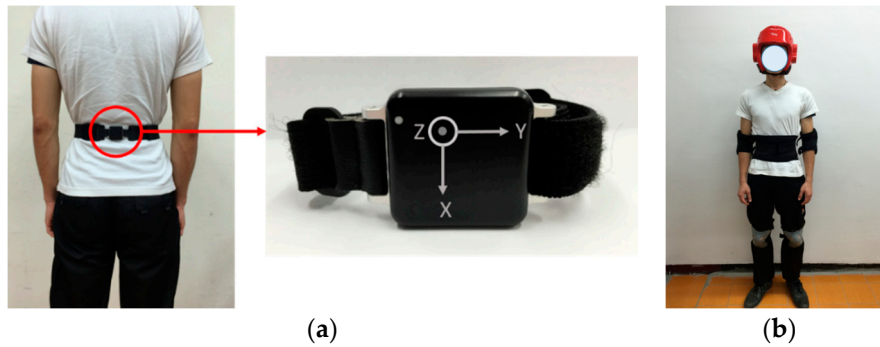
Table 1. Cont.

Article (Year)	Detection Algorithm (Methods)	Sensor(s)	Placement	Features Used for Fall Detection	Fall and ADL Types	Results
He et al. (2016) [40]	Machine learning-based (kNN, NB, Bayes Net, ANN, Decision Tree, Bagging, Ripper)	Tri-axial accelerometer Tri-axial gyroscope	upper trunk	Resultant acceleration ( $\alpha$ ) Resultant angular velocity ( $\omega$ )	Falls: 2 ADLs: 3	kNN ( $k = 3$ ) Sn <sup>2</sup> : 100% Sp <sup>2</sup> : 99.91% Acc <sup>2</sup> : 97.8548%
Chen et al. (2016) [41]	Machine learning-based (SVM)	Tri-axial accelerometer	Waist	Maximum magnitude of the sum vector Rotation angle Slope Acceleration in the xy-plane Standard deviation of the sun vector	Falls: 6 ADLs: 6	Sn <sup>2</sup> : 95.76% Sp <sup>2</sup> : 93.28% Acc <sup>2</sup> : 94.58%
Gibson et al. (2016) [42]	Machine learning-based (ANN, kNN, RBF, Probabilistic Principal Component Analysis, Linear Discriminant Analysis)	Tri-axial accelerometer	Chest	Discrete wavelet transform	Falls: 6 ADLs: 5	Radial Basis Function Sn <sup>2</sup> : 100% Sp <sup>2</sup> : 99.91% Linear Discriminant Analysis Sn <sup>2</sup> : 100% Sp <sup>2</sup> : 99.91%

<sup>1</sup> SNA: single-node analysis; DNA: double-node analysis; <sup>2</sup> Sn: sensitivity; Sp: specificity; and Acc: accuracy.

Table 2. The types and characteristics of falls and ADLs for experiments.

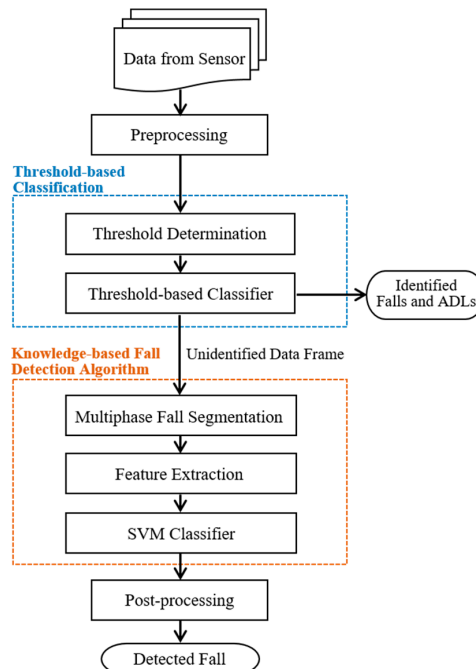
No.	Activities before Fall	Characteristic	No.	Activities of Daily Living	Characteristic
1	Stand	Forward			
2	Stand up	Forward			
3	Sit down	Forward			
4	Stoop	Forward			
5	Walk	Forward			
6	Walk backward	–			
7	Jump	Forward			
1	Stand up	From sit	2	Stand up	From squat
3	Sit down	Normal	4	Sit down	Fast
5	Lie on the bed	Normal	6	Lie on the bed	Fast
7	Go up stairs	Normal	8	Go down stairs	Normal
9	Walk	Normal	10	Walk	Fast
11	Jump	On the ground	12	Jump	On the bed



**Figure 1.** (a) Wearing position and the axial direction of sensor; (b) Schematic view of the participant wearing a helmet, waist, knee, and elbow guards.

### 3.1. Hierarchical Fall Detection Algorithm

The functional diagram of the proposed hierarchical fall detection algorithm is shown in Figure 2. There are four stages in the hierarchical fall detection algorithm, including preprocessing, threshold-based classification, the knowledge-based fall detection algorithm, and post-processing. In preprocessing, we propose a windowing approach to segment each record with the same window size. Firstly,  $Norm_{xyz}$  is defined as the Euclidean norm of tri-axial accelerations as calculated by Equation (1), where  $a_x$ ,  $a_y$ , and  $a_z$  are the acceleration ( $g$ ) in the  $x$ -,  $y$ -, and  $z$ -axes, respectively. Then the time index corresponding to a maximum of  $Norm_{xyz}$  is identified as the critical point. Finally, the data frame is determined by taking 1.5 s ( $128 \text{ Hz} \times 1.5 \text{ s} = 192$  samples) before the critical point and 2.5 s ( $128 \text{ Hz} \times 2.5 \text{ s} = 320$  samples) after the critical point, corresponding to a time window of 513 samples ( $192 + \text{time index of the critical point} + 320$ ).



