Fall Detection in Smart Home Environments Using UWB Sensors and Unsupervised Change Detection

G. Mokhtari[1], S. Aminikhanghahi[2], Q. Zhang[1], D. J. Cook[2]
[1]Australian e-Health Research Centre, CSIRO, Brisbane, Australia.
[2]Washington State University, Pullman, USA.

***Abstract*—Falls are one of the major issues which can endanger lives for older adults. Numerous research studies investigate the use of wearable technologies to detect falls in everyday environments. Although wearable sensor solutions provide good accuracy and sensitivity for fall detection, it may not always be convenient or desirable for older adults to wear a tag or sensor in home environments. This paper discusses using non-wearable UWB radar sensors as a practical, environmental fall detection solution in home settings. Specifically, we apply unsupervised change detection methods on UWB sensor data to detect falls. Furthermore, to evaluate the generality of our unsupervised approach, we also apply it to fall detections from accelerometer sensor data. The proposed methods are assessed using the real UWB sensor data sets acquired from the Living Lab at Australian e-Health Research Centre and public available accelerometer sensor data sets. The results show promising outcomes.**

Keywords—Change detection, Fall detection, Unsupervised learning algorithms, UWB sensor, Smart homes

I. INTRODUCTION

Future homes will include different types of sensing technlogies to monitor resident activities for different applications including health, energy utilization, and safety. For the case of health monitoring, smart home technology shows promising outcomes for providing independent living solutions for older adults in their own homes [1]. In this platform, different types of activity and vital sign sensors are used which monitor activities and health status to detect and respond to any abnormal situation [2]. Different types of activities such as cooking, bathing, sleeping, and even moving around the space are considered which are related to the health status of older adults [3]. Monitoring these activities allows researchers to detect gradual changes in activity patterns that signify possible changes in cognitive and physical health [4]. However, in some cases real-time monitoring of movement and activities is required to detect falls, which require proper and fast detection to avoid life-threatening danger.

Recently, there has been extensive work on using different smart technologies to detect falls. The fall detection approaches listed in the literature can be categorised into three different classes. Wearable devices are one type of fall detection technologies which use inertial sensors such as accelerometers and gyroscopes to detect certain classes of falls [5]. For example, PerFallD [6] uses a smartphone app to detect falls in which accuracy can be affected by the position of the smartphone placed in body. One limitation of using these devices for fall detection is that they put the burden on residents to charge, correctly place, and wear the sensors. Adherence to these constraints is not always strong because people usually do not like to use them while moving around the home [6].

Vision-based technologies such as cameras represent another technology which can also provide an accurate solution for fall detection. Some researchers have employed CCD and Kinect cameras to detect resident falls [8],[9]. Although cameras can also provide good fall detection accuracy [10], the corresponding privacy issue is a concern for using them in a smart home environment.

Non-wearable and ambient devices represent another class of technologies which can be used for multiple smart home applications including fall detection. As these technologies do not put any burden on residents and do not raise many privacy concerns, they are a promising solution for smart home-based health monitoring. In work by Liu et al. [10], Passive Infrared (PIR) motion sensors are used which can produce unique motion reading patterns that correspond to resident movement patterns. These patterns can be analyzed using machine learning or rule-based software to distinguish falls from other types of movements such as walking or lying down. WiFall is a device-free fall detection system which

leverages channel state information in order to detect falls in smart homes [12].

In this paper, we propose to evaluate the use of an ultra-wide-band (UWB) sensor as a non-wearable solution to detect falls in smart home environments. In this set-up, a UWB sensor which includes both transmitter and receiver is installed on the ceiling to monitor different movement activities in its detection zone. The UWB transmitter emits a train of pulses to the environment which are scattered by different objects and stored by the receiver. The scattered signal will be analyzed to detect any movement activity including falls in the sensor monitoring area.

The use of a UWB sensor as a non-wearable technology has been assessed in prior work [3]. However, the main focus of the earlier work is on supervised fall detection. In contrast, in this paper we propose to provide unsupervised fall detection. Unsupervised methods are preferred if they provide effective detection without time-consuming and costly expert labeling. The main contributions of this paper include the following:

- introducing UWB technology as a non-wearable solution to detect falls in home environments,
- designing unsupervised change detection algorithms to detect falls in home environments, and
- comparing the performance of supervised and unsupervised fall detection algorithms based on actual fall sensor data.

II. Related work

A. UWB technology

UWB technology has been tested in multiple radar applications such as target and movement detection due to its high resolution and penetrability [6]. Recently, there has been much interest in using this technology for smart home environments [11], [12]. In work by Mokhtari et al. [6], this technology is used to detect and identify residents in smart homes. Jeon et al. [13] use this technology for the purpose of localization. Breath monitoring represents yet another application of this technology in smart home environments, as investigated by Pitella [14].

