

# Fall Detection using Lifting Wavelet Transform and Support Vector Machine

Hanghan Liang, Wipawee Usaha<sup>†</sup>  
School of Telecommunication Engineering, Suranaree  
University of Technology, Muang, Nakhon Ratchasima  
30000, Thailand  
Email: lianghanghan@gmail.com, wusaha@ieee.org<sup>†</sup>

□

**Abstract**—Frequency domain features of inertial movement enables multi-resolution analysis for fall detection, yet they are computationally intensive. This paper proposes a computationally light frequency domain feature extraction method based on lifting wavelet transform (LWT) which provides computational efficiency suitable for real-time low power devices such as wearable sensors for human fall detection. LWT is combined with support vector machine (SVM) to identify falls from activities of daily living. Performance of the Haar and Biorthogonal 2.2 wavelets were compared with the time domain feature of root-mean square acceleration using a human fall dataset. Results show that the first level detail coefficients features for both Haar and Biorthogonal 2.2 wavelets achieve good overall fall detection accuracy, sensitivity and specificity.

## I. INTRODUCTION

As many countries enter the era of aging society, they face critical elderly people's health threats which are fall and related complications caused by the injury [1]. Considering the need of real-time monitoring and ease of use, wearable sensor systems are one of the most promising systems.

Wearable sensor-based fall detection systems, inherently generate continuous monitoring of physiological measurements. Such system is usually a multi-sensor system, comprising sensors such as accelerometers, gyroscopes, pressure sensors and magnetometers. Datasets collected by such wearable sensors are thus, typically multi-dimensional and in large volumes. Such characteristics may cause hinder data processing and fall detection capabilities. Some researches therefore use feature extraction to reduce the amount and the dimensions of data [2] by extracting only necessary features. Existing feature extraction techniques include two main domains, i.e., time and frequency domains. Research such as [1], [3], [4] extracted time domain features including the mean value, maximum value, minimum value and variance, standard deviation of the patient's physiological movements and other special features such as entropy and vertical direction.

In general, time domain features are straightforward and easy to visualize which means light computational burden for feature extraction. So the system is computationally efficient in achieving a real-time fall detection. However, the time domain statistical features considers only the displayed observable trends [2]. Consequently, time domain features may not suffice for accurate fall detection.

Conversely, frequency domain features make use the spectral domain of the collected data which may not be clearly observable in the time domain. Frequency domain features were deployed for fall detection by [5] which used discrete stationary wavelet transform (SWT). In [6], a short time Fourier transform (STFT) was used for human activity recognition, whereby a fall was a subset of data in a series of continuous activities of daily living (ADLs). Ref. [7] created a prototype wavelet of typical fall pattern by using the average acceleration sum vector. The degree of similarity of the signal to the prototype was then computed through wavelet analysis. Results from the same classifier and real-world dataset revealed that the wavelet based features outperformed than other time domain features: upper and lower peak values.

Feature extraction alone only enhance the features of the data acquired by the wearable sensors. However, to detect weather a fall occurred relies on the performance of the detection mechanism. The most common and simplest fall detection is the threshold method [8]. Nevertheless, the performance heavily depends on the fixed threshold level. Hence, it is rarely used alone, and often combined with other machine learning methods such as decision tree (DT) [9], [10], artificial neural networks (ANN) [11], hidden Markov model (HMM) [12] and Support Vector Machine (SVM) [4], [14], [15] can be combined to outperform the threshold method [8], [14]. Among the machine learning methods, SVM was found the most robust for fall detection when compared to other methods such as threshold-based methods and the decision tree method [8]. However, most works which deploy SVM for fall detection use time-series features [8], [16]. It was found that SVM fall detection performance can be improved by a combination of time and frequency domain features [4]. In particular, the discrete Fourier transform

<sup>†</sup>Corresponding author

This work was financially supported by Suranaree University of Technology under the MOU with Huazhong University of Science and Technology, P.R. China.

(DFT) was used to determine the spectral coefficients which is computationally intensive [4]. On the other hand, the lifting wavelet transform (LWT) is an efficient, light weight frequency domain extraction method [17]. To the best of our knowledge, there is no previous work that has combined LWT with SVM for fall detection. This paper is therefore focused on the study of feature extraction based on LWT used with SVM to detect falls from ADLs using root-mean square value from a single tri-axial acceleration sensor.

The paper is organized as follows. Section II presents the proposed frequency analysis and the support vector machine scheme proposed in the paper. The time domain feature which is used for comparison is also introduced. In section III, the experiment based on a comprehensive fall detection dataset is described. Section IV presents the results and discussion and finally conclusions is given in the final section.