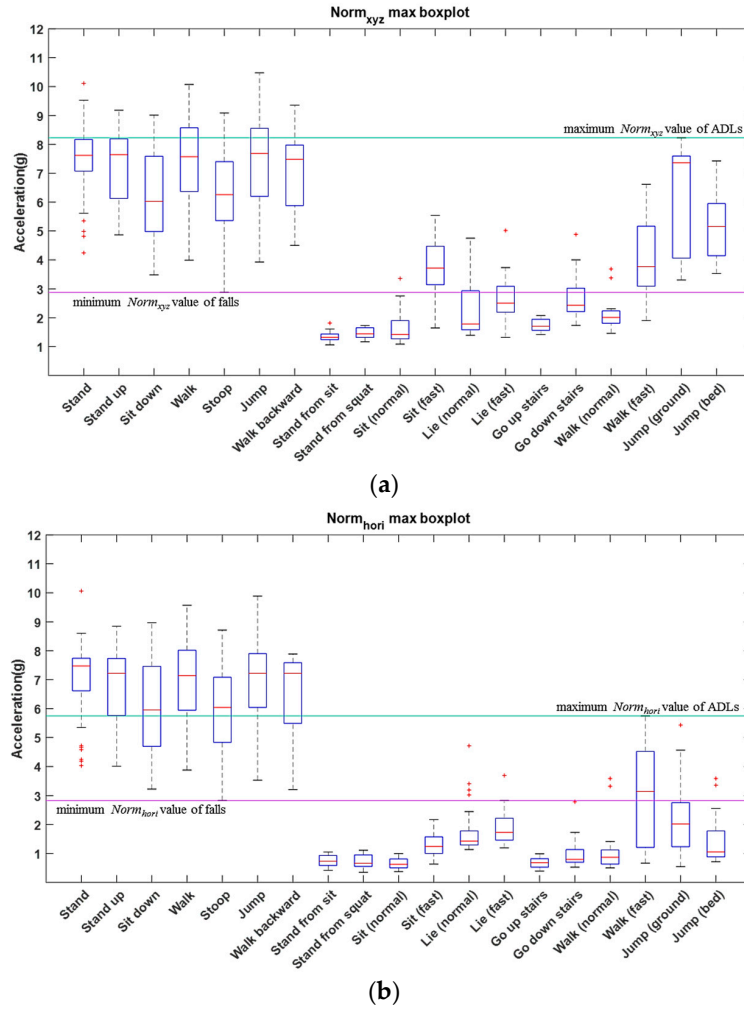
**Figure 2.** The functional diagram of the proposed fall detection algorithm.

In this work, the threshold-based classification is utilized to identify the absolute fall and ADL. Two common features can be extracted from each data frame involving  $Norm_{xyz}$  and  $Norm_{hori}$ .  $Norm_{xyz}$  is used to describe the spatial variation of acceleration during the falling interval.  $Norm_{hori}$  is defined

as the Euclidean norm of acceleration in the horizontal plane and can be calculated by Equation (2), which is used to describe the change of velocity in the horizontal plane of the body. The threshold is determined by observing the distribution of the maximum  $Norm_{xyz}$  and  $Norm_{hori}$  of the data frames. The  $Norm_{xyz}$  and  $Norm_{hori}$  distribution of falls and ADLs are presented in Figure 3.

$$Norm_{xyz} = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (1)$$

$$Norm_{hori} = \sqrt{a_y^2 + a_z^2} \quad (2)$$



**Figure 3.** The illustration of the  $Norm_{xyz}$  and  $Norm_{hori}$  distribution of falls and ADLs. The left seven activities are falls and the right 12 activities are ADLs. The green line is determined by the maximum  $Norm_{xyz}$  value of ADLs. The purple line is determined by the minimum  $Norm_{xyz}$  value of falls. (a) The boxplot of maximum  $Norm_{xyz}$  values for falls and ADLs; (b) The boxplot of maximum  $Norm_{hori}$  values for falls and ADLs.

Given that a training set of a data frame  $F_{train} = \{(f_i^{train}) | i = 1, 2, \dots, N_{train}\}$ , where  $N_{train}$  is the total number of the set  $F_{train}$ , and a testing set of data frame  $F_{test} = \{(f_i^{test}) | i = 1, 2, \dots, N_{test}\}$ , where  $N_{test}$  is the total number of the set  $F_{test}$ , there are two sets of maximum  $Norm_{xyz}$  and  $Norm_{hori}$  corresponding to  $F_{train}$  that can be presented as  $V_{train} = \{v_i^{train} | i = 1, 2, \dots, N_{train}\}$  and  $W_{train} = \{w_i^{train} | i = 1, 2, \dots, N_{train}\}$ , respectively. According to the distribution of  $V_{train}$  and  $W_{train}$ , two thresholds, including  $T_{fall}^{Norm_{xyz}}$  and  $T_{fall}^{Norm_{hori}}$ , for the absolute fall detection can be determined,



where  $T_{fall}^{Norm_{xyz}}$  and  $T_{fall}^{Norm_{hori}}$  are the maximum  $Norm_{xyz}$  and  $Norm_{hori}$  values of ADLs, respectively. Additionally, two thresholds, including  $T_{ADL}^{Norm_{xyz}}$  and  $T_{ADL}^{Norm_{hori}}$ , for the absolute ADL identification can be determined, where  $T_{ADL}^{Norm_{xyz}}$  and  $T_{ADL}^{Norm_{hori}}$  are the minimum  $Norm_{xyz}$  and  $Norm_{hori}$  values of falls, respectively. Similarly, two sets of maximum  $Norm_{xyz}$  and  $Norm_{hori}$  values corresponding to  $F_{test}$  can be presented as  $V_{test} = \{v_i^{test} | i = 1, 2, \dots, N_{test}\}$  and  $W_{test} = \{w_i^{test} | i = 1, 2, \dots, N_{test}\}$ , respectively. Finally, the threshold-based classifier applied to the testing set can be defined as follows:

$$TC(v_i^{test}, w_i^{test}) = \begin{cases} \text{Fall, if } v_i^{test} > T_{fall}^{Norm_{xyz}} \text{ and } w_i^{test} > T_{fall}^{Norm_{hori}} \\ \text{ADL, if } v_i^{test} < T_{ADL}^{Norm_{xyz}} \text{ and } w_i^{test} < T_{ADL}^{Norm_{hori}} \\ \text{Unidentified data frame, others} \end{cases} \quad (3)$$

Fall events can be represented by semantics and rules. A fall is divided into three phases (free fall, impact, and rest phases) based on the attributes of the fall. The free fall phase consists of the sudden movement of the body toward the ground, before hitting the ground, which cannot be rescued by protective strategies. The duration of the free fall phase depends on the subject, and the signal pattern when  $Norm_{xyz}$  is closer to 0g has been studied in various durations, including 120–200 ms [22], 300–500 ms [21], and 400–800 ms [43]. The impact phase is determined by the person hitting the ground and has a signal pattern with a sharp pulsation, usually less than one second. In the rest phase, the person remains motionless in a prone or supine position and usually has a mitigating signal pattern. There is potential for the multiphase fall model to be improved, involving important aspects of the fall event. The knowledge-based fall detection algorithm using a multiphase fall model is proposed to provide the system with advanced functionality to deal with unidentified data frames.