To use UWB technology for indoor applications, usually a UWB transmitter as well as receiver are used. The UWB transmitter propagates a train of pulses (*p(t)*). The pulse train will propagate in the environment and scatter from the objects. The scattered signal will be received by the receiver with different times of arrival (TOA). The received signal corresponding to the *k*th pulse R_k, can be wriiten as a vector (frame) in discrete time domain as shown in Equation (1),

$$R_k = [R_k(1), R_k(2), \dots, R_k(M)] \quad (1)$$

where M represents the number of samples in each frame.

B. Unsupervised change point detection techniques

Change point detection, based on unsupervised learning or supervised learning methods, represents a well-investigated area of research. A change point is a point within a data time series at which the process generating the time series changes state. A change point detection algorithm tries to find the data point in a time series at which the state change occurs and segment the time series based on statistical features of the data. Segmenting time series using unsupervised method is desirable because it does not need prior training for each situation and can handle a variety of real world problems.

Some successful studies demonstrated promising change point detection performance using probabilistic methods. When a new window of data arrives, these algorithms estimates probability distributions based on the data that has been observed since the previous detected change point. As an example, Adams and McKay [15] use Bayes' theorem to estimate a current state's run-length (r_t), which represents the time that has elapsed, or the number of data points in the time series that have been observed, since the last change point. Another example in this category is the Gaussian Process (GP) algorithm which was introduced by Saatçi, et al. [16]. They define a time series data points as noisy Gaussian distribution function values and create a normal distribution-based prediction of the data point at time t. If the predicted data point is different than the actual data point, it will be considered as a change point.

Recently, density ratio change point detection (CPD) techniques have been used to detect changes. These CPD techniques compare the probability distributions of data intervals before and after a possible change point, and decide if there the candidate is a change point or not based on the difference between the two corresponding distributions. As one example, cumulative sum (CUSUM) [17] accumulates deviations relative to a specified target of incoming measurements and identifies change points when the sum is greater than a threshold. Another method in this category is change finder (CF) [18], which is an outlier detection algorithm. Both CUSUM and CF use pre-designed parametric models to detect changes which make them less flexible in real-world problems.

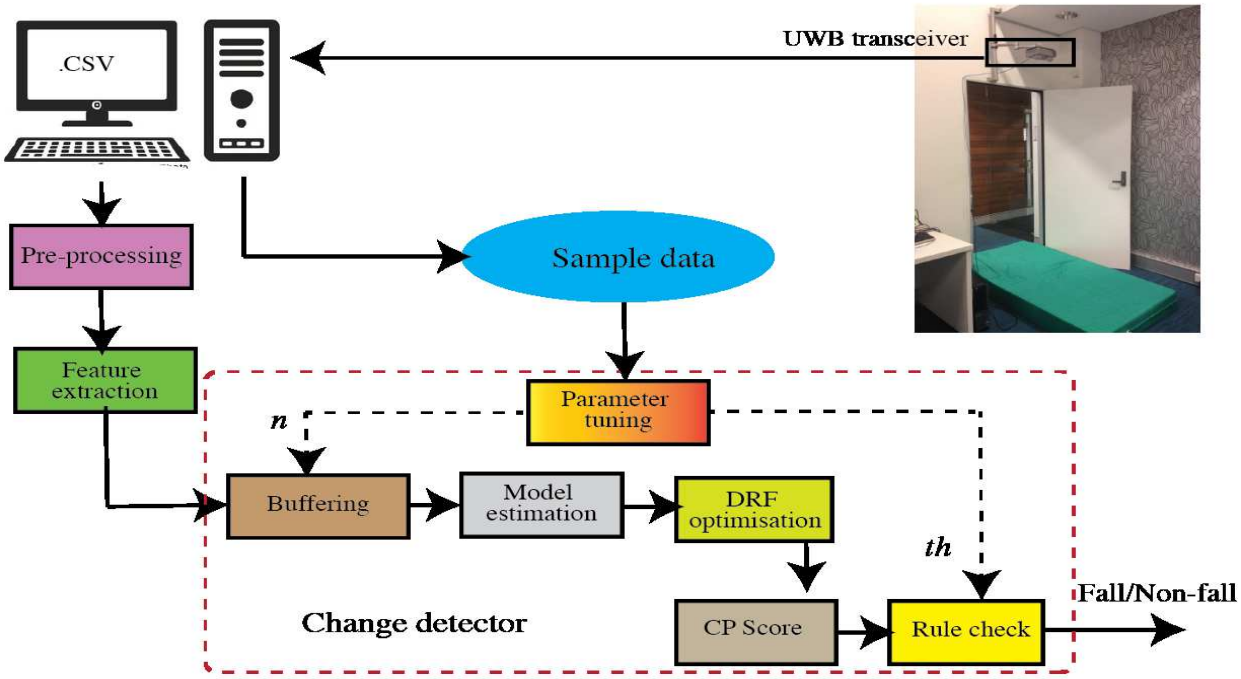

Fig. 1: Proposed unsupervised fall detection setup.

