bme BioMedical Engineering OnLine

Unsupervised machine-learning method for improving the performance of ambulatory fall-detection systems

Mitchell Yuwono[1], Bruce D Moulton[1*], Steven W Su[1], Branko G Celler[2] and Hung T Nguyen[1]

* Correspondence: Bruce.
Moulton@uts.edu.au
[1]Faculty of Engineering and IT,
University of Technology Sydney,
NSW, 2007, Australia
Full list of author information is
available at the end of the article

Abstract

Background: Falls can cause trauma, disability and death among older people. Ambulatory accelerometer devices are currently capable of detecting falls in a controlled environment. However, research suggests that most current approaches can tend to have insufficient sensitivity and specificity in non-laboratory environments, in part because impacts can be experienced as part of ordinary daily living activities.

Method: We used a waist-worn wireless tri-axial accelerometer combined with digital signal processing, clustering and neural network classifiers. The method includes the application of Discrete Wavelet Transform, Regrouping Particle Swarm Optimization, Gaussian Distribution of Clustered Knowledge and an ensemble of classifiers including a multilayer perceptron and Augmented Radial Basis Function (ARBF) neural networks.

Results: Preliminary testing with 8 healthy individuals in a home environment yields 98.6% sensitivity to falls and 99.6% specificity for routine Activities of Daily Living (ADL) data. Single ARB and MLP classifiers were compared with a combined classifier. The combined classifier offers the greatest sensitivity, with a slight reduction in specificity for routine ADL and an increased specificity for exercise activities. In preliminary tests, the approach achieves 100% sensitivity on in-group falls, 97.65% on out-group falls, 99.33% specificity on routine ADL, and 96.59% specificity on exercise ADL.

Conclusion: The pre-processing and feature-extraction steps appear to simplify the signal while successfully extracting the essential features that are required to characterize a fall. The results suggest this combination of classifiers can perform better than MLP alone. Preliminary testing suggests these methods may be useful for researchers who are attempting to improve the performance of ambulatory fall-detection systems.

Background

Falls are recognised by the World Health Organization as a major cause of hospitalization of older people [1]. If no preventative measures are undertaken, it is estimated that costs associated with fall-related trauma will double over the next 20 years [1].

Ambulatory accelerometer devices are currently capable of detecting falls in a controlled environment, and these devices are also potentially useful for assessing gait and

tremor in older people with Parkinson's disease [2,3]. Research regarding acceler-ometer-based fall detection typically uses thresholding algorithms [4-6]. Those algo-rithms typically determine if a person experiences an acceleration above a certain value - that is, an impact acceleration as a person hits the ground - and sometimes combine this with an approach for measuring whether the impact is followed by a period of lying down/not moving. However, research suggests that those approaches can tend to have limited sensitivity for soft-falls (for example, where a person falls against a wall) and break-falls (for example, where a person reduces the impact of their fall by putting out an arm) [e.g. [7]]. Current fall-detection approaches can also tend to result in poor specificity, in part because impacts can be experienced as part of ordinary daily living activities.

The fall-detection work undertaken at the University of Technology Sydney is part of a larger research program focused on health technologies that also focuses on issues associated with sensor-transceivers that stream data such as heart rate, electrocardio-gram (ECG), oxygen saturation, body temperature, and body position [8-11].

Method

The method described here builds on prior work including the work of Shi and others who combined threshold techniques and Support Vector Machines (SVM) to improve the performance of fall detection classifiers [7]. Our approach attempts to improve the specificity even while only using one sensor device, a waist-worn tri-axial accelerometer.

A block diagram describing the whole data processing scheme can be seen in Figure 1. Key steps in the process include real time testing of whether the magnitude of accel-eration at any given moment is greater than a specified threshold. Each time the threshold is exceeded, an interval of the signal is instantiated and queued. A third order Discrete Wavelet Transform (DWT) transform is applied to each interval in the queue, and features are extracted. The extracted features are passed to the classifiers.
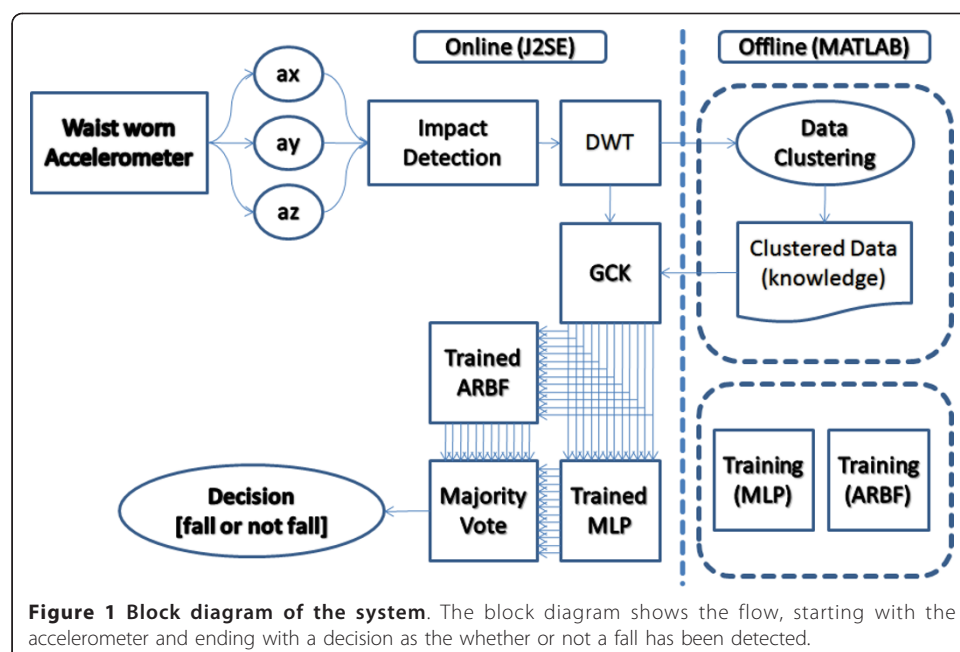

Figure 1 Block diagram of the system. The block diagram shows the flow, starting with the accelerometer and ending with a decision as the whether or not a fall has been detected.