According to observations and the statistical analysis of data frames, the multiphase fall segmentation for the multiphase fall model is presented as follows. To determine the duration of impact phase, two situations are identified based on the severity of falls. One situation is the maximum of  $Norm_{xyz}$  larger than 6g, and the other is the maximum of  $Norm_{xyz}$  less than 6g. The impact phase of the first situation is determined by taking 10 samples before and after the critical point, and a total of 21 samples (10 + critical point + 10) while the maximum of  $Norm_{xyz}$  is larger than 6g. The impact phase of another situation is determined by taking 10 samples before, and 20 samples after, the critical point, for a total of 31 samples (10 + critical point + 20) while the maximum of  $Norm_{xyz}$  is less than 6g. Then, the rest phase is determined by taking the following samples after the defined impact phase within the data frame. Finally, the free fall phase is determined by taking 32 samples before the impact phase. The proposed multiphase fall segmentation is sufficient for the knowledge-based fall detection algorithm to identify multiple situations, so that the necessary information can be obtained. The illustration of the multiphase fall segmentation for two situations is shown in Figure 4, and there are three phases, in turn: the free fall phase, impact phase, and rest phase.

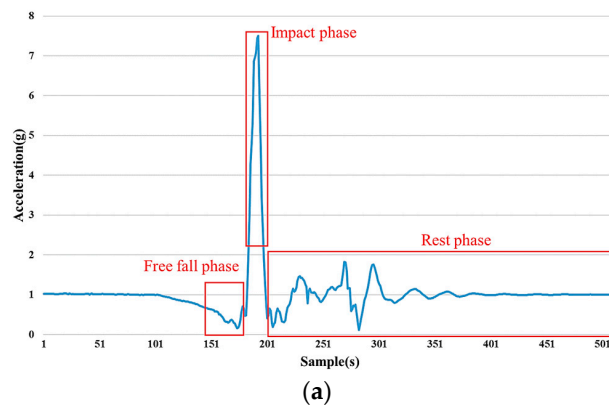
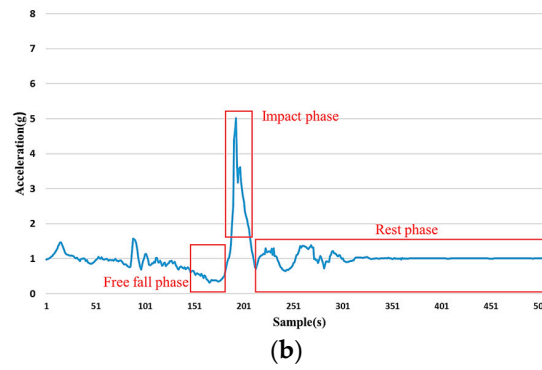


Figure 4. Cont.



**Figure 4.** The illustration of the multiphase fall segmentation for two situations. There are three phases, in turn: free fall phase, impact phase, and rest phase. (a) The situation of the maximum of  $Norm_{xyz}$  is larger than  $6g$ , taking the data frame of the standing forward fall, for example; (b) The situation of the maximum of  $Norm_{xyz}$  is less than  $6g$ , taking the data frame of the back-walking backward fall for example.

Suitable features extracted from signal make for accurate results of the classification model. Based on the previous review study [23], the commonly extracted features were summarized. Time-domain statistical features are selected in this work, including mean, standard deviation, variance, maximum, minimum, range, kurtosis, skewness, and correlation coefficient [44]. The nine features are extracted from each axis including the Euclidean norm of tri-axial acceleration, Euclidean norm of acceleration on the coronal plane, and the Euclidean norm of acceleration on horizontal plane. The list of 54 features used in this work is summarized in Table 3, where  $m$  is the number of total samples in a data frame or phase (free fall, impact, and rest phase). All of the features are employed in all of the data frames or phases for training and testing processes.

**Table 3.** The feature vector for the knowledge-based fall detection algorithm.

Feature Vector, $F = (f_1, f_2, \dots, f_{54}) \in \mathbf{R}^{54}$	Feature Description
$f_1 \sim f_3$	mean $a_x(i)$ ; mean $a_y(i)$ ; mean $a_z(i)$ , where $i = 1, \dots, m^1$
$f_4 \sim f_6$	mean $a_{norm}(i)^2$ ; mean $a_{verti}(i)^3$ ; mean $a_{hori}(i)^4$ , where $i = 1, \dots, m^1$
$f_7 \sim f_9$	std $a_x(i)$ ; std $a_y(i)$ ; std $a_z(i)$ , where $i = 1, \dots, m^1$
$f_{10} \sim f_{12}$	std $a_{norm}(i)^2$ ; std $a_{verti}(i)^3$ ; std $a_{hori}(i)^4$ , where $i = 1, \dots, m^1$
$f_{13} \sim f_{15}$	var $a_x(i)$ ; var $a_y(i)$ ; var $a_z(i)$ , where $i = 1, \dots, m^1$
$f_{16} \sim f_{18}$	var $a_{norm}(i)^2$ ; var $a_{verti}(i)^3$ ; var $a_{hori}(i)^4$ , where $i = 1, \dots, m^1$
$f_{19} \sim f_{21}$	max $a_x(i)$ ; max $a_y(i)$ ; max $a_z(i)$ , where $i = 1, \dots, m^1$
$f_{22} \sim f_{24}$	max $a_{norm}(i)^2$ ; max $a_{verti}(i)^3$ ; max $a_{hori}(i)^4$ , where $i = 1, \dots, m^1$
$f_{25} \sim f_{27}$	min $a_x(i)$ ; min $a_y(i)$ ; min $a_z(i)$ , where $i = 1, \dots, m^1$
$f_{28} \sim f_{30}$	min $a_{norm}(i)^2$ ; min $a_{verti}(i)^3$ ; min $a_{hori}(i)^4$ , where $i = 1, \dots, m^1$
$f_{31} \sim f_{33}$	range $a_x(i)$ ; range $a_y(i)$ ; range $a_z(i)$ , where $i = 1, \dots, m^1$
$f_{34} \sim f_{36}$	range $a_{norm}(i)^2$ ; range $a_{verti}(i)^3$ ; range $a_{hori}(i)^4$ , where $i = 1, \dots, m^1$
$f_{37} \sim f_{39}$	kurtosis $a_x(i)$ ; kurtosis $a_y(i)$ ; kurtosis $a_z(i)$ , where $i = 1, \dots, m^1$
$f_{40} \sim f_{42}$	kurtosis $a_{norm}(i)^2$ ; kurtosis $a_{verti}(i)^3$ ; kurtosis $a_{hori}(i)^4$ , where $i = 1, \dots, m^1$
$f_{43} \sim f_{45}$	skewness $a_x(i)$ ; skewness $a_y(i)$ ; skewness $a_z(i)$ , where $i = 1, \dots, m^1$
$f_{46} \sim f_{48}$	skewness $a_{norm}(i)^2$ ; skewness $a_{verti}(i)^3$ ; skewness $a_{hori}(i)^4$ , where $i = 1, \dots, m^1$
$f_{49}$	Correlation coefficient between $a_x$ and $a_y$
$f_{50}$	Correlation coefficient between $a_x$ and $a_z$
$f_{51}$	Correlation coefficient between $a_y$ and $a_z$
$f_{52}$	Correlation coefficient between $a_{norm}^2$ and $a_{verti}^3$
$f_{53}$	Correlation coefficient between $a_{norm}^2$ and $a_{hori}^4$
$f_{54}$	Correlation coefficient between $a_{verti}^3$ and $a_{hori}^4$

