

“© 2011 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

Fall Detection using a Gaussian Distribution of Clustered Knowledge, Augmented Radial Basis Neural-Network, and Multilayer Perceptron

Mitchell Yuwono, Steven W. Su, Bruce Moulton, *Member, IEEE*

Abstract—The rapidly increasing population of elderly people has posed a big challenge to research in fall prevention and detection. Substantial amounts of injuries, disabilities, traumas and deaths among elderly people due to falls have been reported worldwide. There is therefore a need for a reliable, simple, and affordable automatic fall detection system. This paper proposes a reliable fall detection algorithm using minimal information from a single waist worn wireless tri-axial accelerometer. The method proposed is to approach fall detection using digital signal processing and neural networks. This method includes the application of Discrete Wavelet Transform (DWT), Regrouping Particle Swarm Optimization (RegPSO), a proposed method called Gaussian Distribution of Clustered Knowledge (GCK), and an Ensemble of Classifiers using two different classifiers: Multilayer Perceptron Neural Network (MLP) and Augmented Radial Basis Neural Networks (ARBF). The proposed method has been tested on 8 healthy individuals in a home environment and yields promising result of up to 100% sensitivity on ingroup, 97.65% sensitivity on outgroup, and 99.56% specificity on Activities of Daily Living (ADL) data.

Index Terms—Fall Detection; Discrete Wavelet Transform; Regrouping Particle Swarm Optimization; Gaussian Distribution of Clustered Knowledge; Ensemble of Classifiers; Augmented Radial Basis Neural Networks;

I. INTRODUCTION

According to recent report of WHO, falls has been one of the prominent causes of fatalities in elderly people. Rate of fall related injuries of people over 60 years of age in Western Australia and United Kingdom reaches up to 8.9 per 10,000 population. Fall fatality rate for people of age 65 and above reaches up to 36.8 per 100,000 population. It is estimated that in 2030 fall related injuries will increase by 100% should there be no preventive approaches taken [1].

Recently, a substantial amount of studies on accelerometer-based fall detection has been conducted [2-5]. Kangas' experiment shows that waist and head acceleration measurements provide more distinctive information regarding falls [2]. A popular approach to fall detection is thresholding approaches [2-4]. However, it is argued that these approaches

often result in false alarms [5]. Identifying this issue, Shi combined threshold techniques and SVM to determine lateral falls in his research [5].

The objective in this paper is to optimize the performance of fall detection system using a Neural-Network approach using minimal information from a waist-worn tri-axial accelerometer. DWT and RegPSO will be used to assist in the feature extraction and GCK signal generation. A newly proposed classifier, ARBF, will also be used alongside MLP.

This paper is structured as follows. Section II gives a brief overview of the system and algorithm used in the experiment. Section III provides data preprocessing method. Section IV explains the proposed classification scheme. Section V explains data collection. Section VI provides the results and discussions. Finally Section VII concludes this paper.

II. OVERVIEW

A. System

Input data for the system are sampled 3-D acceleration signals. The accelerometer module used in this project is RD-3152 MMA7260Q – Zstar2 from Freescale Semiconductor. It provides 3-axis acceleration readings using an MMA7260Q accelerometer set to $\pm 6g$ sensitivity range. The wireless communication is based on ZigBee protocol 2.4GHz band to communicate to the receiver board [6]. The accelerometer sensor is put inside the right pocket of a vest.

Data collection is done in real time using Java2SE connected directly to 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.

B. Algorithm

The proposed algorithm can be explained briefly as follows. Firstly the magnitude of acceleration is observed. If the magnitude is greater than a specified threshold, a window is instantiated and signal in the window is pushed to the classification queue. For each signal in the queue, third order DWT filter is applied and N GCK signals are generated. These signals are queried against an ensemble of classifiers.

III. DATA PREPROCESSING

Especially because there is no other source of information other than accelerometer, the method used in this paper relies greatly on quality data preprocessing. The proposed data preprocessing method can be divided into three major steps: impact detection, normalization, and data filtering.

A. Impact Detection

From the observation of the collected data of falls and daily life activities including exercises and trips, acceleration magnitudes of normal activities are generally lower than those of exercises and falls. Uncommon activities thus can be observed when the magnitude of acceleration is above a specified threshold. There is a chance that this uncommon activity may necessarily be an impact.