The feature extraction stage makes use a custom method we refer to as Gaussian distribution of Clustered Knowledge (GCK) signal generation. Clustering is done with Regrouping Particle Swarm Optimization (RegPSO). Classification is undertaken by a newly developed "Augmented Radial Basis Function" (ARBF) neural network [12] alongside a multilayer perceptron (MLP).
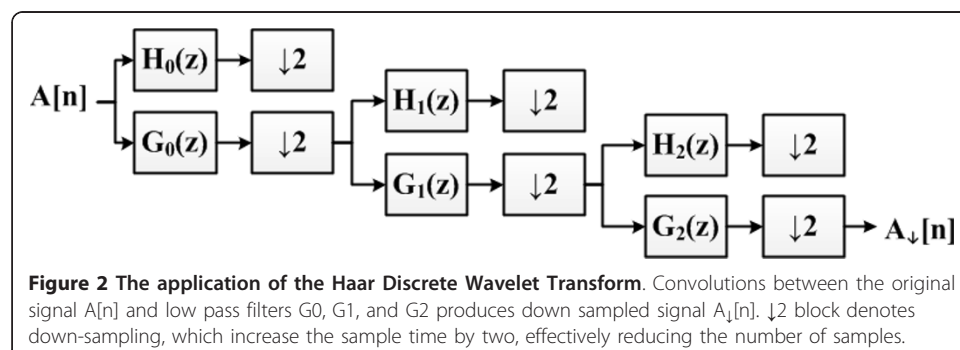
The input data consists of sampled acceleration values in three dimensions. The main accelerometer module used is the RD-3152 MMA7260Q - Zstar2 from Freescale Semiconductor. The module provides 3-axis acceleration data using an MMA7260Q accelerometer set to ± 6 g sensitivity range. Wireless communication is established using a ZigBee protocol 2.4 GHz band to communicate with the receiver board [13]. The accelerometer sensor is placed inside the right pocket of a vest.

At the receiving end of the wireless link, the samples are handled in real time using Java2SE and Matlab. Each signal has a length of 5 seconds sampled with 20Hz sampling frequency. Signals are divided into 2 classes: fall signals and Activities of Daily Living (ADL) signals.

Data preprocessing is undertaken in three steps: uncommon acceleration detection, normalization, and data filtering.

- Uncommon acceleration detection: uncommon accelerations, regardless of whether they are due to falls or not, can be observed when the magnitude of acceleration is above a specified threshold. When an uncommon acceleration occurs at time τ, a window is constructed at $\tau \pm 2.5 \ s$ and acceleration data in that window is pushed to the classification queue.
- Normalization: raw acceleration data $\vec{a}(t)$ has an offset due to the static force of gravity that differs depending on orientation of the accelerometer. Acceleration signals are normalized by subtracting every sample from $\vec{a}(t = 0)$.
- Data Filtering: the DWT decomposes discrete time signals using a digital filter approach. The DWT computes successive convolutions between input signal with discrete low pass and high pass filters [14]. The application of this filter in the system can be seen in Figure 2. The DWT filters the acceleration signal and down-samples it up to the third order using Haar wavelets. The intention is to reduce the signal complexity while still providing sufficient relevant information to the classifiers.

Previous work suggests that K-means Particle Swarm Optimization (PSO) can be a reliable tool for data clustering [15]. The approach was originally introduced by

Figure 2 The application of the Haar Discrete Wavelet Transform. Convolutions between the original signal A[n] and low pass filters G0, G1, and G2 produces down sampled signal $A_{\downarrow}$[n]. $\downarrow$2 block denotes down-sampling, which increase the sample time by two, effectively reducing the number of samples.

Kennedy and Eberhart [16]. It has two base models: Local Best (*lbest*) PSO and Global Best (*gbest*) PSO. Our approach makes use of the *gbest* PSO method. Each particle has x_i: *current coordinate*, v_i: *current velocity*, and p_i: *personal best coordinate*. The Regrouping Particle Swarm Optimization (RegPSO) approach was proposed by Evers and Ghalia in 2009. RegPSO is designed to remedy premature convergence and stagnation due to local minima problems [17]. We use a RegPSO method to cluster N vectors $\vec{z}$ in the dataset S (1). The clustering algorithm is given in Figure 3.

The new GCK method was inspired by Monte Carlo approaches. The GCK method is intended to help classify clustered patterns of statistical characteristics. It refers incoming input signals to cluster centroids, and multiplexes them based on the Gaussian characteristics of the clusters. Each input signal $\bar{\gamma}$ is queried against the cluster centroids and passed through a Radial Basis Kernel (25) to get the rate of membership $\theta_i(\bar{\gamma})$. The cluster with the highest rate of membership, Cluster *I*, is selected as the GCK seed. A knowledge signal $\bar{\gamma}$ is obtained by generating a vector of Gaussian random numbers with mean $\bar{\mu}_I$ and standard deviation $\bar{\sigma}_I$ (26). The generated GCK signal $\bar{\gamma}$ is fused with $\bar{\gamma}$ using a significance ratio of *A:B* (typically 0.8:0.2) to create signal $\bar{\vartheta}(\bar{\gamma})$ (27).

$$\theta_i(\bar{\gamma}) = e^{\dfrac{\|\bar{\gamma} - \mu_i\|}{2\sigma_i{}^2}} \tag{25}$$

$$\bar{\gamma}(\bar{\gamma}) = N(\bar{\mu}_I, \bar{\sigma}_I), I \in \max \theta_i(\bar{\gamma}) \,|\, 1 \le i \le N_c \tag{26}$$

$$\bar{\vartheta}(\bar{\gamma}) = \frac{A\bar{\gamma} + B\bar{\gamma}(\bar{\gamma})}{A + B} \tag{27}$$

Augmented Radial Basis Function neural networks (ARBF) have previously been used in time signal classification of head movement patterns, with promising results [12]. ARBF consists of an RBF layer and an MLP augmentation layer, shown in Figure 4. ARBF is reported to have a sensitivity advantage over conventional RBF and a specificity advantage over MLP [12].