<sup>1</sup>  $m$ : Determined size of data frame (or phase); <sup>2</sup>  $a_{norm}$ : Euclidean norm of tri-axial acceleration; <sup>3</sup>  $a_{verti}$ : Euclidean norm of acceleration on coronal plane; <sup>4</sup>  $a_{hori}$ : Euclidean norm of acceleration on horizontal plane.

In the SVM classifier, we use the multi-class SVM technique to train the multiphase fall classifier to classify the fall event into the three distinct phases. SVM is a machine-learning technique originally for binary classification. Training data of the separate categories are divided by a hyperplane and margins that are as wide as possible. A training algorithm trains a classification model, which can classify testing data into one category or the other. Further, binary classification can extend to multi-class classifiers with more than two categories. There are two approaching types for multi-class SVM. One is combining several binary classifiers (one-versus-one technique), and the other is considering all training data in one optimization formulation (one-versus-all technique) [45]. The one-versus-one technique trains  $x(x - 1)/2$  binary classifiers for a  $x$ -class SVM [46]. The outcome's index of the  $x$ -class SVM has one positive class and the others have negative classes; that positive class is the predicted result of the  $x$ -class SVM. The multi-class SVM classifier is applied with the one-versus-one technique and the linear kernel function in this work.

The unidentified data frame in the testing data is divided into three phases following the multiphase fall segmentation. The trained multiphase fall classifier classifies the unidentified phase into a series of phases by feature extraction. In post-processing, we define the temporal order of the fall, listed as free fall, impact, and rest, and check the temporal order from the SVM classifier outcome whether the temporal order is of a fall or not. The fall phase detection and post-processing are defined with the following description: Suppose the acceleration frame  $A = \{a_1, a_2, a_3\}$ , where  $a_i = (s_i, t_i)$  is the tuple with the fall detection phase  $s_i$  and timestamp  $t_i$ ; then  $1 \leq i \leq 3$ . For each fall detection phase  $s_i \in SF$ , and  $SF$  is the set of distinct semantic fall phases  $SF = \{(sf_j) | 1 \leq j \leq k\}$ , where  $k$  is the number of the defined semantic fall phase. In this work, three semantic fall phases are defined ( $k = 3$ )  $\{\text{"FreeFall"}, \text{"Impact"}, \text{"Rest"}\}$ . Finally, we check the tuple of  $a_i$  in  $A$ . The results of post-processing are detected falls when the series classifying free fall, impact, and rest are in order.

### 3.2. Performance Evaluation Criteria

The result of fall detection is either fall or ADL, which belongs to the binary classification. In the performance evaluation criteria of the binary classification test, a positive condition means that the subject falls, and a negative condition means that the subject performed an ADL. Based on the detected result, a fall alarm belongs to the positive test outcome, and an ADL is the result of a negative test outcome. There are four situations in fall detection, including true positive (TP), false positive (FP), true negative (TN), and false negative (FN) [47]. The fall detection system should avoid getting FP and FN results. Sensitivity, specificity, positive predictive value (PPV), negative predictive value (NPV), and accuracy are the common performance evaluation criteria for the binary classification test, and many studies adopted those criteria to show the results of fall detection [24,47,48]. Sensitivity (or recall) is the capability of detecting falls, and the PPV (or precision) is the quality of detecting exact falls. Sensitivity, specificity, precision, and accuracy show more effective evaluation of human activity classification for the imbalanced dataset [49,50]. The sensitivity, specificity, and precision are computed by Equations (4)–(6), respectively. Accuracy is the proportion of the truth test outcome in the total results, whose calculation follows Equation (7). The higher values of sensitivity, specificity, precision, and accuracy, the higher the performance the system provides.

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (4)$$

$$\text{Specificity} = \frac{TN}{FP + TN} \quad (5)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (6)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (7)$$

#### 4. Results and Discussion

The results of the performance evaluation were obtained by the  $k$ -fold cross-validation method and repeated over five rounds, meaning we chose  $k = 5$  for the  $k$ -fold cross-validation method. The  $k$ -fold cross-validation method randomly partitions the collected data into  $k$  portions and utilizes  $k - 1$  portions as training data and one portion as testing data, then repeating it  $k$  times to evaluate the results. The total number of falls and ADLs were 475 and 364, respectively. Further, the total number of each posture, including stand, stand up, sit down, walk, stoop, jump, walk backward, stand from sit, stand from squat, sit (normal), sit (fast), lie (normal), lie (fast), go upstairs, go downstairs, walk (normal), walk (fast), jump (ground) and jump (bed), was 76, 76, 76, 76, 76, 76, 19, 30, 30, 30, 30, 29, 30, 37, 29, 30, 30, 29, and 30, respectively. The total data and false predictions of falls and ADLs in each round of the five-fold cross-validation are presented in Appendix A in detail. The total testing data of falls and ADLs in each round are shown in Table A1. The testing data of false predictions for falls and ADLs are represented in Tables A2 and A3.

The threshold-based classification was utilized to identify the absolute falls or ADLs and to decrease the computing complexity in the knowledge-based fall detection algorithm. The identified average percentage of falls and ADLs by the initial threshold-based classification was 12.92% and 27.84%, respectively. Thus, there was an unidentified average percentage of 59.24% in the input data of the knowledge-based fall detection algorithm. In comparison with the proposed knowledge-based fall detection algorithm, the knowledge-based approach without the multiphase fall model, called the machine learning-based fall detection algorithm, was also applied in this work. The average accuracy and standard deviations of each five-fold cross-validation over five rounds for the knowledge-based and machine learning-based fall detection algorithms are presented in Tables 4 and 5, respectively. The performances of the sensitivity, specificity, precision, and accuracy using knowledge-based algorithms were higher than the performances using machine learning-based methods in each round.