At any instance of uncommon activity that occurs at time τ , a window is constructed at $\tau \pm 2.5s$ and acceleration data in that window is copied. The data is pushed to the classification queue and undergoes the next steps in the algorithm.

B. Relative Start Scheme

In order to effectively supply information relative to the starting acceleration, a normalization scheme called the relative start scheme is used. The relative start scheme is a simple normalization method where acceleration signals are normalized by subtracting every sample with $\bar{a}(t = 0)$.

Essentially, raw acceleration data transferred by the accelerometer $\bar{a}(t)$ have an offset due to the static force of gravity. This offset differs depending on orientation of the accelerometer.

One of the qualities of a fall is the change of body posture from standing to lying. The starting accelerometer orientation in these cases is different to the post-fall orientation. This phenomenon can be observed in the drift of acceleration offset

before and after fall. The drift of acceleration offset is one of the qualities to describe positive falls.

C. Data Filtering using Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT) is a technique to decompose discrete time signals by using a digital filter approach. DWT is computed by successive convolutions between input signal with discrete low pass and high pass filters [7]. The application of this filter in the system can be seen in Fig. 1.

In this application DWT is used to filter the acceleration signal and down-sampling it up to the third order using Haar wavelets. The intention behind using the DWT is to reduce the signal complexity and thus increase generalization capability of the classifier. The down sampled signal should provide essential information required for the classifier.

IV. CLASSIFICATION SYSTEM

A. Data Clustering using K-Means seeded Regrouping Particle Swarm

Particle Swarm Optimization (PSO) seeded with K-means has been used as a reliable tool for data clustering [8]. It was originally introduced by Kennedy and Eberhart [9]. It has two base models: Local Best (*lbest*) PSO and Global Best (*gbest*) PSO. This paper utilizes the *gbest* PSO. Each particle in contains x_i : *current coordinate*, v_i : *current velocity*, and p_i : *personal best coordinate*.

The Regrouping Particle Swarm Optimization (RegPSO) is proposed by Evers and Ghalia in 2009. RegPSO is designed to remedy premature convergence and stagnation due to local minima problems [10]. In this proposed method RegPSO is used to cluster N vectors \vec{z} in the dataset S (1). \vec{z} in this case is the filtered acceleration signal (2). K denotes the dimension of the data, which in this particular application is the number of discrete samples of the DWT filtered acceleration signal. A

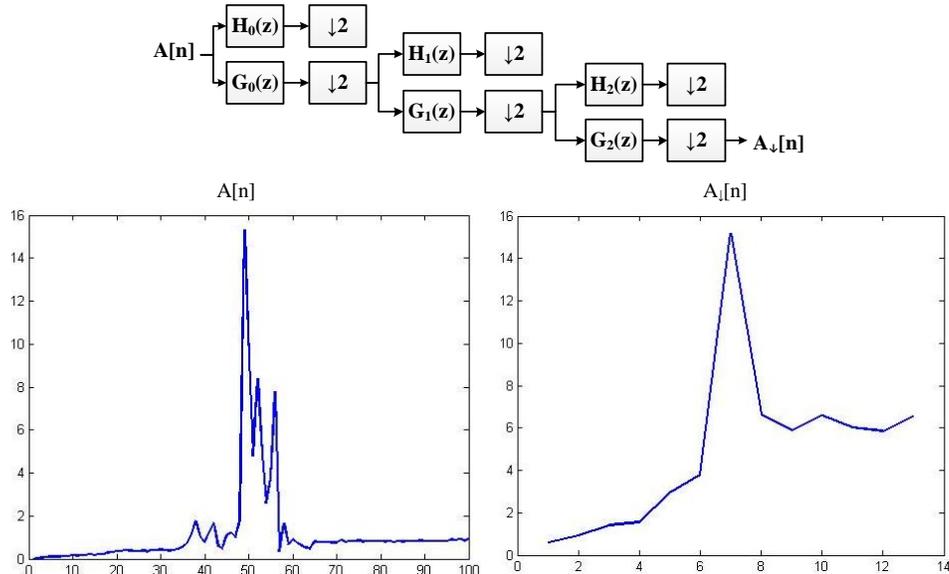


Fig. 1. Third order filtering of Acceleration signal using DWT with Haar Wavelets. Convolutions between the original signal $A[n]$ and low pass filters G_1, G_1 and G_3 produces down-sampled signal $A_1[n]$. $\downarrow 2$ block denotes down-sampling, which increase the sample time by two accordingly to Nyquist theorem.