Some recent approaches estimate the ratio of probability densities instead of performing density estimation which makes the algorithm more flexible and non-parametric. These density ratio-based approaches to change point detection are among the most popular approaches and form the basis of our SEP method described in the next section. The motivation of estimating density ratios is that it gives us enough information about the change between two densities without knowing the exact densities. Thus, direct density-ratio estimation is substantially simpler than density estimation. Therefore, direct density-ratio estimation methods have been developed [19][20].

We hypothesize that falls represent a change in time series data representing movement patterns in a smart home and thus can be detected using change point detection techniques. In this paper, we introduce a novel direct density ratio-based change point detection as a key component of unsupervised fall detection. We compare the performance of this method with the Bayesian CPD algorithm as well as supervised fall detection techniques.

III. Unsupervised Fall Detection Approach

The proposed unsupervised fall detection process is shown in Fig. 1. The entire process is divided into four main steps:

- ***Preprocessing,*** which applies noise and DC reduction to UWB data,
- ***Feature extraction,*** which changes the UWB sequence into time series data,
- ***SEP change detection,*** which detects change points in the time series, and
- ***Rule-based fall detection,*** distinguishes falls from other movement activities in the sensor coverage area.

The details of each step are described below.

A. Pre-processing

Data preprocessing consists of three tasks: DC noise reduction, background removal, and windowing. For each received frame these three tasks can be performed as indicated in Equations (2)-(4).

$$R_i(k) = R_i(k) - \frac{\sum_{j=1}^{M} R_i(j)}{M} \quad i = 1,2,\dots,M \tag{2}$$

$$R_i(k) = R_i(k) - \frac{\sum_{j=1}^{P} R_j(k)}{P} \quad i = 1,2,\dots,M \tag{3}$$

$$W_i(k-n+1) = \begin{cases} 0, & k < n, k > m \\ R_i(k), & n \le k \le m \end{cases} \quad k = 1,2,\dots,M \tag{4}$$

where L is the frame number for static obstacle modeling and n-m+1 is the size of the window W which is used to extract information related to the target.

B. Feature extraction

The UWB received signal will include multiple paths from different body scattering centers which arrive at the receiver with different times of arrival (TOAs). As the value of TOA is proportional to the distance between the sensor and the scattering center, it can be said that when the target moves, TOAs will change proportionally. Therefore, a TOA-based feature as shown in Equation (5) can model each frame into a single value which is proportional to the average of the TOAs for the scattering centers.

$$TOA_i = \sum_{j=1}^{n-m+i} t(j) \,.\, W_i(j)^2 \tag{5}$$

As shown by Mokhtari et al. [3], this feature can easily extract a unique pattern for different movement activities such as normal walk, fast walk, climbing, sitting, lying down, and fall.

$$W_i(k-n+1) = \begin{cases} 0, & k < n, k > m \\ R_i(k), & n \le k \le m \end{cases} \quad k = 1,2,\dots,M \tag{6}$$

C. SEP Change Point Detection

Detecting sudden changes in smart home data due to emergency situations such as falls can be formulated as a real-time change point detection problem. After generating a time series from sensor data, we look for changes in this data. To do this, we need to store two consecutive windows of data in a buffer in order to compare data between each pair of consecutive windows.

Direct density ratio change point detection algorithms are flexible non-parametric techniques that estimate the ratio of probability densities between two windows of data directly without needing to perform density estimation. The main idea is that estimating the density ratio is much easier than estimating the individual densities and is still sufficient for detecting changes [20]. Following this idea, the Kullback-Leibler importance estimation procedure, KLIEP, [21] was developed. KLIEP uses a nonparametric Gaussian kernel model to estimate the density ratio and calculates the change point score using Kullback–Leibler (KL) divergence. Another direct density ratio estimator is the unconstrained least-squares importance fitting, or uLSIF [22], which uses Pearson (PE) divergence to calculate dissimilarity and estimates the change point score using a Gaussian. Relative uLSIF – RuLSIF [20] refines the dissimilarity measure using α-relative PE divergence.

Recently, the SEParation (SEP) change point detection [25] method was introduced based on a hypothesis that dissimilarity metrics with higher resolution in calculating distances yield more sensitive change point detection. The SEP change point detection algorithm uses a Separation distance metric, S, which provides a high distance resolution. Considering two probability densities, $f_t(x)$ and $f_{t-1}(x)$, corresponding to two consecutive windows, each with length n, the separation distance S between them can be calculated as shown in Equation 7.

$$S = Max\left(1 - \frac{f_{t-1}(x)}{f_t(x)}\right) \tag{7}$$

We start by deriving the metric for our SEParation distance CPD algorithm, called SEP. As with the previous methods, we compare the probability densities of $f_t(x)$ and $f_{t-1}(x)$ corresponding to two consecutive windows in the time series data, each with length n. We model the density ratio between these probability densities using a Gaussian kernel function K, as shown in Equations 8 and 9.