**Table 4.** The overall performance of five-fold cross-validation over five rounds for the knowledge-based fall detection algorithm. Std: standard deviation.

Knowledge-Based Fall Detection Algorithm						
Round	1	2	3	4	5	Mean (Std)
Sensitivity (%)	100 (0)	99.79 (0.47)	99.58 (0.57)	99.79 (0.48)	99.79 (0.47)	99.79 (0.43)
Specificity (%)	98.63 (1.36)	98.62 (1.00)	98.90 (0.61)	98.62 (0.98)	98.91 (0.61)	98.74 (0.88)
Precision (%)	98.97 (1.03)	98.96 (0.73)	99.16 (0.47)	98.97 (0.72)	99.17 (0.47)	99.05 (0.66)
Accuracy (%)	99.41 (0.59)	99.29 (0.50)	99.29 (0.26)	99.28 (0.27)	99.41 (0.42)	99.33 (0.40)

**Table 5.** The overall performance of five-fold cross-validation over five rounds for the machine learning-based fall detection algorithm.

Machine Learning-Based Fall Detection Algorithm						
Round	1	2	3	4	5	Mean (Std)
Sensitivity (%)	99.58 (0.58)	99.16 (1.37)	99.36 (0.94)	99.15 (0.89)	98.95 (0.74)	99.24 (0.89)
Specificity (%)	98.63 (1.68)	98.63 (1.66)	98.63 (2.37)	97.80 (2.50)	98.37 (1.48)	98.41 (1.84)
Precision (%)	98.97 (1.26)	98.95 (1.29)	98.99 (1.73)	98.35 (1.88)	98.76 (1.10)	98.81 (1.38)
Accuracy (%)	99.17 (0.90)	98.93 (1.47)	99.05 (0.90)	98.56 (1.02)	98.69 (0.49)	98.88 (0.95)

The standard deviations of the performances using the proposed algorithm were lower than those using machine learning-based methods in each round. The overall performances of the sensitivity, specificity, precision, and accuracy using the knowledge-based algorithm were 99.79%, 98.74%, 99.05%, and 99.33%, respectively. Considering the same round, the highest sensitivity and precision were 100% and 98.97% in the first round using the knowledge-based fall detection algorithm, and 99.58% and 98.97% using the machine learning-based fall detection algorithm. Precision is the quality of detecting

exact falls and the highest performance of the proposed algorithm was 99.17%. Compared to using the machine learning-based algorithm, the improvement in the overall performances of sensitivity, specificity, precision, and accuracy using the knowledge-based algorithm were 0.55%, 0.33%, 0.24% and 0.45%, respectively.

The confusion matrices of the knowledge-based and machine learning-based algorithms for each type of fall and ADL are shown in Tables 6 and 7, respectively. Two confusion matrices present the numbers classified by the two algorithms from the total testing data with five rounds. The results show that the false-negative rate using the knowledge-based algorithm in stoop was 1.32%, and the false-negative rates using the machine learning-based algorithm in sit down, stoop, jump, and walk backward were 0.79%, 2.1%, 0.53% and 5.26%, respectively. Obviously, the false predicted types, as well as the numbers of false predicted falls, using the proposed algorithm were less than those when using the machine learning-based algorithm. Similarly, the types of false classified ADLs using the proposed algorithm were less than those using the machine learning-based algorithm. However, the proposed algorithm was still too weak to deal with the ADLs in the lie on the bed activity at normal and fast speeds. This is because the pattern of falls and lying are inherently similar, and the experiment assumed that the faller lost consciousness. Furthermore, the soft mattress set on the ground in the experimental environment caused some subjects to perform the lie on the bed activity at a normal or fast speed, which led to a larger impact signal on the soft mattress and, thus, a similar pattern to a fall. Such falsely predicted ADLs can be improved significantly through the additional information from the context. For example, most lying activities exist when the subject lies on the bed in the bedroom. Additional context information, such as object interaction and location, can provide a fall detection system with the ability to exclude normal ADLs.

Becker et al. [22] proposed a multiphase fall model based on real-world fall recordings. The multiphase fall model contains pre-fall, falling, impact, resting, and recovery phases. We adopted falling, impact, and resting phases as core phases of the fall model in this study, while the experiment assumed that the faller lost consciousness. Fortino et al. [51] proposed a two-stage fall detection system. The first stage is a threshold-based trigger and the second stage utilizes posture recognition of the recovery phase to classify the different levels of alarm severity. For example, the recovery phase may be as important as the impact phase and can judge the severity of a fall event from a clinical perspective. The pre-fall phase can provide more information for the caregiver or healthcare professional when the faller has forgotten or misremembered the circumstances about the fall event. Kangas et al. [52] presented that the impact signal of a fall in an emulation experiment is lower than in real-world falls. The smaller impact signal of a fall may result in ADLs and falls having similar impact signals, which poses a challenging task for the fall detection algorithm. The experimental results have shown that the proposed algorithm can efficiently deal with such challenging tasks and it improves the performance of the machine learning-based algorithm. Furthermore, the meaningful segmentation based on the multiphase fall model can tackle the issue of ambiguity, while some short-duration ADLs have similar attributes to falls, such as sitting down, standing up, and jumping.

The standard deviation of the performances can indicate the stability of the algorithm and/or the system. Compared to a previous study [31], the standard deviation of the performances had a higher score in this work. The standard deviation of the performances was larger than the gain in specificity and precision in this work. Perhaps the greatest reason is that the number of data frames for the testing and training processes was smaller. A larger number of data frames that can train a robust model can provide a more correct classification, thus achieving a more stable algorithm. In future work, we plan to include more emulation experiments for validation, and to consider more phases of a fall to develop a robust fall detection algorithm. Such robust fall detection algorithms have great potential to be successfully applied in various applications involving fall prevention and long-term fall event recording. Further, the real-time automatic fall detection system based on the proposed algorithm will be implemented in the real world.



## 5. Conclusions

The problems generated by fall accidents are important issues in an aging and aged society. Rescuing the victims in time can not only reduce injuries caused by falls but can also increase the confidence in performing ADLs in the elderly. The automatic fall detection system has the opportunity to provide real-time emergency alarms and services for improving safety and health-related quality of life. In this work, we presented a novel hierarchical fall detection algorithm to detect fall events by using a multiphase fall model. Two sensing domains were analyzed, which are directly linked to the sequence of ADLs and fall events. The evaluation results showed that the proposed hierarchical fall detection algorithm can efficiently deal with the issues of the individual differences, such as adaptability and flexibility. Different from previous studies with respect to the machine learning-based method for fall detection, the proposed fall detection algorithm classified the phases—free fall, impact, and rest on the ground—during falling. The overall performances of sensitivity, specificity, precision, and accuracy using the knowledge-based algorithm were 99.79%, 98.74%, 99.05%, and 99.33%, respectively. The improvement of the overall performances of sensitivity, specificity, precision, and accuracy using the knowledge-based algorithm were 0.55%, 0.33%, 0.24%, and 0.45%, respectively, compared to using the machine learning-based algorithm. In future work, we plan to refine the fall detection system for continuous monitoring, and evaluate it in an out-of-lab environment.