cluster is represented by a centroid \bar{c}_j (3). The goal of this method is to optimize \bar{c}_j such that the fitness function in (5) is minimum. Each particle x therefore represents an optimum set of centroid candidates as seen in (4) where N_c is the predefined number of centroids. The fitness function is measured by calculating the sum of Euclidian distances between cluster elements z_p to its specified cluster centroid for each cluster (5).

$$S = [\bar{z}_1 \quad \bar{z}_2 \quad \bar{z}_n \quad \dots \quad \bar{z}_N] \quad (1)$$

$$\bar{z} = [z_1 \quad z_2 \quad z_k \quad \dots \quad z_K]^T \quad (2)$$

$$\bar{C}_j = [C_{j1} \quad C_{j2} \quad C_{jk} \quad \dots \quad C_{jK}]^T \quad (3)$$

$$x_i = [\bar{c}_{i1} \quad \bar{c}_{i2} \quad \bar{c}_{ij} \quad \dots \quad \bar{c}_{iNc}] \quad (4)$$

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

The complete algorithm can be seen in Fig. 4. K-means is used to initialize all the N_p particles p_i in the swarm (6). The particle initialized with the best fitness is then selected as the global best particle $g(0)$ (7). The prior search space before regrouping Ω^0 is initialized by taking the upper bound z^U and lower bound z^L of each dimension k of the input dataset S (8). Initial search space range $range(\Omega^0)$ is calculated by taking the minimum of the data upper and lower bound and the largest distance of the swarm multiplied by a regrouping coefficient ρ (9). ρ is set to $5/6\varepsilon$, where the stagnation threshold ε is set to $1.1e-4$ as advised in [10]. Superscript 0

indicates that regroup count $r = 0$, which means no regrouping has occurred. The initial velocity clamp v_{\max}^0 is calculated as a proportion of λ of $range(\Omega^0)$ (10).

Velocity and position are updated every iteration depending on random numbers rnd_1 and rnd_2 , cognitive constant c_1 , social constant c_2 , and inertia weight w which is set to decay with a constant rate of γ (11-13)[8]. x_i and v_i are clamped inside Ω^r and v_{\max}^r respectively (14)[10]. Each particle in the swarm was queried with the fitness function in (5). p_i and g are updated as a better fitted particle is found (15-16).

Premature convergence is detected when the normalized swarm radius δ is lower than ε (17) [10]. When a premature convergence is detected, a regroup command is issued (18). On regroup, the regroup count r is incremented and the search space Ω^r and velocity clamp v_{\max}^r for the current regroup are recalculated (19-21). The new Ω^r is obtained by observing the global best's upper and lower bound added with the newly calculated search space range $range(\Omega^r)$ (19-20). The positions of the particles are then randomized around the current global best g (22) [10]. The number of centroid is incremented afterwards incremented (23-24). A new cluster centroid \bar{c}_{iNc+1} is chosen by selecting the data vector $\bar{z}_{farthest}$ which has the largest Euclidian distance to its cluster centroid $x_{ij}(t)$ (23). Note that every particle $x_i(t)$ is a collection of N_c centroid vectors (4). \bar{c}_{iNc+1} is then appended to the end of the specified particle (24). The clustering progress can be seen in Fig.2. The sample of the clustered data can be seen in Fig.3.