$$g_t(x) = \frac{f_{t-1}(x)}{f_t(x)} = \sum_{i=1}^{n} \theta_l \prod_{j=1}^{n} K\left(x_t^i, x_{t-1}^j\right) \tag{8}$$

$$K(x_1, x_2) = exp\left(-\frac{\|x_1 - x_2\|^2}{2\sigma^2}\right) \quad (9)$$

In these equations, $\theta = (\theta_1, \dots, \theta_n)^T$ represents the set of parameters for the ratio function to be learned from existing data points and $\sigma > 0$ represents the kernel parameter. For every pair of consecutive windows, the parameters θ are selected to minimize a chosen dissimilarity measure. We determine the parameters θ in the model such that the difference between the actual and estimated ratios is minimized, as shown in Equation 10.

$$J(x) = \int \left|\frac{f_{t-1}(x)}{f_t(x)} - g_t(x)\right| f_t(x)\, dx$$
$$= \begin{cases} \int -[f_{t-1}(x) - g_t(x) f_t(x)]\, dx, & \frac{f_{t-1}(x)}{f_t(x)} < g_t(x) \\ \int [f_t(x) - g_t(x) f_t(x)]\, dx, & \frac{f_{t-1}(x)}{f_t(x)} \geq g_t(x) \end{cases} \quad (10)$$

By substituting $g_t(x)$ into Equation 10 and approximating the integrals using empirical averages, we can convert the minimization problem to the one shown in Equation 11.

$$min_\theta \left[|\hat{h}^T \theta| + \frac{\lambda}{2}\theta^T \theta\right] \quad (11)$$

The second term in Equation 11 is included for the purpose of regularization. $\lambda \geq 0$ represents the regularization parameter, which is selected empirically by cross-validation [20]. $\hat{h}$ is the n-dimensional vector which is defined in Equation 12.

$$\hat{h}_l = \frac{1}{n}\sum_{i=1}^{n} K(x_t^j, x_{t-1}^l) \quad (12)$$

When solving the optimization by setting the first-order differential to zero, parameter θ can be analytically obtained as defined in Equation 13.

$$\theta = -\frac{1}{\lambda}\hat{h} \quad (13)$$

In the next step an approximator of the SEP change point score can be calculated using the density-ratio estimator $g_t(x)$, as shown in Equation 14.

$$\widehat{SEP} = |\frac{1}{2} - \frac{1}{n}\sum_{i=1}^{n} g(x_i)| \quad (14)$$

We can then use SEP scores to detect change points in our time series data and label these as possible falls. Considering the fact that a larger SEP score means that the probability of a change point is greater, we reject all candidate points whose SEP values are lower than a threshold value. The threshold value will be chosen based on optimal performance for a particular time series. In our experiments, we identify a threshold value that optimizes the tradeoff between TPR and FPR for a subset of the data. Another important parameter in SEP is the window length, *n*, which makes the algorithm *n*-real time. As a result, the SEP algorithm needs to observe *n* data points in the future to detect possible falls at the current point in time. As with the threshold value, we vary the window size for each dataset in order to find the best window length in terms of both acceptable accuracy and real-time detection.

Although we still need to choose both the threshold value and the window length using a sample of pre-labeled data, the general configuration would be good enough to detect changes, and change-related falls, without training.

D. *Rule-Enhanced Fall Detection*

Applying SEP change point detection to UWB sensor data will remove all of the activities from consideration as possible falls except lying on the ground. In some cases, the lying down activity is detected as a fall-positive fall because falls and lying down have similar movement patterns. Although the speed of change in a fall activity is faster than lying down, the transition time of the lying activity is still smaller than change point detection algorithm window length.

In order to distinguish between falls and lying-down activities, we design a semantic reasoner to analyze the time series data after a change is detected. Investigating samples of fall and lying activity data, we find the data mean for fall activities is higher than for lying activities. Thus, the only rule we need after detecting change points is to examine the data mean in the second window. If the value is greater than a threshold there is a fall at the detected change point.

IV. RESULTS

In this section, performance of the proposed unsupervised fall detection method is evaluated experimentally. We describe our experimental conditions and performance measures then summarize the results of the method applied to collected sensor data. We also show the generality of our proposed unsupervised fall detection model by applying it on accelerometer data.

A. UWB Data

1) Experimental Setup

We employ the configuration at the Australian e-Health Research Center for our experimental evaluation. The UWB sensor used in this set-up is manufactured by NOVELDA. As shown in Fig. 2, this sensor is mounted over the door frame to detect movement activities in its detection zone. This sensor is connected to a computer to record the UWB raw data. The UWB frame rate in this setup is 100 Hz while each frame includes 512 samples.

To simulate different movement activities, two individuals participated in the experiments. For the scenarios listed in Table 1, each individual performed each movement activity 20 times. In total, three datasets are collected and are labeled Individual 1 (120 scenarios), Individual 2 (120 scenarios and Both Individuals (240 scenarios).