## II. METHOD

### A. Frequency domain feature extraction

Feature extraction based on frequency analysis of the body inertia collected from sensors has been studied in the recent literature. Discrete wavelet transform (DWT) has been proposed for mobility monitoring, posture transition and activities classification in [18] using a single chest-mounted sensors. In [19], another frequency domain feature extraction method using short-time Fourier transform (STFT) was proposed to shorten the calculation time of DWT. Despite good results, the short time windows in STFT may not always be suitable for human motion which varies greatly. If windows are too short, STFT may be unable to identify the frequency in such a short period of time. If windows are too large, more information in time domain will be lost. If the STFT window size is fixed, STFT may not be suitable for fall detection as human activities are flexible. Unlike DFT in [4], LWT can be constructed from time series signal directly. Unlike DWT in [18], LWT does not require convolution, translation or dilation of traditional mother wavelets. Furthermore, LWT allows in place calculation, with no need for auxiliary memory. Therefore, LWT provides computational efficiency suitable for real-time low power devices such as wearable sensors. In the following subsection, we describe LWT in more details.

### B. Lifting Wavelet Transform

LWT has been introduced by Sweldens in 1997 [17]. The scheme theory is often described as three steps: split, predict and update. The split step is to split a signal into two independent sequences, i.e., the even half and odd half sequences. Let  $x_i$  be the original discrete signal at time index  $i$ . Let  $even_i$  ( $odd_i$ ) denote the  $i^{\text{th}}$  index of the even (odd) sequence. We have that  $even_i = x_{2i}$  and  $odd_i = x_{2i+1}$ ,  $i \in I$ .

LWT is a recursive algorithm which splits the signal into halves at each level. If the original signal has  $2^n$  elements, then the next level will operate on  $2^{(n-1)}$  elements. Hence, if the original signal has 256 elements, there will be 8 levels with

the next level having 128 elements. The subsequent levels will have 64, 32, 16, 8, 4, 2 and 1 element. The odd values in the next level  $j+1$  is predicted from the even value at level  $j$ :

$$cD_{j+1,i} = odd_{j,i} - P(even_{j,i}) \quad (1)$$

where  $P$  is the predict function which approximates the signal,  $cD$  is the high frequency component of  $x_i$ . This is called the *Predict* phase. The even values at the next level can be found from

$$cA_{j+1,i} = even_{j,i} + U(cD_{j+1,i}) \quad (2)$$

where  $U$  is the update operation that updates on the differences from the odd values,  $cA$  is the low frequency component of  $x_i$ . This is called the *Update* phase. The multi-level lifting scheme can be summarized in Fig. 1. The averages are sometimes called approximate coefficients whereas the differences are called the detail coefficients. There are two types of wavelets used in this paper.

#### 1) Haar wavelet:

Predict :

$$cD_{j+1,i} = odd_{j,i} - even_{j,i} \quad (3)$$

Update :

$$cA_{j+1,i} = even_{j,i} + \frac{1}{2}cD_{j+1,i} \quad (4)$$

#### 2) Biorthogonal 2.2 wavelet:

Predict :

$$cD_{j+1,i} = odd_{j,i} - \frac{1}{2}(even_{j,i} + even_{j,i+1}) \quad (5)$$

Update :

$$cA_{j+1,i} = even_{j,i} + \frac{1}{4}(cD_{j+1,i-1} + cD_{j+1,i}) \quad (6)$$

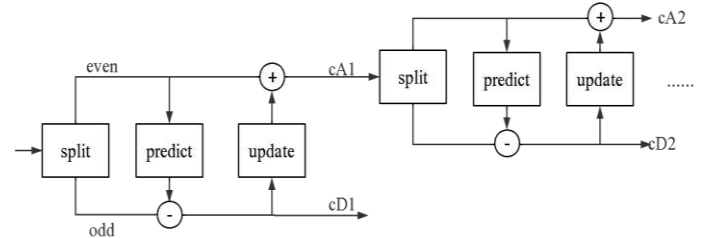


Fig. 1 Forward lifting scheme

Figure 2 shows a sample fall plot of the original signal and the *first* level LWT coefficient. The number of coefficients of  $cA1$  (average or low frequency part) and  $cD1$  (detail or high frequency part) are half of the original signal according to the number of data points. By comparing  $cA1$ ,  $cD1$  and the root-mean square acceleration ( $SV_{total}$ ) in Fig. 2, it is seen that  $cA1$  greatly correlates with the original signal. Note that  $cD1$  also shows a peak similar to the original signal signifying a fall which occurred during the red highlighted window of one second. However, the baseline zero illustrated a more distinguished fall feature than  $cA1$ . Therefore,  $cD1$  was preferable than  $cA1$  for feature extraction of falls.