The ARBF function uses a Gaussian radial basis kernel. It can be described as a K-dimensional Gaussian distribution, where K is the number of dimensions of the input. The output of the RBF layer is a vector where μ_n and σ_n correspond to cluster centroids and the standard deviation of each RBF node. The RBF centroids are optimized using RegPSO. The MLP layer uses a sigmoid kernel in the hidden layer and a linear kernel in the output layer. No normalization method is required at this stage because the RBF layer has already normalized the input signals from 0 to 1. The MLP layer is trained with resilient back-propagation. This combination of MLP and ARBF was used because of each classifier's different characteristics. MLP networks tend to perform better in global generalization, while RBF-kernel based classifiers such as ARBF tend to perform better in local generalization [12]. The ensemble receives the collection of signals consisting of the original signal $\bar{\gamma}$ and N GCK-Fused signals $\bar{\vartheta}$, and each neural network outputs $N + 1$ classifications of the input vectors. The outputs are then combined based on majority vote.

S : Dataset
$\vec{z}$: DWT-Filtered acceleration vector
K : Data dimension (length of vector $\vec{z}$)
$\vec{C}$: Cluster centroid
N_c : Number of centroids
N_p : Number of particle
x : Particle coordinate
$d(\vec{x}, \vec{y})$: Euclidian distance between vector x and vector y
$f(x)$: Fitness function
v : Particle velocity
p : Particle personal best coordinate
g : Swarm global best coordinate
r : Regroup counter
Ω^r : Swarm search space at regroup count r
z^U : Data upper bound
z^L : Data lower bound
ε : Stagnation threshold (1.1e-4)
ρ : Regrouping coefficient (5/6ε)
v_{max}^r : Velocity clamp at regroup count r
λ : Dampening proportion constant
rnd : Uniform random vector
c_1 : Cognitive constant
c_2 : Social constant
γ : Inertia weight decay rate constant
δ : Normalized swarm radius (particle distance to global best)

Definitions:

$$S = \begin{bmatrix} \vec{z}_1 & \vec{z}_2 & \vec{z}_n & \cdots & \vec{z}_N \end{bmatrix} \tag{1}$$

$$\vec{z} = \begin{bmatrix} z_1 & z_2 & z_k & \cdots & z_K \end{bmatrix}^T \tag{2}$$

$$\vec{C}_j = \begin{bmatrix} C_{j1} & C_{j2} & C_{jk} & \cdots & C_{jK} \end{bmatrix}^T \tag{3}$$

$$x_i = \begin{bmatrix} \vec{C}_{i1} & \vec{C}_{i2} & \vec{C}_{ij} & \cdots & \vec{C}_{iNc} \end{bmatrix} \tag{4}$$

$$f(x_i) = \sum_{j=1}^{N_c} \sum_{\forall \vec{z}_p \in \vec{C}_{ij}} d(\vec{z}_p, \vec{C}_{ij}) \tag{5}$$

Algorithm g = RegPSO_Cluster($S, N_c, N_p, \gamma, c_1, c_2, \lambda, \varepsilon$)

  Seed swarm with k-means
$$p_i(0) = x_i = kmeans(S), v_i = \vec{0}, \forall i : 1 \le i \le N_p \tag{6}$$
$$g(0) = p_i, p_i \in \max(f(p_i)) \tag{7}$$

  Define prior search space
$$\Omega^0 = \begin{bmatrix} z_1^U & z_k^U & \cdots & z_K^U \\ z_1^L & z_k^L & \cdots & z_K^L \end{bmatrix} \tag{8}$$

  Calculate prior search space range
$$range_k(\Omega^0) = \min\left(\left| z_k^U - z_k^L \right|, \rho \max_{1 \le i \le Np} \left(\left| x_i(0)_k - g(0)_k \right| \right) \right) \tag{9}$$

  Calculate prior velocity clamp
$$v_{max}^0 = \lambda range(\Omega^0) \tag{10}$$

  While stopping criterion is not satisfied

    Decay the inertia weight
$$w(t+1) = \gamma w(t) \tag{11}$$

    For each particle i, $1 \le i \le N_p$

      Update velocity and position
$$v_i(t+1) = w(t)v_i(t) + c_1 rnd_1(p_i(t) - x_i(t)) + c_2 rnd_2(g(t) - x_i(t)) \tag{12}$$
$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{13}$$
$$clamp(x_i, \Omega^r) ; clamp(v_i, \pm v_{max}^r) \tag{14}$$

      Update personal best and global best coordinates
$$p_i(t) = \begin{cases} x_i(t+1) & if \quad f(x_i(t+1)) > f(p_i(t)) \\ p_i(t) & otherwise \end{cases} \tag{15}$$
$$g(t) = \begin{cases} x_i(t+1) & if \quad f(x_i(t+1)) > f(g(t)) \\ g(t) & otherwise \end{cases} \tag{16}$$

      Calculate swarm radius
$$\delta_i = \frac{\| x_i(t+1) - g(t+1) \|}{\| range(\Omega^r) \|} \tag{17}$$
    End For

    If premature convergence detected
    If $\max(\delta_i) < \varepsilon, \forall i : 1 \le i \le N_p$ (18)

      Increment regroup count and calculate search space range for next regroup
$$range_k(\Omega^{r+1}) = \min\left(\left| g(t)_k^U - g(t)_k^L \right|, \rho \max_{1 \le i \le Np} \left(\left| x_i(t)_k - g(t)_k \right| \right) \right) \tag{19}$$