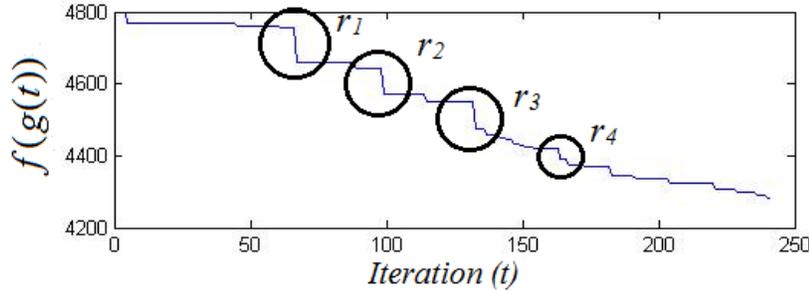


Fig.2. gbest fitness graph for the first 250 iterations, r denotes regroup

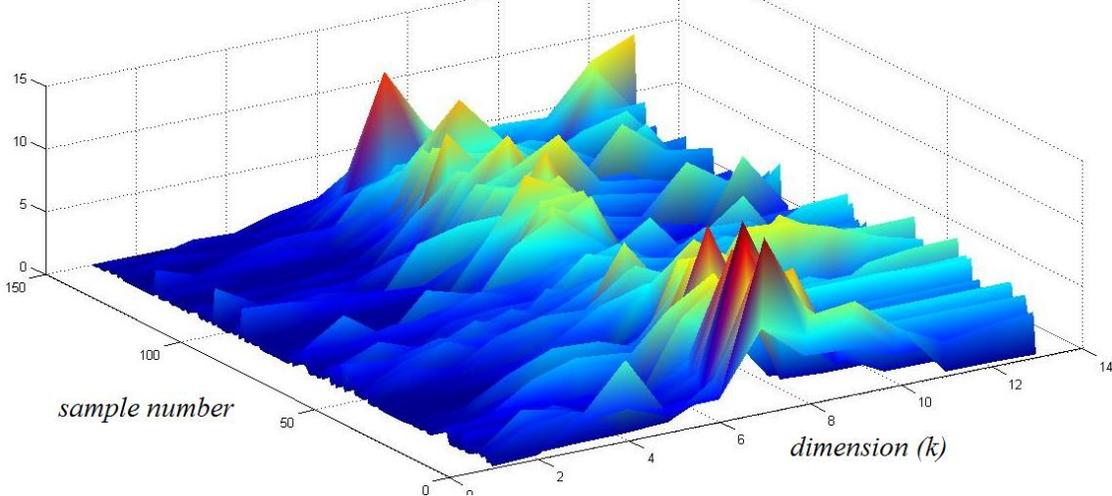


Fig.3. Clustered 150 fall signals

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

$$p_i(0) = x_i = \text{kmeans}(S), v_i = \vec{0}, \forall i: 1 \leq i \leq N_p \quad (6)$$

$$g(0) = p_i, p_i \in \max(f(p_i)) \quad (7)$$

$$\Omega^0 = \begin{bmatrix} z_1^U & z_k^U & \cdots & z_K^U \\ z_1^L & z_k^L & \cdots & z_K^L \end{bmatrix} \quad (8)$$

$$\text{range}_k(\Omega^0) = \min \left(\left| z_k^U - z_k^L \right|, \rho \max_{1 \leq i < N_p} (|x_i(0)_k - g(0)_k|) \right) \quad (9)$$

$$v_{\max}^0 = \lambda \text{range}(\Omega^0) \quad (10)$$

While stopping criterion is not satisfied

$$w(t+1) = \gamma w(t) \quad (11)$$

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

$$v_i(t+1) = w(t)v_i(t) + c_1 \text{rand}_1(p_i(t) - x_i(t)) + \quad (12)$$

$$c_2 \text{rand}_2(g(t) - x_i(t))$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (13)$$

$$\text{clamp}(x_i, \Omega^r); \text{clamp}(v_i, \pm v_{\max}^r) \quad (14)$$

$$p_i(t) = \begin{cases} x_i(t+1) & \text{if } f(x_i(t+1)) > f(p_i(t)) \\ p_i(t) & \text{otherwise} \end{cases} \quad (15)$$

$$g(t) = \begin{cases} x_i(t+1) & \text{if } f(x_i(t+1)) > f(g(t)) \\ g(t) & \text{otherwise} \end{cases} \quad (16)$$

$$\delta_i = \frac{\|x_i(t+1) - g(t+1)\|}{\|\text{range}(\Omega^r)\|} \quad (17)$$

End For

$$\text{If } \max(\delta_i) < \varepsilon, \forall i: 1 \leq i \leq N_p \quad (18)$$

$$\text{range}_k(\Omega^{r+1}) = \min \left(\left| g(t)_k^U - g(t)_k^L \right|, \rho \max_{1 \leq i \leq N_p} (|x_i(t)_k - g(t)_k|) \right) \quad (19)$$

$$\Omega^{r+1} = \begin{bmatrix} g(t)_1^U & g(t)_2^U & \cdots & g(t)_K^U \\ g(t)_1^L & g(t)_2^L & \cdots & g(t)_K^L \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \text{range}_k(\Omega^{r+1}) \\ -\text{range}_k(\Omega^{r+1}) \end{bmatrix} \quad (20)$$