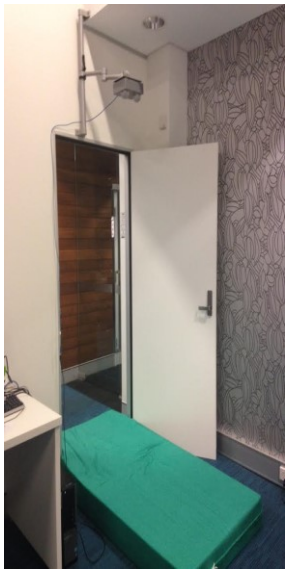

Fig. 2: Experimental setup.

Table 1: Experiment scenarios.

	Activity	**Scenario**	Number
Fall	Forward fall	Walk at normal speed, trip and fall forward onto the floor.	20
	Side fall	Walk at normal speed, trip and fall sideward on the floor.	20
Non-fall	Normal walk	Walk through the doorway at normal speed.	20
	Trip and Recovery (T/R)	Walk at normal speed, trip but recover and carry on walking.	20
	Fast walk	Walk through the doorway at fast speed.	20
	Lying	Walk through the doorway, stop and lay down backward onto the floor.	20

2) Parameter Tuning

Determination of SEP change point detection algorithm parameters is important because they greatly influence its performance. A sensitivity analysis of these parameters is very important to validate the current model and serves to guide future research efforts.

We apply random search on a subset of data including two instances of each activity performed by Individual 1 for parameter tuning of the SEP algorithm. The random search samples algorithm parameters from a random distribution for a fixed number of iterations. A model is constructed and evaluated for each chosen combination of parameters. These parameters include window size (n) and the threshold value for identifying a possible fall activity from the SEP score (*threshold*), respectively.

Fig. 3 shows the results of the random search. Too small of a window does not contain enough information about activities and as can be seen from the plot, the resulting accuracy is very low. Larger windows not only contain too much information and make detecting fall-related changes more difficult, but they also incur a greater computational cost. Based on existing data, the best window size for fall detection is between 50 and 150 data points. Considering our sensor sampling rate of 0.01 seconds, this means that the SEP algorithm needs to look ahead 0.5 to 1.5 seconds to detect a fall. This also represents the expected delay between falling and the smart home detecting (and responding to) a fall. A random search of alternative threshold values shows larger thresholds decrease the number of detected changes. Using the random search results

and considering the fact that we prefer lower computational cost for a real-time fall detection process, we choose *n=100* and *threshold=0.3*. Therefore, the fall detection algorithm is 100-real time, which means the algorithm looks ahead one hundred sensor events to detect falls.

3) Fall Detection Performance

A number of different measures are commonly used to evaluate the performance of fall detection methods. Among these, we use three different performance metrics to evaluate the ability of our system to detect both fall and non-fall data points in the time series. These are accuracy (utilizing both true positive and true negative measures to assess the overall performance to distinguish falls from non-fall activities), precision (the ratio of detected fall activities to all activities), and Detection Delay (the elapsed time between the detected fall and the actual fall activity). In these experiments, we ignore falls detected within 10 seconds of a previously-detected fall.

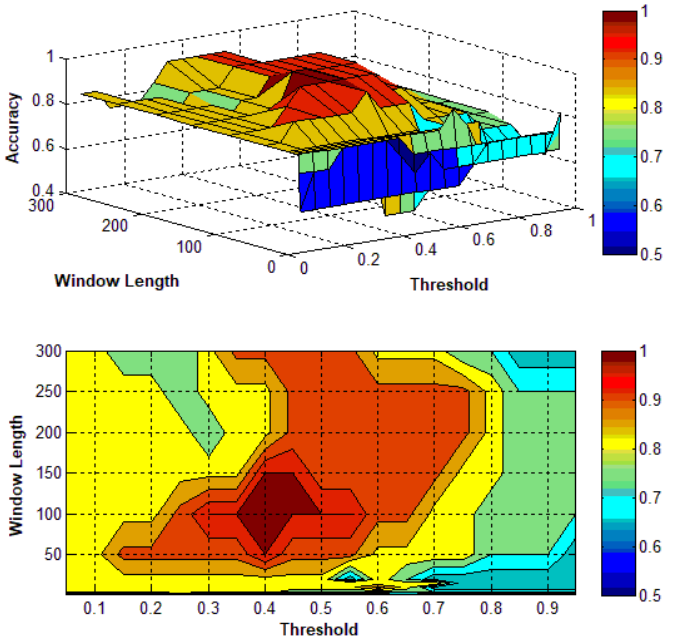

Fig. 3: SEP algorithm parameter tuning.

To evaluate the ability of detecting falls, we compare our SEP-based approach with three alternative methods.

Method 1: RuLSIF. Relative unconstrained Least-Squares Importance Fitting (RuLSIF) [20] is a non-parametric window-based change point detection algorithm which utilizes the Pearson divergence dissimilarity measure to estimate the change point score. After applying the sensitivity analysis preprocessing step, the window length and the threshold values for the RuLSIF were set to 100 and 0.75, respectively.