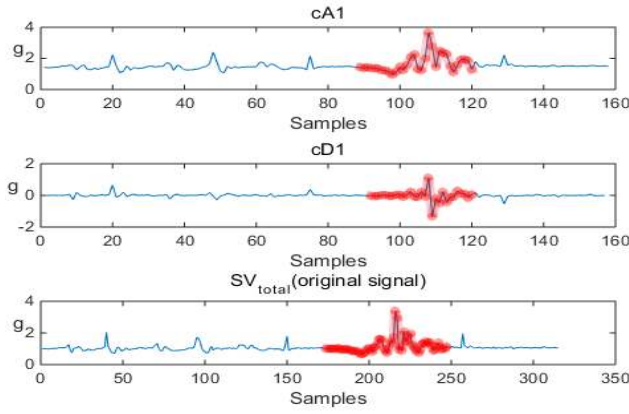


Fig. 2 A sample fall plot of  $SV_{total}$  and after LWT cD1 and cA1 with data points inside “fall” window highlighted

### C. Time domain feature

The tri-axial acceleration data collected contains  $A_x$ ,  $A_z$ ,  $A_y$  in x-axis, z-axis and y-axis as a function of time. All accelerometer data were in factors of gravity units (g). The accelerometer components were used to calculate the root-mean square acceleration denoted by total sum vector  $SV_{total}$ :

$$SV_{total} = \sqrt{A_x^2 + A_y^2 + A_z^2} \quad (7)$$

### D. Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning model which is commonly used for anomaly detection and classification [20], [21], [22]. As a supervised learning model, SVM requires training from datasets with “labels.” The SVM concept is to map a set of data points from the real-world to a higher dimensional space. A boundary or hyperplane is created in a high dimensional space by training datasets to classify the features into fall or non-fall. Since the fall detection system inherently generates long-term continuous monitoring of physiological measurements, such datasets are usually large. Such characteristic may cause difficulty in data processing. To reduce the amount of data and achieve a higher calculating speed, the features of the data may be extracted from these raw datasets.

To train the SVM, the data points in the dataset must be labeled. For example, in time domain,  $SV_{total}$  was directly used as input feature. We labeled all the ADLs data points with “-1” whereas falls were labeled “1.” Fig. 3 depicts a sample plot of a fall along with non-fall activities like walking around and lying on the ground. Point A shows the peak value of the dataset. A highlighted window size with point A placed at the middle of the window is constructed. Within such window, all the data points are labeled “1” and the remaining data points outside this window are labeled “-1.” The goal of SVM is therefore to distinguish the labels among the tested datasets using the model obtained from the trained data. The data points are typically non-linearly separable to classify in low dimensional space. However, if these points are projected onto a higher dimensional space, it is possible to find a hyperplane to classify the labels. Such projection is

obtained through use of kernel functions such as linear, polynomial, sigmoidal, or the Gaussian radial base functions. It is with this kernel trick that makes SVM a powerful model to classify the labels in higher dimensional space. In the next section, the experiment settings are presented.

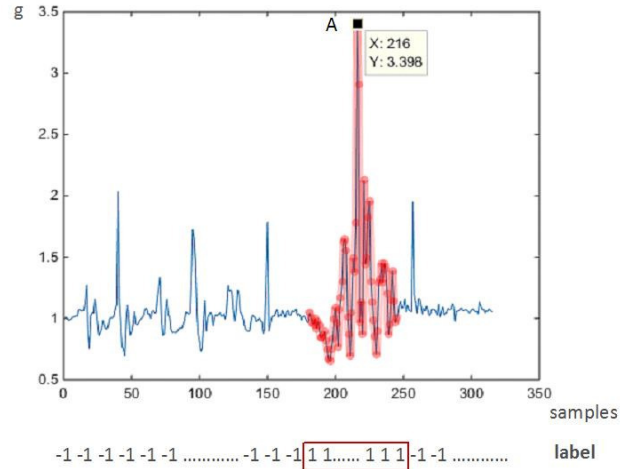


Fig. 3 A sample plot of fall  $SV_{total}$  with data points inside window highlighted

## III. EXPERIMENT

As mentioned in the previous section, SVM requires training labeled datasets. As data input in the fall detection scenario involves both non-falls and falls data, We trained with both falls and non-falls italic in Table I. we first evaluate the SVM model with a hybrid fall and non-fall activities. The objective is to evaluate a suitable training dataset for SVM to detect falls. For the sake of simplicity, only the time domain feature ( $SV_{total}$ ) is studied.