**Acknowledgments:** This work was supported by the Ministry of Science and Technology under grant number MOST 105-2221-E-010-008. The authors would like to thank the volunteers who participated in the experiments for their efforts and time.

**Author Contributions:** Chia-Yeh Hsieh, Kai-Chun Liu, and Chih-Ning Huang coordinated and performed the experiments, analyzed the data, and prepared the initial draft of the manuscript. Woei-Chyn Chu gave suggestions and supervised this work. Chia-Tai Chan proposed the research topic, supervised this work, and contributed significantly to the writing and organization of the manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

Table A1. The total testing data of falls and ADLs in each round of five-fold cross-validation rounds.

Round	Stand	Stand up	Sit down	Walk	Stoop	Jump	Walk Backward	Stand from Sit	Stand from Squat	Sit (Normal)	Sit (Fast)	Lie (Normal)	Lie (Fast)	Go up Stairs	Go down Stairs	Walk (Normal)	Walk (Fast)	Jump (Ground)	Jump (Bed)
1	15	15	16	16	15	15	3	6	6	6	6	6	6	7	6	6	6	6	6
	16	15	15	15	15	15	4	6	6	6	6	6	6	8	6	6	6	6	6
	15	15	15	15	16	16	4	6	6	6	6	6	6	7	6	6	6	5	6
	15	16	15	15	15	15	4	6	6	6	6	6	6	7	6	6	6	6	6
	15	15	15	15	15	15	4	6	6	6	6	5	6	8	5	6	6	6	6
2	15	15	15	16	15	15	4	6	6	6	6	5	6	7	5	6	6	6	6
	16	15	15	15	15	16	4	6	6	6	6	6	6	7	6	6	6	5	6
	15	16	16	15	15	15	4	6	6	6	6	6	6	8	6	6	6	6	6
	15	15	15	15	16	15	4	6	6	6	6	6	6	8	6	6	6	6	6
	15	15	15	15	15	15	3	6	6	6	6	6	6	7	6	6	6	6	6
3	15	15	15	15	15	15	3	6	6	6	6	6	6	7	6	6	6	5	6
	15	15	15	16	15	15	4	6	6	6	6	6	6	7	6	6	6	6	6
	15	15	15	15	15	16	4	6	6	6	6	6	6	7	6	6	6	6	6
	15	16	15	15	16	15	4	6	6	6	6	5	6	8	6	6	6	6	6
	16	15	16	15	15	15	4	6	6	6	6	6	6	8	5	6	6	6	6
4	16	15	16	15	15	15	4	6	6	6	6	6	6	8	6	6	6	6	6
	15	16	15	16	15	15	4	6	6	6	6	6	6	7	6	6	6	6	6
	15	15	15	15	15	15	3	6	6	6	6	5	6	8	5	6	6	6	6
	15	15	15	15	15	15	4	6	6	6	6	6	6	7	6	6	6	6	6
	15	15	15	15	16	16	4	6	6	6	6	6	6	7	6	6	6	5	6
5	16	15	15	16	15	15	4	6	6	6	6	6	6	7	6	6	6	6	6
	15	15	16	15	16	15	4	6	6	6	6	6	6	7	5	6	6	6	6
	15	15	15	15	15	15	4	6	6	6	6	5	6	7	6	6	6	5	6
	15	16	15	15	15	16	4	6	6	6	6	6	6	8	6	6	6	6	6
	15	15	15	15	15	15	3	6	6	6	6	6	6	8	6	6	6	6	6
total	380	380	380	380	380	380	95	150	150	150	150	145	150	185	145	150	150	145	150



**Table A2.** The total testing data of false predictions for falls and ADLs using the knowledge-based algorithm.

Round	Stand	Stand up	Sit down	Walk	Stoop	Jump	Walk Backward	Stand from Sit	Stand from Squat	Sit (Normal)	Sit (Fast)	Lie (Normal)	Lie (Fast)	Go up Stairs	Go down Stairs	Walk (Normal)	Walk (Fast)	Jump (Ground)	Jump (Bed)
1												1	1						
												1	1						
2					1							2	1						
													1						
3							1						1						
							1						1						
4												1	1						
													2						
5							1						1						
													1						
													1						
total	0	0	0	0	5	0	0	0	0	0	0	8	15	0	0	0	0	0	0

**Table A3.** The total testing data of false predictions for falls and ADLs using the machine learning-based algorithm.

Round	Stand	Stand up	Sit down	Walk	Stoop	Jump	Walk Backward	Stand from Sit	Stand from Squat	Sit (Normal)	Sit (Fast)	Lie (Normal)	Lie (Fast)	Go up Stairs	Go down Stairs	Walk (Normal)	Walk (Fast)	Jump (Ground)	Jump (Bed)
1			1									2	1						
2							1					1				1			
3			1		2														
4						1						2	2						
5					1	1						1	1						
total	0	0	3	0	8	2	5	0	0	0	4	8	11	0	0	1	4	1	