Method 2: BCPD. Bayesian Change Point Detection (BCPD) [15] uses Bayes' theorem to estimate a current state's run-length (r_t) which represents the time that has elapsed in the time series since the last change point. BCPD set a run length in each time point to 0 if a change point occurs or increase the previous length by 1 if the current state continues for one more time unit.

Method 3: Supervised fall detection. Our supervised fall detection system is based on Mokhtari et al.'s approach [3]. The z-score test was applied to sliding window of size 10 frames to compare the mean value between two windows and detect the starting point of each activity. After detecting the starting point, the streamed frames of size 500 are buffered from the starting point. When the buffer is full, a binary classifier is used to label fall and non-fall activities. 3-fold cross validation is used to evaluate the performance of this classifier.

In all cases, we evaluate the fall detection algorithm for each individual separately as well as for participants combined. In the case of supervised algorithms, this type of evaluation shows the dependency of the algorithm on subject-specific training and testing data. For an unsupervised method, this evaluation shows how much the algorithm is general and can be used for different individuals although we select parameters based on just one individual.

We begin by studying the performance of different classifiers including Random Forest (RF), Logistic Regression (LR), Nearest Centroid (NC), Decision Tree (DT), Support Vector Machine (SVM), and Naïve Bayes (NB) for supervised fall detection. The accuracy and precision values for this experiment are summarized in Table 2. It can be observed that the RF, NC, and SVM classifiers each yield a precision of 100% which means they successfully detect all fall activities. The overall accuracy of fall detection is almost the same in the case of using RF or SVM classifiers but both of them

have slightly better accuracy than NC. Although there is not a significant difference between RF and SVM classifier models, due to its lower computational cost we use RF as our supervised classifier for the remainder of the paper.

Figures 4 and 5 show the accuracy and sensitivity of the proposed SEP, RuLSIF, BCPD, and Supervised RF methods, respectively. The graphs show that SEP outperforms all other algorithms in case of accuracy for each individual separately and combined. However, in terms of sensitivity both SEP and the supervised method are able to detect all falls. In summary, among unsupervised algorithms, SEP has better performance in detecting falls and non-falls activities. Comparing to supervised method, although both method can detect all falls, the supervised algorithm considers more non-fall activities as fall. Furthermore, the supervised method requires a sufficient number of labeled fall examples to accurately learn the concept which is not needed for the unsupervised method.

Table 2. Classsification accuracy (%) of movement activities.

Classifier	Individual 1	Individual 2	All data
RF	**AR: 95.74** **PR: 100**	**AR: 91.48** **PR: 100**	**AR: 92.87** **PR: 100**
LR	AR: 90.70 PR: 99.25	AR: 86.68 PR: 99.00	AR: 88.08 PR: 99.75
NC	AR: 93.14 PR: 100	AR: 86.20 PR: 100	AR: 92.08 PR:100
DT	AR: 86.46 PR:95.75	AR: 79.40 PR: 95.25	AR: 87.14 PR:99.125
SVM	AR: 95.93 PR: 100	AR: 90.55 PR: 100	AR: 92.80 PR: 100
NB	AR: 81.64 PR: 77.25	AR: 71.31 PR: 81.00	AR: 77.05 PR: 76.87

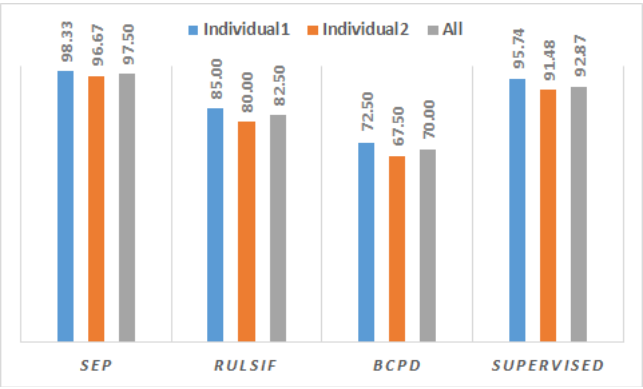

Fig. 4: Fall detection algorithm accuracy.

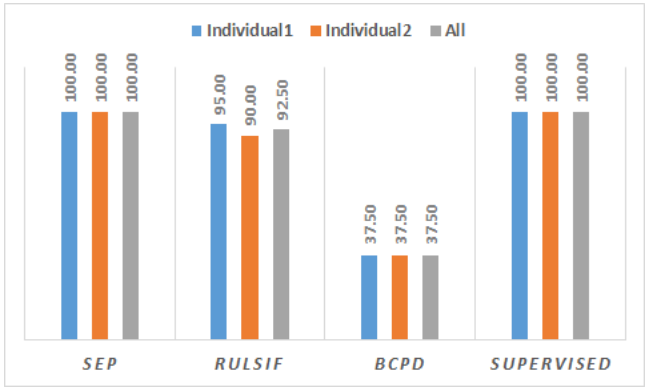

Fig. 5: Fall detection algorithm sensitivity.