Once a SVM model is trained, we proceed to study the comparison between features in the time domain ( $SV_{total}$ ) and frequency domain (LWT using Haar and Biorthogonal 2.2 wavelets). Note that there are existing works which combined features in both time domain and frequency domain of data, the type of sensors, the number and position of sensors on human body, and in the volume of dataset for training and testing [1], [5]. From results gathered from existing literature, we focus on data collected from a single tri-axial acceleration sensor due to its low cost, reliability and efficiency.

### A. Performance metrics

To evaluate the performance, we measure the True Positives (TP) or True Negatives (TN) which refers to the number of events correctly identified or correctly rejected. False Positives (FP) or False Negatives (FN) which represent the number of events incorrectly identified or incorrectly rejected [23]. These measurements provide the following necessary metrics required to evaluate the fall detection method:

1) *Sensitivity (SE)* or true positive rate is the capability to detect a fall correctly. It is an indicator to judge whether a system will miss a fall. It is given by

$$SE = \frac{TP}{TP+FN} \times 100\% \quad (8)$$

2) *Specificity (SP)* or true negative rate is the ability to detect a fall only if a fall really occurred. It is to avoid false alarm given by

$$SP = \frac{TN}{TN+FP} \times 100\% \quad (9)$$

3) *Accuracy (AC)* or correct rate refers to the overall freedom from false. This is given by

$$AC = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (10)$$

It is worth noting that SVM classifies data points individually. However, to detect a fall within a certain window as shown in Fig. 3, a set of data points must be classified rather than just a single data point. Therefore, to determine a suitable decision region to decide whether a fall has occurred, we use a simple calculation for the percentage

of predicted fall label “1” over the number of labels observed in an activity to compare with a predetermined threshold:

$$T = \frac{\text{the number of predicted "1"}}{\text{the number of testing data points}} \quad (11)$$

If  $T > \text{threshold}$ , the activity is a fall. Otherwise, else it is a non-fall activity.

### B. Training SVM Model

We hypothesize that the best type of training dataset will be the combined set of both fall and ADLs dataset. Since not only falls but also ADLs data are contained in the hybrid training dataset, the more comprehensive information contained in training dataset, the more likely the model will decide correctly.

The dataset we used to train and test the SVM models have been obtained from [24] including 70 activities (tri-axial acceleration of 30 falls and 40 non-falls collected and video recorded with Kinect camera) with details given in Table I. The tri-axial accelerometer data was sampled at 60Hz. Therefore, a one-second window for fall detection consists of