      Redefine search space
$$\Omega^{r+1} = \begin{bmatrix} g(t)_1^U & g(t)_2^U & \cdots & g(t)_K^U \\ g(t)_1^L & g(t)_2^L & \cdots & g(t)_K^L \end{bmatrix} + \frac{1}{2}\begin{bmatrix} range_k(\Omega^{r+1}) \\ -range_k(\Omega^{r+1}) \end{bmatrix} \tag{20}$$

      Redefine velocity clamp for next regroup
$$v_{max}^{r+1} = \lambda range(\Omega^{r+1}) \tag{21}$$

      For each particle i, $1 \le i \le N_p$

        Regroup particle inside the search space around the global best coordinate
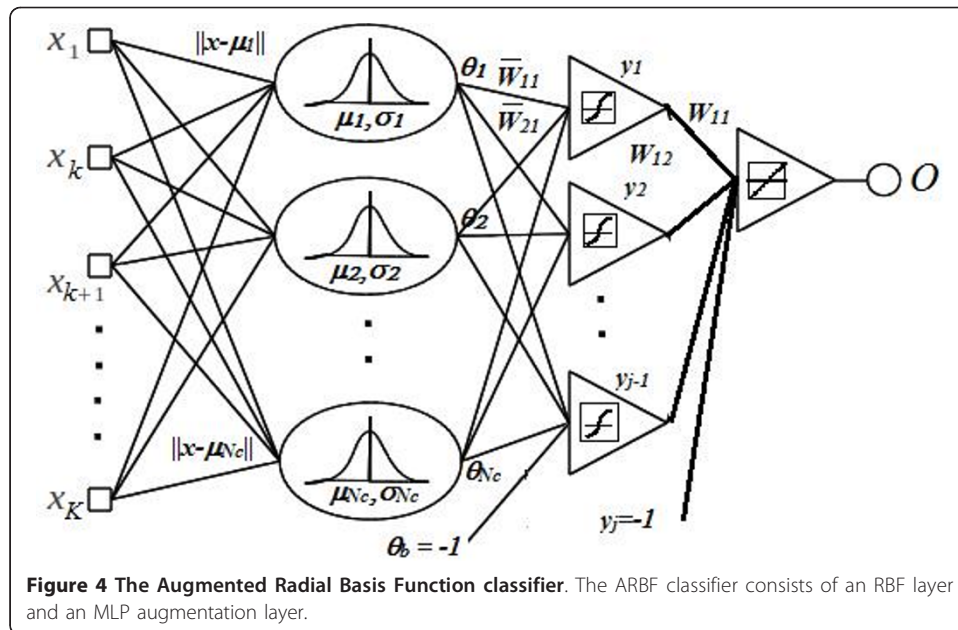$$x_i(t) = g(t) + rand([0,1]) \circ range(\Omega^{r+1}) - 0.5 range(\Omega^{r+1}) \tag{22}$$

        Assign the farthest datapoint as the next centroid candidate
$$\vec{C}_{iNc+1} = \vec{z}_{farthest} \mid \vec{z}_{farthest} \in \max_{\forall \vec{z}_p \in x_{ic}(t)} d(\vec{z}_p, x_{ig}(t)) \tag{23}$$
$$x_i = \begin{bmatrix} \vec{C}_{i1} & \vec{C}_{ij} & \cdots & \vec{C}_{iNc} & \vec{C}_{iNc+1} \end{bmatrix} \tag{24}$$
      End For
    End If
  End While

  Return the global best coordinate
  Return g

End Algorithm

Figure 3 The clustering algorithm. A RegPSO method is used to cluster N vectors $\vec{z}$ in the dataset S (1).

Figure 4 The Augmented Radial Basis Function classifier. The ARBF classifier consists of an RBF layer and an MLP augmentation layer.

A brief summary of the steps taken is as follows:

1. Pre-process the data;

2. Create the clustered-knowledge database using RegPSO;

3. Separate the In-group fall data and ADL data into training and validation sets with ratio of 4:1;

4. Train MLP using resilient back propagation;

5. Train ARBF;

6. Create an RBF layer with cluster centroids taken from the clustered-knowledge result from RegPSO;

7. Pass the pre-processed data to the RBF layer;

8. Pass the output of the RBF layer to MLP layer;

9. Train the MLP with resilient back propagation;

10. Merge the RBF layer with the MLP layer.

Table 1 provides a description of each set of data. The project was conducted in compliance with the Helsinki Declaration, and in accordance with the University of Technology Sydney (UTS) research guidelines and clearance granted by the UTS Human Research Ethics Committee.

Results and Discussion

An example of the output from the Haar DWT third order filtering processing stage is given in Figure 5 The original signal A[n] is shown on the left, and the processed signal $A_{\downarrow}[n]$ is shown on the right. It can be seen that the processed signal appears to have reduced complexity, but still retains the essential features. Figure 6 shows 150 pre-processed fall signals stacked together. Figure 7 provides an example of the progression though the first 250 iterations of the classification algorithm. Figure 8 shows a fall signal, and Figure 9 shows an ADL sit down signal.