Finally, the detection delay for all algorithms is plotted in Fig. 6. The detection delay can be thought of as the average time between when a fall occurs and when the fall is detected. Ideally, we would like to minimize these times to identify falls and help residents as soon as possible. For the SEP algorithm, the average value is close to 2 seconds, which is much lower than supervised method (5 seconds). These values for the RuLSIF and BCPD algorithms are 1.58 and 1.06 seconds, which both are faster than SEP in detecting falls. By considering the fact that SEP has much higher accuracy in detecting falls and the difference between the detection delays is less than 1 second, we can conclude that SEP is a better choice for a fall detection algorithm.

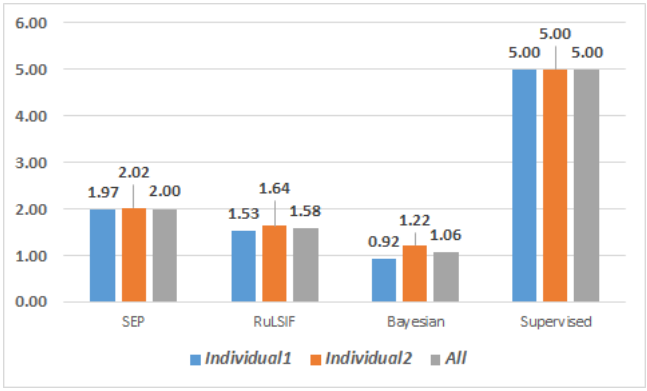

Fig. 6: Fall detection algorithm detection delay.

B. *Accelerometric Data*

To show the generality of our unsupervised fall detection, we evaluate its performance on UR fall detection dataset [26]. This dataset contains 30 falls activities and 40 activities of daily living including sitting down, crouching down, and picking up an object from the floor and lying on the sofa. All events are recorded with Microsoft Kinect cameras and

corresponding accelerometer data. Here we only use the accelerometer data which collected using x-IMU (256Hz) devices. The measured acceleration components were median filtered with a window length of three samples to suppress the sensor noise. The x-IMU inertial device consists of triple-axis 12-bit accelerometer. The sampled acceleration components were used to calculate the total sum vector as follows:

$$SV(t) = \sqrt{a_x^2(t) + a_y^2(t) + a_z^2(t)} \quad (15)$$

where $a_x(t)$, $a_y(t)$, and $a_z(t)$ are the acceleration in the x, y, and z axes at time t, respectively. The SV contains both the dynamic and static acceleration components, and thus it is equal to 1 g for standing.

In the next step, we randomly select 3 fall and 5 non-fall activities for parameter tuning and then apply our unsupervised fall detection to all data. We compare the results with existing supervised SVM algorithm which uses images and corresponding acceleration data and threshold-based method which only uses accelerometer data [26][27]. Table 3 demonstrates the accuracy and sensitivity different algorithms. All three methods can detect fall activity correctly and the sensitivity is 100 percent. But the ability of algorithm to detect non fall ADL activities is different which cause difference in their accuracy. Threshold based method has lower accuracy at 95% while supervised SVM has highest accuracy at 98%. Our unsupervised SEP has the accuracy of 97% which is higher than threshold based method but still lower than supervised method. Recalling that supervised SVM is using both accelerometer and camera data while unsupervised SEP is working just based on accelerometer data, we can conclude the unsupervised SEP method is simpler and address the privacy issue more than existing supervised SVM algorithm.

Table 3. Performance of accelerometer data fall detection

	SEP	SVM	Threshold-based
Accuracy (%)	97.14	98.33	95.00
Sensitivity (%)	100.00	100.00	100.00

V. Conclusions

In this paper, an unsupervised approach is proposed to use UWB sensor data to detect and distinguish fall from other types of activities. In the proposed experimental set-up, a UWB sensor is mounted over the ceiling to monitor movement activities in areas such as bathroom which are more vulnerable. Two individuals simulate different types of movement activities including falls, normal walk, fast walk and lying which results in three data sets. Unsupervised fall detection is performed using our SEP-based change point detection algorithm. The results of unsupervised fall detection on different data sets are provided in results section and are compared with the supervised fall detection. The results indicate that SEP-based unsupervised detection is as accurate as the supervised method, but it can detect falls much quicker than other algorithms, although all of the methods would benefit from increased computational efficiency.

VI. Ethics Committee Approval

The UWB participant data was collected with ethics approval from CSIRO Health and Medical Research Ethics Committee– Proposal #LR 12/2016. This work was supported in part by the National Science Foundation under Grant No. 1543656.

VII. References

[1] S. Greene, H. Thapliyal, and D. Carpenter, "IoT-Based Fall Detection for Smart Home Environments," 2016, pp. 23–28.

[2] G. Mokhtari, Q. Zhang, G. Nourbakhsh, S. Ball, and M. Karunanithi, "BLUESOUND: A New Resident Identification Sensor—Using Ultrasound Array and BLE Technology for Smart Home Platform," *IEEE Sens. J.*, vol. 17, no. 5, pp. 1503–1512, 2017.