TABLE I.  
DATASETS USED IN EXPERIMENT<sup>1</sup>

Data file	Activities description	Data file	Activities description
<b>Falls Activities</b>			
<i>fall-01-acc</i>	<i>From vertical falling left on the floor</i>	<i>fall-16-acc</i>	From sitting falling right on the floor
<i>fall-02-acc</i>	<i>From sitting falling left on the floor</i>	<i>fall-17-acc</i>	From vertical falling forward on the floor
<i>fall-03-acc</i>	From vertical falling left on the floor	<i>fall-18-acc</i>	From sitting falling left on the floor
<i>fall-04-acc</i>	From sitting falling left on the floor	<i>fall-19-acc</i>	From vertical falling right on the floor
<i>fall-05-acc</i>	<i>From vertical falling right on the floor</i>	<i>fall-20-acc</i>	From sitting falling right on the floor
<i>fall-06-acc</i>	<i>From sitting falling right on the floor</i>	<i>fall-21-acc</i>	From vertical falling right on the floor
<i>fall-07-acc</i>	From vertical falling left on the floor	<i>fall-22-acc</i>	From sitting falling left on the floor
<i>fall-08-acc</i>	From sitting falling right on the floor	<i>fall-23-acc</i>	From vertical falling right on the floor
<i>fall-09-acc</i>	From vertical falling left on the floor	<i>fall-24-acc</i>	From sitting falling left on the floor
<i>fall-10-acc</i>	From sitting falling left on the floor	<i>fall-25-acc</i>	From vertical falling forward on the floor
<i>fall-11-acc</i>	From vertical falling right on the floor	<i>fall-26-acc</i>	<i>From sitting falling forward on the floor</i>
<i>fall-12-acc</i>	From sitting falling right on the floor	<i>fall-27-acc</i>	From vertical falling forward on the floor
<i>fall-13-acc</i>	<i>From vertical falling forward on the floor</i>	<i>fall-28-acc</i>	From sitting falling forward on the floor
<i>fall-14-acc</i>	From sitting falling right on the floor	<i>fall-29-acc</i>	From vertical falling forward on the floor
<i>fall-15-acc</i>	From vertical falling forward on the floor	<i>fall-30-acc</i>	From sitting falling forward on the floor
<b>Non-falls Activities (ADLs)</b>			
Data file	Activities description	Data file	Activities description
<i>adl-01-acc</i>	<i>Walking, then squatting</i>	<i>adl-21-acc</i>	From vertical lying on the bed
<i>adl-02-acc</i>	Walking, then squatting	<i>adl-22-acc</i>	From vertical lying on the bed
<i>adl-03-acc</i>	Walking, then squatting	<i>adl-23-acc</i>	From vertical lying on the bed
<i>adl-04-acc</i>	<i>Bending 90 degree to pick up something</i>	<i>adl-24-acc</i>	Walking, then squatting
<i>adl-05-acc</i>	Squatting to pick up something	<i>adl-25-acc</i>	From vertical to sitting onto a chair
<i>adl-06-acc</i>	Squatting to pick up something	<i>adl-26-acc</i>	Walking, then squatting
<i>adl-07-acc</i>	<i>From vertical to sitting onto a chair</i>	<i>adl-27-acc</i>	From vertical to sitting onto a chair
<i>adl-08-acc</i>	From vertical to sitting onto a chair	<i>adl-28-acc</i>	Walking, then squatting
<i>adl-09-acc</i>	From vertical to sitting onto a bed	<i>adl-29-acc</i>	From vertical to sitting onto a chair
<i>adl-10-acc</i>	<i>From vertical lying on the bed</i>	<i>adl-30-acc</i>	From vertical lying leftward on the ground
<i>adl-11-acc</i>	From vertical lying rightward on the bed	<i>adl-31-acc</i>	<i>From vertical lying forward on the ground</i>
<i>adl-12-acc</i>	Walking, then squatting	<i>adl-32-acc</i>	From vertical lying forward on the ground
<i>adl-13-acc</i>	Walking, then squatting	<i>adl-33-acc</i>	From vertical lying forward on the ground
<i>adl-14-acc</i>	Walking, then squatting	<i>adl-34-acc</i>	From vertical lying forward on the ground
<i>adl-15-acc</i>	Bending 90 degree to pick up something	<i>adl-35-acc</i>	From vertical lying forward on the ground
<i>adl-16-acc</i>	Bending 90 degree to pick up something	<i>adl-36-acc</i>	From vertical lying rightward on the ground
<i>adl-17-acc</i>	<i>Squatting to pick up something</i>	<i>adl-37-acc</i>	From vertical lying rightward on the ground
<i>adl-18-acc</i>	From vertical to sitting onto a bed	<i>adl-38-acc</i>	From vertical lying forward on the ground
<i>adl-19-acc</i>	From vertical to sitting onto a chair	<i>adl-39-acc</i>	From vertical lying forward on the ground
<i>adl-20-acc</i>	From vertical to sitting onto a bed	<i>adl-40-acc</i>	From vertical lying forward on the ground

Source: <http://fenix.univ.rzeszow.pl/~mkepski/ds/uf.html>

<sup>1</sup> The italic activities were used as training dataset.

60 data points. The dataset was divided into training set and testing set based on activities in the matching video of each data file. Table I consists of fall and non-fall (ADLs) activities. The SVM model has been trained with the datasets obtained in italics in Table I for a comprehensive dataset of various falls and ADL activities.

Once the data points are labeled and trained, the SVM model is obtained. The SVM model is then used to classify the testing data. The dataset remaining (non-italic activities) in Table I are used for testing. For each dataset tested, a data point is labeled “1” for data points predicted as a fall data point, or “-1” for data points predicted as non-fall data point. If the ratio of fall labels in an activity exceeds the determined threshold, then a fall has been detected. For each tested dataset, TP, TN, FP and FN is measured for the calculation of SE, SP and AC to evaluate the SVM model. Results are presented in Section IV.