Table 2 shows the results comparing ARB + MLP combined with ARB alone and MLP alone, where the number of GCK signals equals 5. The table shows that the
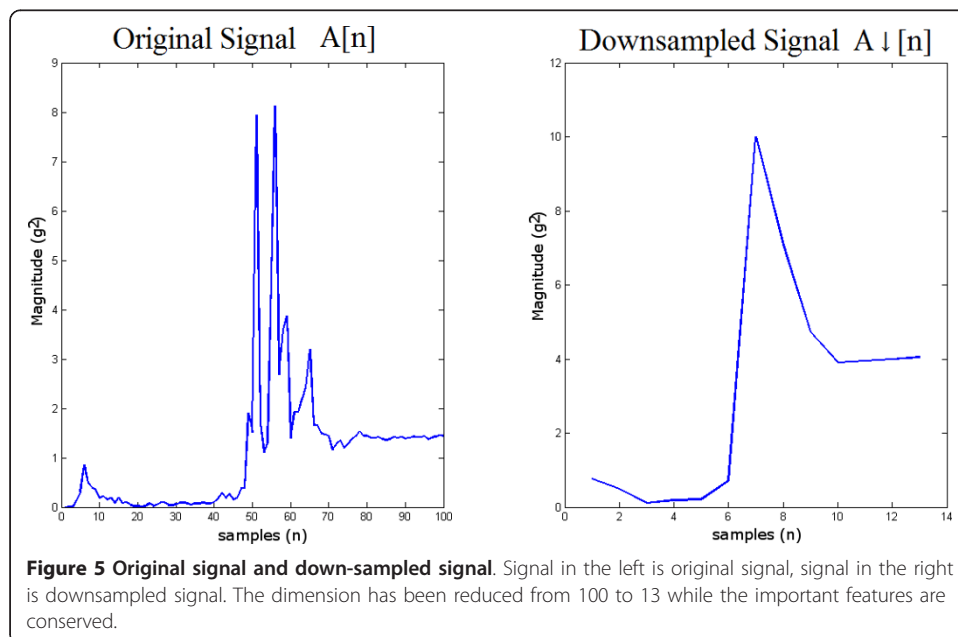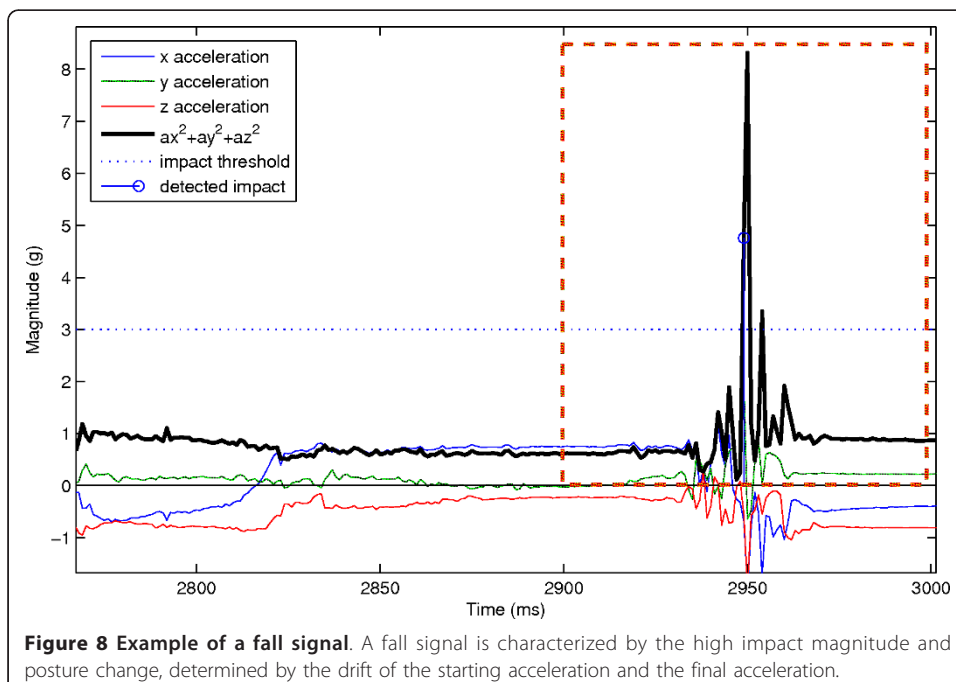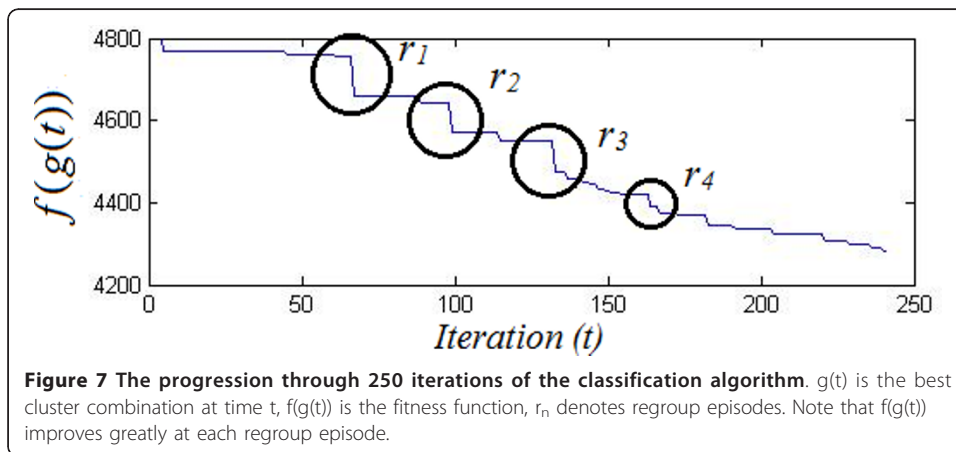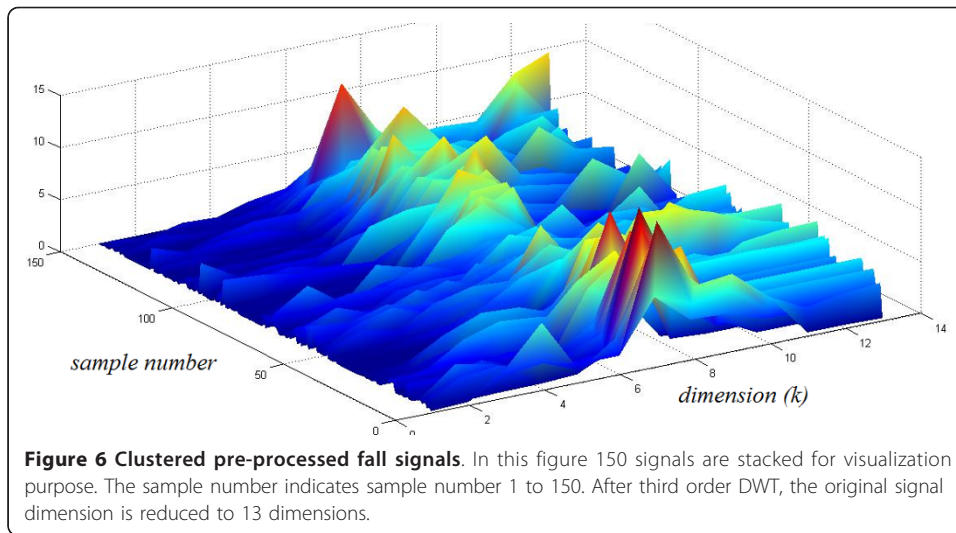
Table 1 The data