$$v_{\max}^{r+1} = \lambda \text{range}(\Omega^{r+1}) \quad (21)$$

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

$$x_i(t) = g(t) + \text{rand}([0,1]) \circ \text{range}(\Omega^{r+1}) - 0.5 \text{range}(\Omega^{r+1}) \quad (22)$$

$$\bar{C}_{iNc+1} = \bar{z}_{\text{farthest}} \mid \bar{z}_{\text{farthest}} \in \max_{\forall \bar{z}_p \in x_{ij}(t)} d(\bar{z}_p, x_{ij}(t)) \quad (23)$$

$$x_i = [\bar{C}_{i1} \quad \bar{C}_{ij} \quad \cdots \quad \bar{C}_{iNc} \quad \bar{C}_{iNc+1}] \quad (24)$$

End For

End If

End While

Fig. 4. Regrouping Particle Swarm Clustering Algorithm

B. Gaussian Distribution of Clustered Knowledge Signal Fusion

The Gaussian Distribution of Clustered Knowledge (GCK) method is inspired by the Monte Carlo Experiments in which it relies random probabilities. GCK takes advantage of the clustered patterns statistical characteristics. It refers every incoming input signal to the cluster centroids as seen in Fig.5 and multiplexes it based on Gaussian characteristics of the cluster in which the signal is a member. This method, thus, is highly dependent on the quality and variability of training data provided and ultimately the quality of clusters.

Firstly the input signal $\bar{\gamma}$ is queried against the cluster

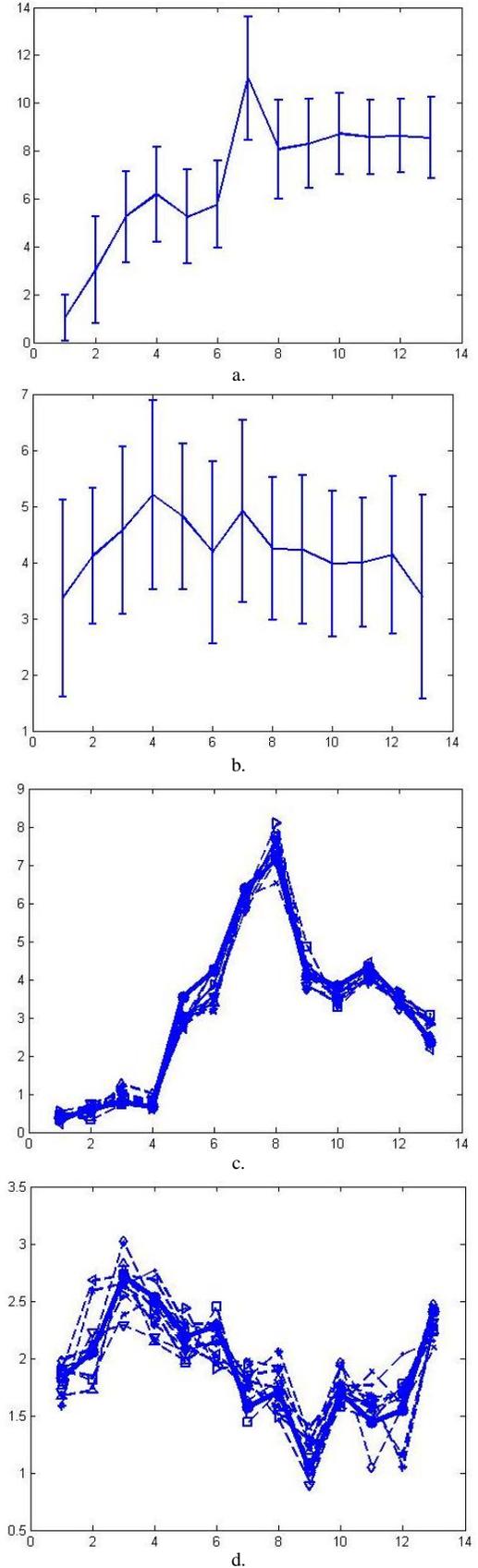


Fig. 5. a. and b. Sample knowledge signals from 2 clusters. c. denotes a fall right signal, d. shows an ADL signal. Both signals in c. and d. are fused with 10 GCK signals with $A = 0.8$ and $B = 0.2$. Straight line in c. and d. are original input signals $\bar{\gamma}$, while dashed lines are fused GCK signals $\mathfrak{G}(\bar{\gamma})$.

centroids and passed through a Radial Basis Kernel (25) to get the rate of membership $\theta_i(\bar{\gamma})$. Cluster I with the highest rate of membership is selected as the GCK seed. Knowledge signal $\bar{\gamma}$ is fabricated by generating a vector of Gaussian random number which mean $\bar{\mu}_I$ and standard deviation $\bar{\sigma}_I$ are the mean and standard deviation of the cluster (26). $\bar{\gamma}$ and the generated GCK signal $\bar{\gamma}$ is fused with a significance ratio of $A:B$ to create signal $\bar{\mathfrak{g}}(\bar{\gamma})$ (27). The result of the algorithm is shown in Fig. 5.