### C. Comparing Time and Frequency domain features

This part of the experiment is to compare the time domain feature (based on  $SV_{total}$ ) and the frequency domain features (based on Haar and Biorthogonal 2.2 wavelets). Using the SVM model obtained in the previous experiment, a suitable level threshold level to detect a fall event for each feature is then found. For each feature, the percentage levels of threshold is tested at 10%, 20%, 30%, 40% and 50%. Then level is tested at finer threshold values. Results are shown in Table II.

We then investigate closely how multiple levels of LWT coefficients affect the fall detection performance by evaluating the first five levels of coefficients of the Haar and Biorthogonal 2.2 wavelets. Results are shown in Table III.

## IV. RESULTS AND DISCUSSION

### A. Training SVM Model

Results show that the SVM model trained and tested with *time domain* datasets of both falls and ADL activities gave a 100% sensitivity, 97.14% of specificity and 98.31% accuracy. It should be noted that the 100% sensitivity is obtained from *offline* datasets with a predetermined threshold found from observing these datasets. Furthermore, a larger dataset collected from online simulated falls is currently under investigation.

### B. Comparing Time and Frequency domain feature

Table II shows the performance comparison between time and frequency domain features at different levels of thresholds.

1) *Root-mean square acceleration*: Table II shows that the best threshold for the time domain feature should be between 10% to 20%. With fine threshold tuning, it is found that a threshold of 17-18% showed better preference than others (shown in bold fonts). Therefore, we chose 17% as the threshold to classify a fall or non-fall for time domain feature.

2) *LWT with Haar Wavelet*: The appropriate threshold for Haar LWT is found by also ranged from 10% to 50%. As

shown in Table II, the best achieved threshold should be under 10%. To fine tune the threshold levels, the threshold is varied from 2% to 10%. It is found that the threshold at 8% outperformed other levels (shown in bold fonts). Thus, we chose 8% as the threshold for LWT using Haar wavelet. In Table III, multiple levels of LWT coefficients (cD1 to cD5) are evaluated. When tested with ADLs & Falls dataset, all specificity, specificity and accuracy values of 100% was achieved only in cD1 (shown in bold fonts). This result indicated that Haar LWT CD1 coefficients achieved a goal such that no ADL has been misclassified as a fall and detected most of the falls when training and testing using finite activities in Table I

3) *LWT with Biorthogonal 2.2 Wavelet*: From Table II, the optimal threshold for Biorthogonal 2.2 (Bior 2.2) should be under 10%. With a finer threshold search, results indicate that threshold level of 6% is the best level with 100% sensitivity, specificity and accuracy (shown in bold fonts). Similar to Haar LWT, Bior 2.2 LWT coefficients also show a good performance distinguishing falls from ADLs when using most cD levels. In Table III, cD1 also outperformed other levels of coefficients similar to Haar wavelet (shown in bold fonts). The reason may be the information contained in the frequency components that is helpful to classify activities by SVM. The cD1 components contained the most distinguishable information of falls, while cD5 contained the least information. Generally, Haar was slightly better at distinguishing ADLs from falls than Bior 2.2, whereas both LWT features outperform the time domain feature of root-mean square acceleration alone. It is worth noting that these results are obtained by a comprehensive human fall dataset with video captures obtained from [24] which allow the thresholds and detail coefficients to be predetermined offline. Current ongoing work involves implementing the LWT and SVM on actual wearable sensor devices to be evaluated online for human fall detection for accuracy and efficiency.

## V. CONCLUSIONS

In this paper, we propose a computationally light frequency domain feature extraction method called lifting wavelet transform (LWT) for a wearable sensor human fall detection device combined with a fall identifier using support vector machine model. The performance of the LWT using Haar and Biorthogonal 2.2 wavelets, together with the time domain feature of root-mean square acceleration have been evaluated with raw dataset acquired from a single tri-axial acceleration sensor from an existing human fall and activities of daily living dataset.

Based on the dataset, suitable thresholds and level of detail coefficients can be predetermined. Consequently, the LWT frequency domain features are shown to have better performance than time domain features in terms of sensitivity, specificity and accuracy. Given a one-second window size under a sampling frequency of 60Hz, the best threshold in terms of the percentage of fall labels (“1”) per window is as follows, 18% for the time domain feature using the root-mean square acceleration, and 8% for Haar and 6% for

Biorthogonal 2.2 LWT wavelets when the SVM model is trained with both fall and non-fall datasets. The frequency domain feature from cD1 for both Haar and Biorthogonal 2.2 wavelets achieved 100% overall accuracy whereas 98.31% overall accuracy was attained for the time domain feature,  $SV_{total}$ . All features achieved 100% sensitivity from this dataset. In terms of specificity, the time domain feature,  $SV_{total}$ , attained up to 97.14% whereas the two LWT features attained 100%. In a final note, ongoing work involves implementing the LWT and SVM on actual wearable sensor devices to be evaluated for human fall detection accuracy and reliability in real-time.