Data	Participants	Signals
In-group fall data	Collected from 5 healthy volunteers, 2 females and 3 males. Volunteers aged between 19 and 28 years.	293 fall signals were collected. Of these, 153 signals were used for training, and 140 signals used for testing (in-group performance)
Out-group fall data	Collected from 3 different healthy male volunteers whose data was not included in the training data. Volunteers aged between 19 and 28 years.	This set included 85 signals, all used to test "out-group" performance. The term "out-group" is used to indicated that these people's data was not used as training data.
The Activities of Daily Living (ADL) training data	Collected from 3 people. A total of 8 hours of ADL data was collected in a home environment. An additional hour of exercise data was recorded from 2 people in a gym environment. Volunteers aged between 19 and 28 years.	1831 ADL signals were collected. 1000 randomly selected ADL signals were used for the training set while 831 were used for testing. Of the 1000 randomly selected signals used for training, 750 related to ADL routine, and 250 related to ADL exercise. Of the 831 signals used for testing, 400 related to ADL routine and 381 related to ADL gym exercise.
Validation set		Taken randomly from the training set with the ratio of training versus validation = 4:1.

combined classifier offers the greatest sensitivity, with a slight reduction in specificity for routine ADL and an increased specificity for exercise activities. Figure 10 shows the sensitivity and specificity of each approach where GCK = 5. This number was selected because trials indicated that sensitivity improves up to GCK = 5 but stays the same at greater than 5, while specificity of the system decreases when GCK greater than 5 is used. The GCK effects to the classifier performance can be seen in Figure 11.

In preliminary tests, the approach achieves 100% sensitivity on in-group falls, 97.65% on out-group falls, 99.33% specificity on routine ADL, and 96.59% specificity on exercise ADL.

Limitations applicable to these results include the following. First, the number of subjects is relatively small. Second, the method for acquiring the falls data did not
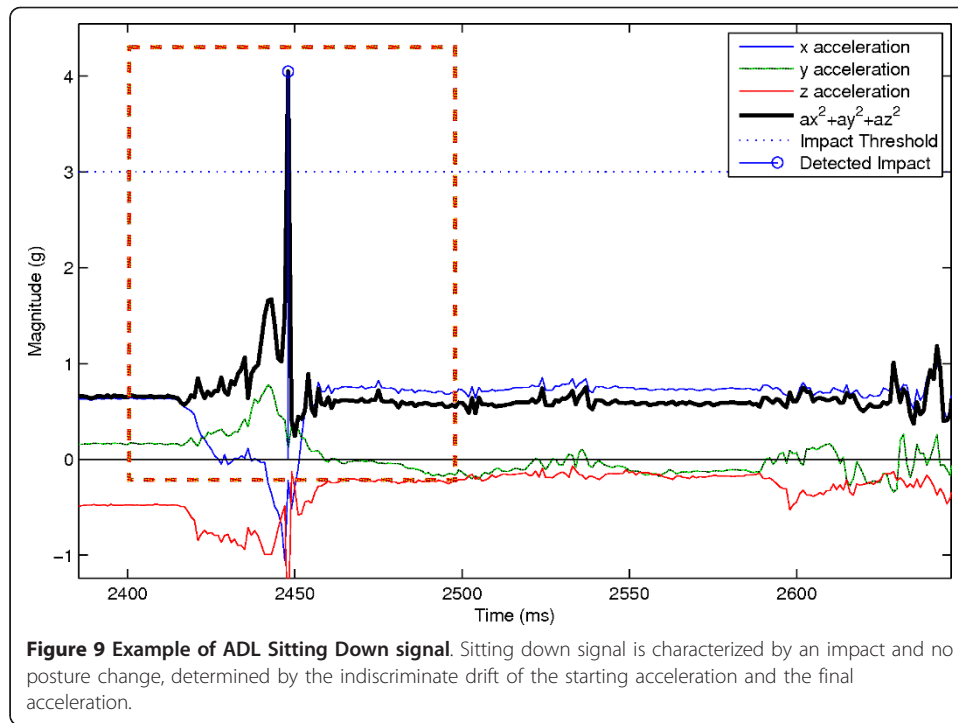

Figure 5 Original signal and down-sampled signal. Signal in the left is original signal, signal in the right is downsampled signal. The dimension has been reduced from 100 to 13 while the important features are conserved.

Figure 6 Clustered pre-processed fall signals. In this figure 150 signals are stacked for visualization purpose. The sample number indicates sample number 1 to 150. After third order DWT, the original signal dimension is reduced to 13 dimensions.

Figure 7 The progression through 250 iterations of the classification algorithm. g(t) is the best cluster combination at time t, f(g(t)) is the fitness function, r_n denotes regroup episodes. Note that f(g(t)) improves greatly at each regroup episode.

Figure 8 Example of a fall signal. A fall signal is characterized by the high impact magnitude and posture change, determined by the drift of the starting acceleration and the final acceleration.