[3] G. Mokhtari, Q. Zhang, and A. Fazlollahi, "Non-wearable UWB sensor to detect falls in smart home environment," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops*, 2017, pp. 274–278.

[4] A. Alberdi *et al.*, "Smart home-based prediction of multi-domain symptoms related to Alzheimer's Disease," *J. Biomed. Health Inform.*, 2018.

[5] A. Bourke, J. O'brien, and G. Lyons, "Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm," *Gait Posture*, vol. 26, no. 2, pp. 194–199, 2007.

[6] J. Dai, X. Bai, Z. Yang, Z. Shen, and D. Xuan, "PerFallD: A pervasive fall detection system using mobile phones," presented at the Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on, 2010, pp. 292–297.

[7] G. Mokhtari, Q. Zhang, C. Hargrave, and J. C. Ralston, "Non-Wearable UWB Sensor for Human Identification in Smart Home," *IEEE Sens. J.*, vol. 17, no. 11, pp. 3332–3340, 2017.
[8] V. Vishwakarma, C. Mandal, and S. Sural, "Automatic detection of human fall in video," in *Pattern Recognition and Machine Intelligence*, Springer, 2007, pp. 616–623.
[9] B. Ni, C. D. Nguyen, and P. Moulin, "RGBD-camera based get-up event detection for hospital fall prevention," 2012, pp. 1405–1408.
[10] M. Skubic, B. H. Harris, E. Stone, K. C. Ho, B.-Y. Su, and M. Rantz, "Testing non-wearable fall detection methods in the homes of older adults," in *IEEE International Conference of the Engineering in Medicine and Biology Society*, 2016, pp. 557–560.
[11] T. Liu, X. Guo, and G. Wang, "Elderly-falling detection using distributed direction-sensitive pyroelectric infrared sensor arrays," *Multidimens. Syst. Signal Process.*, vol. 23, no. 4, pp. 451–467, 2012.
[12] Y. Wang, K. Wu, and L. M. Ni, "Wifall: Device-free fall detection by wireless networks," *IEEE Trans. Mob. Comput.*, vol. 16, no. 2, pp. 581–594, 2017.
[13] P. Kumar, A. Braeken, A. Gurtov, J. Iinatti, and P. H. Ha, "Anonymous Secure Framework in Connected Smart Home Environments," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 4, pp. 968–979, 2017.
[14] E. Hou, L. Dai, and Z. Wen, "Method, apparatus and electronic device for controlling smart home device," Jun. 2017.
[15] S. Jeon, K.-D. Kang, H. Lee, and S. H. Son, "Smart-Bin Using Ultrawideband Localization to Assist People with Movement Disabilities," 2016, pp. 259–259.
[16] E. Pittella, "UWB Radar System for Breath Activity Monitoring," no. December, 2010.
[17] R. P. Adams and D. J. C. MacKay, "Bayesian Online Changepoint Detection," Machine Learning, Oct. 2007.
[18] Y. Saatçi, R. D. Turner, and C. E. Rasmussen, "Gaussian Process Change Point Models," in *International Conference on Machine Learning*, 2010, pp. 927–934.
[19] M. Basseville and I. Nikiforov, *Detection of abrupt changes: theory and application*. Englewood Cliffs: Prentice Hall, 1993.
[20] K. Yamanishi and J. Takeuchi, "A unifying framework for detecting outliers and change points from non-stationary time series data," in *8th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '02*, 2002, p. 676.
[21] Y. Kawahara and M. Sugiyama, "Sequential Change-Point Detection Based on Direct Density-Ratio Estimation," in *SIAM International Conference on Data Mining*, 2009, pp. 389–400.
[22] S. Liu, M. Yamada, N. Collier, and M. Sugiyama, "Change-point detection in time-series data by relative density-ratio estimation.," *Neural Netw. Off. J. Int. Neural Netw. Soc.*, vol. 43, pp. 72–83, Jul. 2013.
[23] M. Sugiyama, T. Suzuki, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe, "Direct importance estimation for covariate shift adaptation," *Ann. Inst. Stat. Math.*, vol. 60, no. 4, pp. 699–746, 2008.
[24] T. Kanamori, S. Hido, and M. Sugiyama, "A Least-squares Approach to Direct Importance Estimation," *J. Mach. Learn. Res.*, vol. 10, pp. 1391–1445, 2009.
[25] S. Aminikhanghahi, T. Wang, D. J. Cook, and I. Fellow, "Real-Time Change Point Detection with application to Smart Home Time Series Data," *Submitt. IEEE Trans. Knowl. Data Eng.*, 2018.
[26] B. Kwolek and M. Kepski, "Human fall detection on embedded platform using depth maps and wireless accelerometer," *Comput. Methods Programs Biomed.*, vol. 117, no. 3, pp. 489–501, Dec. 2014.
[27] A. K. Bourke, J. V. O'Brien, and G. M. Lyons, "Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm," *Gait Posture*, vol. 26, no. 2, pp. 194–199, Jul. 2007.