## References

- Bloom, D.E.; Boersch-Supan, A.; McGee, P.; Seike, A. Population aging: Facts, challenges, and responses. *Benefits Compens. Int.* **2011**, *41*, 22.
- Stevens, J.A.; Corso, P.S.; Finkelstein, E.A.; Miller, T.R. The costs of fatal and non-fatal falls among older adults. *Inj. Prev.* **2006**, *12*, 290–295. [[CrossRef](#)] [[PubMed](#)]
- Kannus, P.; Parkkari, J.; Koskinen, S.; Niemi, S.; Palvanen, M.; Jarvinen, M.; Vuori, I. Fall-induced injuries and deaths among older adults. *JAMA* **1999**, *281*, 1895–1899. [[CrossRef](#)] [[PubMed](#)]
- Gurley, R.J.; Lum, N.; Sande, M.; Lo, B.; Katz, M.H. Persons found in their homes helpless or dead. *N. Engl. J. Med.* **1996**, *334*, 1710–1716. [[CrossRef](#)] [[PubMed](#)]
- Wild, D.; Nayak, U.; Isaacs, B. How dangerous are falls in old people at home? *Br. Med. J. (Clin. Res. Ed.)* **1981**, *282*, 266–268. [[CrossRef](#)]
- Hartholt, K.A.; van Beeck, E.F.; Polinder, S.; van der Velde, N.; van Lieshout, E.M.; Panneman, M.J.; van der Cammen, T.J.; Patka, P. Societal consequences of falls in the older population: Injuries, healthcare costs, and long-term reduced quality of life. *J. Trauma Acute Care Surg.* **2011**, *71*, 748–753. [[CrossRef](#)] [[PubMed](#)]
- Shumway-Cook, A.; Ciol, M.A.; Gruber, W.; Robinson, C. Incidence of and risk factors for falls following hip fracture in community-dwelling older adults. *Phys. Ther.* **2005**, *85*, 648–655. [[PubMed](#)]
- Delahoz, Y.S.; Labrador, M.A. Survey on fall detection and fall prevention using wearable and external sensors. *Sensors* **2014**, *14*, 19806–19842. [[CrossRef](#)] [[PubMed](#)]
- Tinetti, M.E. Preventing falls in elderly persons. *N. Engl. J. Med.* **2003**, *348*, 42–49. [[CrossRef](#)] [[PubMed](#)]
- Prevention, O.F.; Panel, O.S. Guideline for the prevention of falls in older persons. *J. Am. Geriatr. Soc.* **2001**, *49*, 664–672.
- Zecevic, A.A.; Salmoni, A.W.; Speechley, M.; Vandervoort, A.A. Defining a fall and reasons for falling: Comparisons among the views of seniors, health care providers, and the research literature. *Gerontol.* **2006**, *46*, 367–376. [[CrossRef](#)]
- Igual, R.; Medrano, C.; Plaza, I. Challenges, issues and trends in fall detection systems. *Biomed. Eng. Online* **2013**, *12*, 1–66. [[CrossRef](#)] [[PubMed](#)]
- Zigel, Y.; Litvak, D.; Gannot, I. A method for automatic fall detection of elderly people using floor vibrations and sound—Proof of concept on human mimicking doll falls. *IEEE Trans. Biomed. Eng.* **2009**, *56*, 2858–2867. [[CrossRef](#)] [[PubMed](#)]
- Stone, E.E.; Skubic, M. Fall detection in homes of older adults using the microsoft kinect. *IEEE J. Biomed. Health Inform.* **2015**, *19*, 290–301. [[CrossRef](#)] [[PubMed](#)]
- Daher, M.; Diab, A.; El Najjar, M.E.B.; Khalil, M.; Charpillat, F.; Daher, M.; Diab, A.; El Najjar, M.E.B.; Khalil, M.; Charpillat, F. Automatic fall detection system using sensing floors. *Int. J. Comput. Inform. Sci.* **2016**, *12*, 75.
- Lin, C.-S.; Hsu, H.C.; Lay, Y.-L.; Chiu, C.-C.; Chao, C.-S. Wearable device for real-time monitoring of human falls. *Measurement* **2007**, *40*, 831–840. [[CrossRef](#)]
- Chen, J.; Kwong, K.; Chang, D.; Luk, J.; Bajcsy, R. Wearable sensors for reliable fall detection. In Proceedings of the 27th Annual International Conference of the Engineering in Medicine and Biology Society, Shanghai, China, 17–18 January 2005; pp. 3551–3554.
- Bourke, A.K.; Lyons, G.M. A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. *Med. Eng. Phys.* **2008**, *30*, 84–90. [[CrossRef](#)] [[PubMed](#)]
- Lamb, S.E.; Jørstad-Stein, E.C.; Hauer, K.; Becker, C. Development of a common outcome data set for fall injury prevention trials: The prevention of falls network europe consensus. *J. Am. Geriatr. Soc.* **2005**, *53*, 1618–1622. [[CrossRef](#)] [[PubMed](#)]
- Shinmoto Torres, R.L.; Visvanathan, R.; Hoskins, S.; van den Hengel, A.; Ranasinghe, D.C. Effectiveness of a batteryless and wireless wearable sensor system for identifying bed and chair exits in healthy older people. *Sensors* **2016**, *16*, 546. [[CrossRef](#)] [[PubMed](#)]
- Noury, N.; Rumeau, P.; Bourke, A.; ÓLaghin, G.; Lundy, J. A proposal for the classification and evaluation of fall detectors. *IRBM* **2008**, *29*, 340–349. [[CrossRef](#)]