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

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

$$\bar{\mathfrak{g}}(\bar{\gamma}) = \frac{A\bar{\gamma} + B\bar{\gamma}(\bar{\gamma})}{A + B} \quad (27)$$

C. Augmented Radial Basis Neural-Network

Augmented Radial Basis Neural-Network (ARBF) has been previously used in time signal classification of head movement patterns with promising results. ARBF consists of an RBF layer and an MLP-NN augmentation layer as can be seen in Fig. 6. ARBF is reported to have a sensitivity advantage over conventional RBF and a specificity advantage over MLP [11].

The radial basis kernel used is the Gaussian radial function. The function can be described as a K-dimensional Gaussian distribution, where K is the dimension of the input. The output of the RBF layer is a vector of cluster membership. μ_n and σ_n corresponds to cluster centroids and the standard deviation of an RBF node.

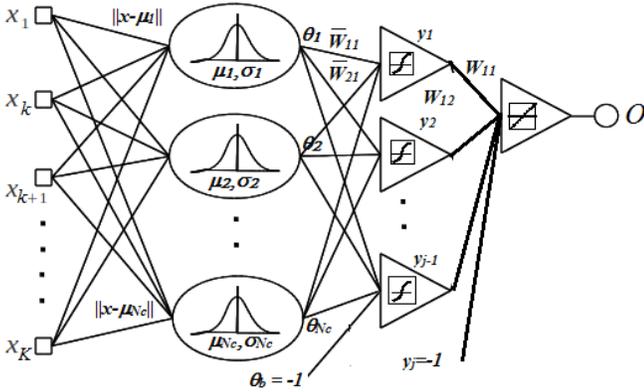


Fig.6. ARBF Configuration

The RBF centroids are optimized using RegPSO as has been explained in section IV.A. The standard deviation corresponds to the specified cluster standard deviation, set to a minimum bound of 5 as in Fig.7.b, which gives the best training result. Effect of varying standard deviation thresholds can be seen in the membership rate graph in Fig. 7.

The MLP layer uses sigmoid kernel in the hidden layer and linear kernel in the output layer. No normalization method is required since the RBF layer has already normalized the input signals from 0 to 1. The MLP layer is trained with resilient back-propagation.

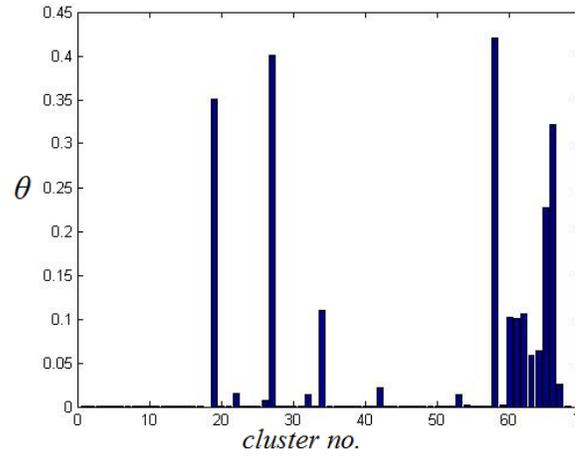
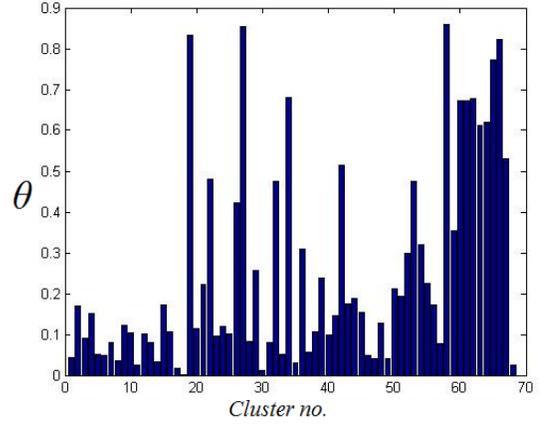