#### ACKNOWLEDGMENT

The authors would like to thank Suranaree University of Technology for the financial support under the MoU with Huazhong University of Science and Technology, P.R. China for Ms.Hanghan Liang to conduct this research.

#### REFERENCES

- [1] Pierleoni P, Pernini L, Belli A, et al. "SVM-based fall detection method for elderly people using Android low-cost smartphones," *IEEE Sensors Applications Symposium (SAS 2015)*, 2015: 1-5.
- [2] Banaee H, Ahmed M U, Loutfi A., "Data mining for wearable sensors in health monitoring systems: a review of recent trends and challenges," *Sensors*, 2013, 13(12): 17472-17500.
- [3] Carlsson T., "Individualized Motion Monitoring by Wearable Sensor: Pre-impact fall detection using SVM and sensor fusion," Masters Thesis, School of Technology and Health, KTH Royal Institute of Technology, Stockholm, Sweden, 2015.
- [4] Özdemir A T, Barshan B., "Detecting falls with wearable sensors using machine learning techniques," *Sensors*, 2014, 14(6): 10691-10708.
- [5] Su B Y, Ho K C, Rantz M J, et al., "Doppler radar fall activity detection using the wavelet transform," *IEEE Transactions on Biomedical Engineering*, 2015, 62(3): 865-875.
- [6] Björklund S, Petersson H, Hendeby G. , "Features for micro-Doppler based activity classification," *IET Radar, Sonar & Navigation*, 2015, 9(9): 1181-1187.
- [7] Palmerini L, Bagalà F, Zanetti A, et al., "A wavelet-based approach to fall detection," *Sensors*, 2015, 15(5): 11575-11586.
- [8] Aziz O, Musngi M, Park E J, et al., "A comparison of accuracy of fall detection algorithms (threshold-based vs. machine learning) using waist-mounted tri-axial accelerometer signals from a comprehensive set of falls and non-fall trials," *Medical & Biological Engineering & Computing*, 2017, 55(1): 45-55.
- [9] Bilski P, Mazurek P, Wagner J, et al., "Application of Decision trees to the Fall Detection of elderly People using Depth-based sensors," *Proc. IEEE International Conference on Intelligent Data Acqut and Advanced Computing Systems (IDAACS 2015)*, Warsaw, Poland, September, 2015.
- [10] Parkka J, Ermes M, Korpipaa P, et al. , "Activity classification using realistic data from wearable sensors," *IEEE Transactions on Information Technology in Biomedicine*, 2006, 10(1): 119-128.
- [11] Wang Z, Jiang M, Hu Y, et al., "An incremental learning method based on probabilistic neural networks and adjustable fuzzy clustering for human activity recognition by using wearable sensors," *IEEE Transactions on Information Technology in Biomedicine*, 2012, 16(4): 691-699.

TABLE II.  
PERFORMANCE COMPARISON<sup>1</sup> OF DIFFERENT THRESHOLDS FOR TIME AND FREQUENCY DOMAIN FEATURES

	INITIALLY ESTIMATED THRESHOLDS					FINE TUNED THRESHOLDS <sup>2</sup>			
<b>TIME DOMAIN (<math>SV_{TOTAL}</math>)</b>									
<b>THRESHOLD</b>	10%	20%	30%	40%	50%	15%	<b>17%</b>	<b>18%</b>	19%
<b>SE (%)</b>	100	95.83	91.67	87.50	87.50	100	<b>100</b>	<b>100</b>	100
<b>SP (%)</b>	80	100	100	100	100	94.29	<b>97.14</b>	<b>97.14</b>	91.43
<b>AC (%)</b>	88.14	98.31	96.61	94.92	94.92	96.61	<b>98.31</b>	<b>98.31</b>	98.31
<b>FREQUENCY DOMAIN (HAAR, CD1)</b>									
<b>THRESHOLD</b>	10%	20%	30%	40%	50%	2%	4%	6%	<b>8%</b>
<b>SE (%)</b>	95.83	87.50	87.50	66.67	33.33	100	100	100	<b>100</b>
<b>SP (%)</b>	100	100	100	100	100	82.35	85.29	97.06	<b>100</b>
<b>AC (%)</b>	98.28	94.83	94.83	86.21	72.41	89.66	91.38	98.28	<b>100</b>
<b>FREQUENCY DOMAIN (BIOR2.2, CD1)</b>									
<b>THRESHOLD</b>	10%	20%	30%	40%	50%	4%	5%	<b>6%</b>	
<b>SE (%)</b>	95.83	87.50	83.33	62.50	33.33	100	100	<b>100</b>	
<b>SP (%)</b>	100	100	100	100	100	88.57	97.14	<b>100</b>	
<b>AC (%)</b>	98.31	94.92	93.22	84.75	72.88	93.22	98.31	<b>100</b>	