Figure 9 Example of ADL Sitting Down signal. Sitting down signal is characterized by an impact and no posture change, determined by the indiscriminate drift of the starting acceleration and the final acceleration.

Table 2 Comparison of the results for the three classifier method
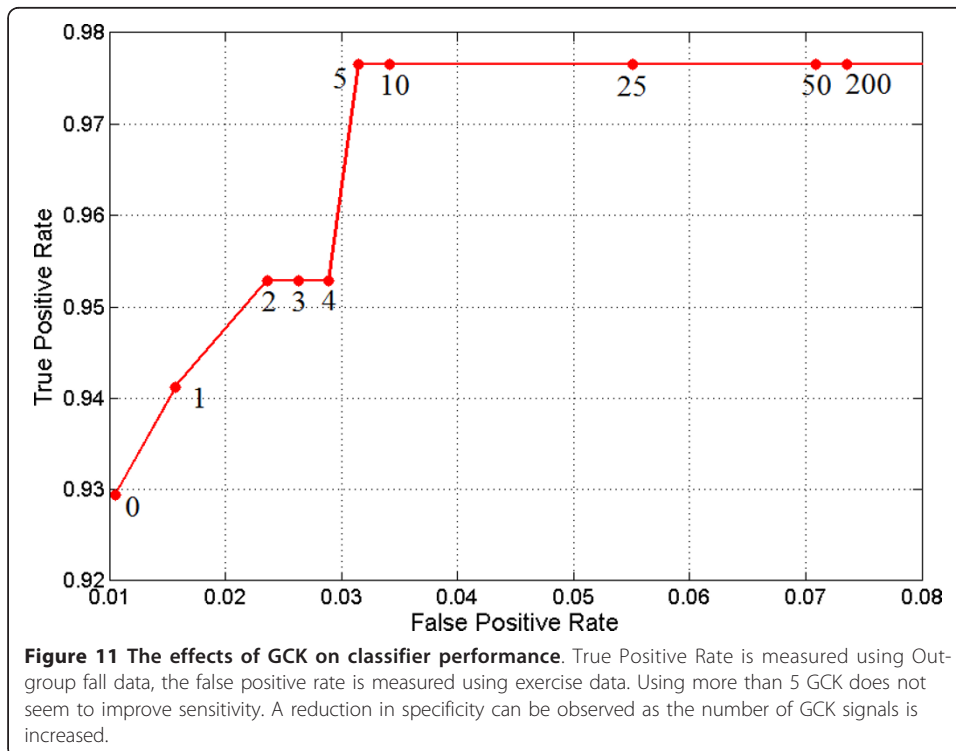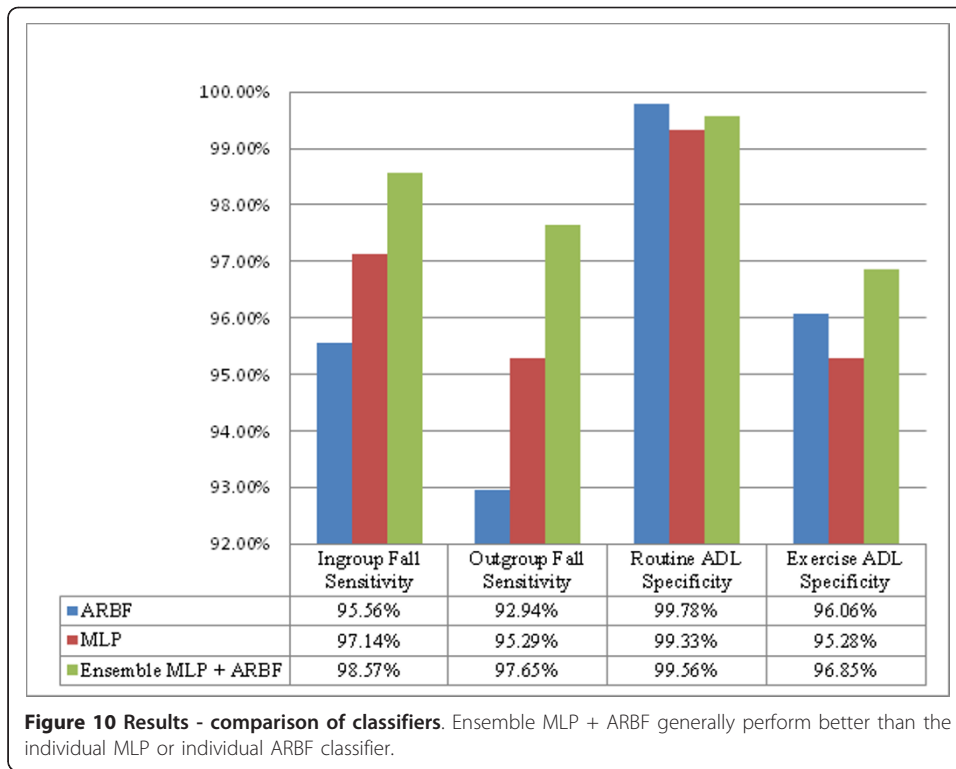
Classifier Scheme	Ingroup Fall Sensitivity (N = 140)	Outgroup Fall Sensitivity (N = 85)	Routine ADL Specificity (N = 450)	Exercise ADL Specificity (N = 381)
ARBF	95.56%	92.94%	99.78%	96.06%
MLP	97.14%	95.29%	99.33%	95.28%
Ensemble MLP + ARBF	98.57%	97.65%	99.56%	96.85%

include truly accidental falls - the falls data was acquired when subjects deliberately fell onto a mattress. Third, the ADL data included only activities around the home and activities in the gym - data relating to ordinary work, transportation and other non-home activities would likely be more representative of some people's typical daily activities. Fourth, the data was acquired from people aged 19-28 years - it would be preferable for future work to include people from older age brackets.

These results are an exploratory step towards gaining knowledge about potential elements of a fall detection system. The implications of the results are somewhat limited due to limitations of the data acquisition processes. Notwithstanding, the results suggest the methods described here warrant further development and experimental investigation.