Fig. 7.a. Cluster Membership of a fall front signal with $\sigma = 12$; b. Cluster membership of the same signal with $\sigma = 5$

D. Ensemble of MLP and ARBF

Two classifiers are selected: Multilayer Perceptron (MLP), and ARBF. The classification scheme can be seen in Fig. 8. The combination between MLP and ARBF is selected because of the different characters that the two classifiers have. MLP networks perform better in global generalization while RBF-kernel based classifiers like ARBF perform better in local generalization [11]. It is thus proposed that the ensemble will be superior in both sensitivity and specificity.

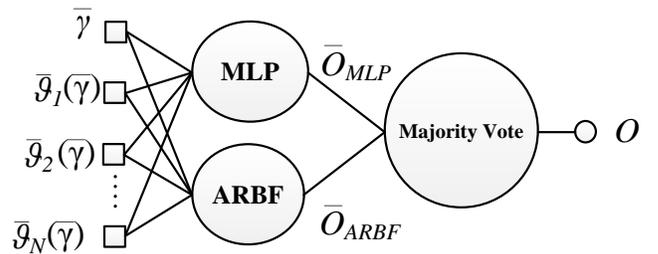


Fig. 8. MLP and ARBF Ensemble of Classifier

The input to the ensemble is a collection of signals that consists of the original signal $\bar{\gamma}$ and N GCK-Fused signals $\bar{\mathfrak{g}}$. Each Neural Network outputs $N + 1$ classifications of the input vectors. The outputs are then combined based on majority vote.

V. DATA COLLECTION

Data is divided into fall data and ADL. Training data is collected from 5 healthy volunteers, 2 females and 3 males. Outgroup test data is collected from 3 healthy male volunteers. The volunteers are aged between 19 and 28 years.

Falls are performed on top of a mattress. A total of 293 fall signals were recorded. 153 signals were used for training, 140 were used to test the system for ingroup performance, 85 fall signals were recorded to test outgroup performance.

ADL training data was collected from 3 people. A total of 8 hours of ADL is collected in a home environment. Another additional hour of exercise data in a gym environment is recorded from 2 individuals. 1831 ADL signals have been collected. 1000 randomly selected ADL signals were used for the training set while 831 were used for testing.

VI. RESULTS AND DISCUSSIONS

The collected data reflects the same quality mentioned in Chen. They reported that the recorded minimum impact of a fall is 3G. Acceleration magnitudes in ADL, except for heavy activities such as exercises rarely pass 3G [3].

Varying numbers of GCK signals with $A = 0.8$ and $B = 0.2$ are generated on the experimental inputs. It is shown in Table I that the addition of GCK signals improves overall system sensitivity in expense of specificity. It is also shown that ARBF does not generalize outgroup data as well as MLP.

The fall collection and testing was a highly challenging and time consuming task. Since most subjects are reluctant to fall, some of the falls are similar to sitting down or leaning. Another problem that is observed with the current training data is that the classifier works well only if the subject stays down after the fall. Break fall attempts are considered as positive falls as long as the subject falls in a specific impact magnitude and stays down afterwards. If the subject stands soon after the fall, the classifier will classify the fall as false. Based on this result, it is evident that collecting data of higher variability from more individuals would be required.

VII. CONCLUSION

Based on the experiment, it can be concluded that the proposed algorithm has so far given promising results on classifying falls based on a tri-axial accelerometer. DWT has successfully extracted essential features to characterize a fall. K-Means seeded RegPSO proves to be an ideal optimization

tool for data clustering. The proposed GCK algorithm has shown to improve sensitivity with the expense of specificity. In this application, the ensemble of MLP and ARBF has proven to be more reliable than a standalone classifier. The current ensemble with GCK fusion has successfully achieved 100% sensitivity on ingroup falls, 97.65% on outgroup falls, 99.33% specificity on routine ADL, and 96.59% specificity on exercise ADL.

In future research, additional data with higher variability will be required. Further analysis on fall related signals using other motion sensors such as gyroscope will need to be carried. In order to advance this research to a practical stage, we are currently embedding this algorithm under AndroidOS using the smartphone's internal accelerometer.