<sup>1</sup>Trained with ADLs & Falls SVM Model and tested by ADLs & Falls dataset

<sup>2</sup> Bold fonts indicate the best performance for each feature

TABLE III.  
PERFORMANCE COMPARISON<sup>1</sup> OF DIFFERENT FREQUENCY DOMAIN COMPONENTS<sup>2</sup>

FEATURES	CD1	CD2	CD3	CD4	CD5
<b>HAAR</b>					
<b>SE (%)</b>	<b>100</b>	83.33	95.83	87.50	95.83
<b>SP (%)</b>	<b>100</b>	100	100	100	100
<b>AC (%)</b>	<b>100</b>	93.10	98.28	94.83	98.28
<b>BIOR2.2</b>					
<b>SE (%)</b>	<b>100</b>	100	91.67	100	100
<b>SP (%)</b>	<b>100</b>	94.29	100	82.86	5.71
<b>AC (%)</b>	<b>100</b>	96.61	96.61	89.83	44.07

<sup>1</sup>Trained with SVM Model-3 and tested by ADLs & Falls dataset

<sup>2</sup> Bold fonts indicate the best performance for each feature

- [12] Tong L, Song Q, Ge Y, et al., "HMM-based human fall detection and prediction method using tri-axial accelerometer," *IEEE Sensors Journal*, 2013, 13(5): 1849-1856.
- [13] Kianoush S, Savazzi S, Vicentini F, et al., "Leveraging RF signals for human sensing: fall detection and localization in human-machine shared workspaces," // *Industrial Informatics (INDIN)*, 2015 IEEE 13<sup>th</sup> International Conference on. IEEE, 2015: 1456-1462.
- [14] Pierleoni P, Belli A, Palma L, et al. "A high reliability wearable device for elderly fall detection," *IEEE Sensors Journal*, 2015, 15(8): 4544-4553.
- [15] Liu S H, Cheng W C., " Fall detection with the support vector machine during scripted and continuous unscripted activities," *Sensors*, 2012, 12(9): 12301-12316.
- [16] Shibuya N, Nukala B T, Rodriguez A I, et al., "A real-time fall detection system using a wearable gait analysis sensor and a support vector machine (SVM) classifier," *Mobile Computing and Ubiquitous Networking (ICMU)*, 2015 Eighth International Conference on. IEEE, 2015: 66-67.
- [17] Sweldens, Wim. "The Lifting Scheme: A Construction of Second Generation Wavelets," *Journal on Mathematical Analysis*. SIAM. 1997, 29 (2): 511-546.
- [18] Wójtowicz B, Dobrowolski A, Tomczykiewicz K., "Fall detector using discrete wavelet decomposition and SVM classifier," *Metrol. Meas. Syst*, 2015, 22(2): 303-314.
- [19] Isu Shin, Jongsang Son, Soonjae Ahn, Jeseong Ryu, Sunwoo Park, Jongman Kim, Baekdong Cha, Eunkyong Choi, and Youngho Kim, "A Novel Short-Time Fourier Transform-Based Fall Detection Algorithm Using 3-Axis Accelerations," *Mathematical Problems in Engineering*, 2015, Article ID 394340.
- [20] Hsu C W, Chang C C, Lin C J. *A practical guide to support vector classification*, 2003.
- [21] Cortes C, Vapnik V., "Support-vector networks," *Machine Learning*, 1995, 20(3): 273-297.
- [22] Chang C C, Lin C J., " LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, 2011, 2(3): 27.
- [23] Altman D G, Bland J M., "Diagnostic Tests : Sensitivity and specificity," *British Medical Journal*, 1994, 308(6943): 1552.
- [24] Kwolek B, Kepski M., "Human fall detection on embedded platform using depth maps and wireless accelerometer," *Computer Methods and Programs in Biomedicine*, 2014, 117(3): 489-501.