A further implication of these results is that some of the methods described here may also be applicable for body movement analysis and gait analysis relating to conditions that affect balance such as Parkinson's disease.

Future research will include acquiring data from different age groups, and developing methods to make use of data from ambulatory devices that include gyroscopes and a magnetometer.

Figure 10 Results - comparison of classifiers. Ensemble MLP + ARBF generally perform better than the individual MLP or individual ARBF classifier.

	Ingroup Fall Sensitivity	Outgroup Fall Sensitivity	Routine ADL Specificity	Exercise ADL Specificity
ARBF	95.56%	92.94%	99.78%	96.06%
MLP	97.14%	95.29%	99.33%	95.28%
Ensemble MLP + ARBF	98.57%	97.65%	99.56%	96.85%

Figure 11 The effects of GCK on classifier performance. True Positive Rate is measured using Out-group fall data, the false positive rate is measured using exercise data. Using more than 5 GCK does not seem to improve sensitivity. A reduction in specificity can be observed as the number of GCK signals is increased.

Conclusions

Preliminary testing suggests the methods described here are noteworthy particularly for researchers who are attempting to improve the performance of ambulatory fall-detection systems. The methods should also be of interest for researchers who use (or are considering using) accelerometers to measure body movement. The pre-processing and feature-extraction steps appear to simplify the signal while successfully extracting the essential features that are required to characterize a fall. The results suggest that the approach used here performs better than MLP alone.

Author details
[1]Faculty of Engineering and IT, University of Technology Sydney, NSW, 2007, Australia. [2]ICT Centre, CSIRO, Marsfield, NSW, 2122, Australia.

Authors' contributions
MY carried out the research. MY and BM drafted the manuscript. BM, SS, BC and HN supervise the research. All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

References
1. World Health Organization: WHO Global Report on Falls Prevention in Older Age. *France* 2007, **ch.1**.
2. Barroso MC, Esteves GP, Nunes TP, MG Silva L, CD Faria A, Melo PL: A telemedicine instrument for remote evaluation of tremor: design and initial applications in fatigue and patients with Parkinson's Disease. *BioMedical Engineering OnLine* 2011, **10**(14), doi:10.1186/1475-925X-10-14.
3. Moore ST, Dilda V, Hakim B, MacDougall HG: Validation of 24-hour ambulatory gait assessment in Parkinson's disease with simultaneous video observation. *BioMedical Engineering OnLine* 2011, **10**(82), doi:10.1186/1475-925X-10-82.
4. Kangas M, Konttila A, Winblad I, Jämsä T: Determination of simple thresholds for accelerometry-based parameters for fall detection. *Proc of the 29th Annual International Conference of the IEEE EMBS, Lyon* 2007, 1367-1370.
5. Chen J, Kwong K, Chang D, Luk J, Bajcsy R: Wearable Sensors for Reliable Fall Detection. *Proc of the 27th Annual Conference of the IEEE EMBS, Shanghai* 2005, 3551-3554.
6. Bourke AK, VD Ven P, Gamble M, O'Connor R, Murphy K, Bogan E, McQuade E, Finucane P, ÓLaighin G, Nelson J: Assessment of Waist-worn Tri-Axial Accelerometer Based Fall-detection Algorithm using Continuous Unsupervised Activities. *Proc of the 32nd Annual Conference of the IEEE EMBS, Buenos Aires* 2010, 2782-2785.
7. Shi G, Chan CS, Li WJ, Leung KS, Zou Y, Jin Y: Mobile Human Airbag System for Fall Protection Using MEMS Sensor and Embedded SVM Classifier. *IEEE Sensors Journal* 2009, **9**(5):495-503.
8. Su SW, Chen W, Liu D, Kuang W, Yu X, Guo T, Celler BG, Nguyen HT: Dynamic Modelling of Heart Rate Response Under Different Exercise Intensity. *The Open Medical Informatics Journal* 2011, **4**:81-85.
9. Moulton B, Croucher G, Varis A, Chen J: Method for increasing the energy efficiency of wirelessly networked ambulatory health monitoring devices. *Journal of Convergence Information Technology* 2009, **5**(1):7-14.
10. Moulton B, Chaczko Z, Karatovic M: Data fusion and aggregation methods for pre-processing ambulatory monitoring and remote sensor data for upload to personal electronic health records. *International Journal of Digital Content Technology and its Applications* 2009, **3**(4):120-127.
11. Moulton B, Chaczko Z, Karatovic M: Updating electronic health records with information from sensor systems: considerations relating to standards and architecture arising from the development of a prototype system. *Journal of Convergence Information Technology* 2009, **4**(4):21-26.
12. Yuwono M, Handojoseno AMA, Nguyen HT: Optimization of Head Movement Recognition Using Augmented Radial Basis Function Neural Network. *Proc of the 33rd Annual International Conference of the IEEE EMBS, Boston* 2011, 2776-2779.
13. Lajšner P, Kozub R: Using the MMA7360L ZSTAR2 Demo Board. *Freescale Semiconductor* 2007.
14. Rioul O, Duhamel P: Fast Algorithm for Discrete and Continuous Wavelet Transforms. *IEEE trans. on Information Theory* 1992, **38**(2):569-586.
15. Merwe DW, Engelbrecht AP: Data Clustering using Particle Swarm Optimization. *Congress on Evolutionary Computation* 2003, **1**:215-220.
16. Eberhart R, Kennedy J: A New Optimizer Using Particle Swarm Theory. *IEEE Sixth International Symposium on Micro Machine and Human Science* 1995, 39-43.
17. Evers GI, Ghalia MB: Regrouping Particle Swarm Optimization: A New Global Optimization Algorithm with Improved Performance Consistency Across Benchmarks. *International Conference on Systems, Man, and Cybernetics 2009, San Antonio, TX, USA* 2009, 3901-3908.