22. Becker, C.; Schwickert, L.; Mellone, S.; Bagala, F.; Chiari, L.; Helbostad, J.L.; Zijlstra, W.; Aminian, K.; Bourke, A.; Todd, C.; et al. Proposal for a multiphase fall model based on real-world fall recordings with body-fixed sensors. *Zeitschrift für Gerontologie und Geriatrie* **2012**, *45*, 707–715. [[CrossRef](#)] [[PubMed](#)]
23. Pannurat, N.; Thiemjarus, S.; Nantajeewarawat, E. Automatic fall monitoring: A review. *Sensors* **2014**, *14*, 12900–12936. [[CrossRef](#)] [[PubMed](#)]
24. Noury, N.; Fleury, A.; Rumeau, P.; Bourke, A.K.; Laighin, G.O.; Rialle, V.; Lundy, J.E. Fall detection—Principles and methods. *Conf. Proc. IEEE Eng. Med. Biol. Soc.* **2007**, *2007*, 1663–1666. [[PubMed](#)]
25. Bourke, A.; O’Brien, J.; Lyons, G. Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait Posture* **2007**, *26*, 194–199. [[CrossRef](#)] [[PubMed](#)]
26. Huang, C.N.; Chan, C.T. A zigbee-based location-aware fall detection system for improving elderly telecare. *Int. J. Environ. Res. Public Health* **2014**, *11*, 4233–4248. [[CrossRef](#)] [[PubMed](#)]
27. Kangas, M.; Konttila, A.; Winblad, I.; Jamsa, T. Determination of simple thresholds for accelerometry-based parameters for fall detection. In Proceedings of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), Lyon, France, 22–26 August 2007; pp. 1367–1370.
28. Kangas, M.; Konttila, A.; Lindgren, P.; Winblad, I.; Jämsä, T. Comparison of low-complexity fall detection algorithms for body attached accelerometers. *Gait Posture* **2008**, *28*, 285–291. [[CrossRef](#)] [[PubMed](#)]
29. Albert, M.V.; Kording, K.; Herrmann, M.; Jayaraman, A. Fall classification by machine learning using mobile phones. *PLoS ONE* **2012**, *7*, e36556. [[CrossRef](#)] [[PubMed](#)]
30. Yuwono, M.; Moulton, B.D.; Su, S.W.; Celler, B.G.; Nguyen, H.T. Unsupervised machine-learning method for improving the performance of ambulatory fall-detection systems. *BioMed. Eng. Online* **2012**, *11*, 9. [[CrossRef](#)] [[PubMed](#)]
31. Ozdemir, A.T.; Barshan, B. Detecting falls with wearable sensors using machine learning techniques. *Sensors* **2014**, *14*, 10691–10708. [[CrossRef](#)] [[PubMed](#)]
32. Choi, Y.; Ralhan, A.; Ko, S. A study on machine learning algorithms for fall detection and movement classification. In Proceedings of the 2011 International Conference on Information Science and Applications (ICISA), Jeju Island, Korea, 27–29 April 2011; pp. 1–8.
33. Apdm Wearable Technologies. Available online: <http://www.apdm.com/wearable-sensors/> (accessed on 29 August 2016).
34. Dinh, A.; Shi, Y.; Teng, D.; Ralhan, A.; Chen, L.; Dal Bello-Haas, V.; Basran, J.; Ko, S.-B.; McCrowsky, C. A fall and near-fall assessment and evaluation system. *Open Biomed. Eng. J.* **2009**, *3*, 1–7. [[CrossRef](#)] [[PubMed](#)]
35. Chao, P.K.; Chan, H.L.; Tang, F.T.; Chen, Y.C.; Wong, M.K. A comparison of automatic fall detection by the cross-product and magnitude of tri-axial acceleration. *Physiol. Meas.* **2009**, *30*, 1027–1037. [[CrossRef](#)] [[PubMed](#)]
36. Bourke, A.; Van de Ven, P.; Gamble, M.; O’Connor, R.; Murphy, K.; Bogan, E.; McQuade, E.; Finucane, P.; O’Laighin, G.; Nelson, J. Evaluation of waist-mounted tri-axial accelerometer based fall-detection algorithms during scripted and continuous unscripted activities. *J. Biomech.* **2010**, *43*, 3051–3057. [[CrossRef](#)] [[PubMed](#)]
37. Rescio, G.; Leone, A.; Siciliano, P. Support vector machine for tri-axial accelerometer-based fall detector. In Proceedings of the 2013 5th IEEE International Workshop on Advances in Sensors and Interfaces (IWASI), Bari, Italy, 13–14 June 2013; pp. 25–30.
38. Huynh, Q.T.; Nguyen, U.D.; Irazabal, L.B.; Ghassemian, N.; Tran, B.Q. Optimization of an accelerometer and gyroscope-based fall detection algorithm. *J. Sens.* **2015**, *2015*, 452078. [[CrossRef](#)]
39. Palmerini, L.; Bagala, F.; Zanetti, A.; Klenk, J.; Becker, C.; Cappello, A. A wavelet-based approach to fall detection. *Sensors* **2015**, *15*, 11575–11586. [[CrossRef](#)] [[PubMed](#)]
40. He, J.; Hu, C.; Wang, X. A smart device enabled system for autonomous fall detection and alert. *Int. J. Distrib. Sens. Netw.* **2016**, *2016*, 2308183. [[CrossRef](#)]
41. Chen, K.-H.; Yang, J.-J.; Jaw, F.-S. Accelerometer-based fall detection using feature extraction and support vector machine algorithms. *Instrum. Sci. Technol.* **2016**, *44*, 333–342. [[CrossRef](#)]
42. Gibson, R.M.; Amira, A.; Ramzan, N.; Casaseca-de-la-Higuera, P.; Pervez, Z. Multiple comparator classifier framework for accelerometer-based fall detection and diagnostic. *Appl. Soft Comput.* **2016**, *39*, 94–103. [[CrossRef](#)]

43. Klenk, J.; Becker, C.; Lieken, F.; Nicolai, S.; Maetzler, W.; Alt, W.; Zijlstra, W.; Hausdorff, J.; Van Lummel, R.; Chiari, L. Comparison of acceleration signals of simulated and real-world backward falls. *Med. Eng. Phys.* **2011**, *33*, 368–373. [[CrossRef](#)] [[PubMed](#)]
44. Cleland, I.; Kikhia, B.; Nugent, C.; Boytsov, A.; Hallberg, J.; Synnes, K.; McClean, S.; Finlay, D. Optimal placement of accelerometers for the detection of everyday activities. *Sensors* **2013**, *13*, 9183–9200. [[CrossRef](#)] [[PubMed](#)]
45. Hsu, C.W.; Lin, C.J. A comparison of methods for multiclass support vector machines. *IEEE Trans. Neural Netw.* **2002**, *13*, 415–425. [[PubMed](#)]
46. Anzai, Y. *Pattern Recognition & Machine Learning*; Elsevier: Amsterdam, The Netherlands, 2012.
47. Kangas, M.; Vikman, I.; Wiklander, J.; Lindgren, P.; Nyberg, L.; Jamsa, T. Sensitivity and specificity of fall detection in people aged 40 years and over. *Gait Posture* **2009**, *29*, 571–574. [[CrossRef](#)] [[PubMed](#)]
48. Bagala, F.; Becker, C.; Cappello, A.; Chiari, L.; Aminian, K.; Hausdorff, J.M.; Zijlstra, W.; Klenk, J. Evaluation of accelerometer-based fall detection algorithms on real-world falls. *PLoS ONE* **2012**, *7*, e37062. [[CrossRef](#)] [[PubMed](#)]
49. Bersch, S.D.; Azzi, D.; Khusainov, R.; Achumba, I.E.; Ries, J. Sensor data acquisition and processing parameters for human activity classification. *Sensors* **2014**, *14*, 4239–4270. [[CrossRef](#)] [[PubMed](#)]
50. Chawla, N.V. Data mining for imbalanced datasets: An overview. In *Data Mining and Knowledge Discovery Handbook*; Springer: New York, NY, USA, 2009; pp. 875–886.
51. Fortino, G.; Gravina, R. Fall-mobileguard: A smart real-time fall detection system. In Proceedings of the 10th EAI International Conference on Body Area Networks, Sydney, Australia, 28–30 September 2015; ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering): South Portland, ME, USA, 2015; pp. 44–50.
52. Kangas, M.; Vikman, I.; Nyberg, L.; Korpelainen, R.; Lindblom, J.; Jamsa, T. Comparison of real-life accidental falls in older people with experimental falls in middle-aged test subjects. *Gait Posture* **2012**, *35*, 500–505. [[CrossRef](#)] [[PubMed](#)]



© 2017 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).