REFERENCES

- [1] World Health Organization, *WHO Global Report on Falls Prevention in Older Age*. France, 2007, ch.1.
- [2] M. Kangas, A. Konttila, I. Winblad, and T. Jämsä, "Determination of simple thresholds for accelerometry-based parameters for fall detection", in *Proc. of the 29th Annual International Conference of the IEEE EMBS*, Lyon, 2007, pp.1367-1370.
- [3] J. Chen, K. Kwong, D. Chang, J. Luk, R. Bajcsy, "Wearable Sensors for Reliable Fall Detection", in *Proc. of the 27th Annual Conference of the IEEE EMBS*, Shanghai, 2005, pp. 3551-3554.
- [4] A.K. Bourke, P. vd Ven, M. Gamble, R. O'Connor, K. Murphy, E. Bogan, E. McQuade, P. Finucane, G. ÓLaighin, and J. Nelson, "Assessment of Waist-worn Tri-Axial Accelerometer Based Fall-detection Algorithm using Continuous Unsupervised Activities", in *Proc. of the 32nd Annual Conference of the IEEE EMBS*, Buenos Aires, 2010, pp. 2782-2785.
- [5] G. Shi, C.S. Chan, W.J. Li, K.S. Leung, Y. Zou, and Y. Jin, "Mobile Human Airbag System for Fall Protection Using MEMS Sensor and Embedded SVM Classifier", *IEEE Sensors Journal*, vol. 9, no. 5, May. 2009, pp.495-503.
- [6] P. Lajšner and R. Kozub, *Using the MMA7360L ZSTAR2 Demo Board*, Freescale Semiconductor, November 2007.
- [7] O. Rioul and P. Duhamel, "Fast Algorithm for Discrete and Continuous Wavelet Transforms", *IEEE trans. on Information Theory*, vol. 38, no. 2, March. 1992, pp.569-586.
- [8] D.W. Merve and AP Engelbrecht, "Data Clustering using Particle Swarm Optimization" in *Congress on Evolutionary Computation*, vol.1, 2003, pp. 215 - 220.
- [9] R. Eberhart and J. Kennedy, "A New Optimizer Using Particle Swarm Theory", in *IEEE Sixth International Symposium on Micro Machine and Human Science*, 1995, pp. 39-43.
- [10] G.I. Evers and M.B. Ghalia, "Regrouping Particle Swarm Optimization: A New Global Optimization Algorithm with Improved Performance Consistency Across Benchmarks", in *International Conference on Systems, Man, and Cybernetics 2009*, San Antonio, TX, USA, 2009, pp.3901-3908.
- [11] M. Yuwono, A.M.A. Handojoseno, and H.T. Nguyen, "Optimization of Head Movement Recognition Using Augmented Radial Basis Function Neural Network", in *Proc. of the 33rd Annual Conference of the IEEE EMBS*, Boston, 2011, to be published.

GCK Signals	ARBF				MLP				Ensemble MLP + ARBF			
	Fall Sensitivity		ADL Specificity		Fall Sensitivity		ADL Specificity		Fall Sensitivity		ADL Specificity	
	Ingroup (N=140)	Outgroup (N=85)	Routine (N=450)	Exercise (N=381)	Ingroup (N=140)	Outgroup (N=85)	Routine (N=450)	Exercise (N=381)	Ingroup (N=140)	Outgroup (N=85)	Routine (N=450)	Exercise (N=381)
0	92.59%	85.88%	100.00%	98.95%	96.43%	89.29%	99.33%	96.85%	96.43%	92.94%	100.00%	98.95%
1	93.33%	87.94%	100.00%	98.95%	97.86%	94.12%	99.33%	95.80%	98.57%	94.12%	99.78%	98.43%
2	95.56%	87.94%	100.00%	97.90%	97.86%	95.29%	99.33%	95.54%	98.57%	95.29%	99.78%	97.64%
3	94.07%	91.76%	100.00%	97.90%	98.57%	95.29%	99.33%	95.54%	100.00%	95.29%	99.78%	97.38%
4	94.07%	91.76%	99.78%	97.64%	98.57%	95.29%	99.33%	95.28%	100.00%	95.29%	99.56%	97.11%
5	95.56%	88.24%	99.78%	97.11%	100.00%	95.29%	99.33%	95.28%	100.00%	97.65%	99.56%	96.85%
10	96.30%	88.24%	99.78%	97.11%	100.00%	95.29%	99.33%	95.28%	100.00%	97.65%	99.33%	96.59%

TABLE I: EXPERIMENTAL RESULTS