Identification of Soft Falls based on Falling State Occurrences

Thein Gi Kyaw[1], Anant Choksuriwong[2], and Nikom Suvonvorn[3]

ABSTRACT: Fall detection techniques for helping the elderly have been developed based on identifying falling states using simulated falls. However, some real-life falling states were left undetected, which led to this work on analysing falling states. The aim was to find the differences between active daily living and soft falls (syncope) where falling states were undetected. Our new fall detection method is based on threshold-based algorithms using acceleration data stored in an activity database. This study addresses soft falls in addition to general falls based on two falling states. Despite the number of false alarms being higher, rising up to 28%, the accuracy and sensitivity were increased by up to 71%, and 66.5%, respectively. Our experimental results show the importance of state occurrence for soft fall detection, and could be used to build a learning model for soft fall detection.

Keywords: Accelerometer, Soft Falls, Activity of Daily Living (ADLs), Falling States, Threshold-based Approach

DOI: 10.37936/ecti-cit.2021153.240720

Article history: received March 12, 2020; revised August 30, 2020; accepted September 3, 2020; available online November 12, 2021

1. INTRODUCTION

Falling is a significant danger for the elderly. Approximately 30-50% of people living in long term care institutions fall each year, and 40% of them experience recurrent falls [1]. The effect of medication can generate dizziness or weakness in the balance mechanism [2]. Gait disorders, syncope and dizziness, and sudden falls without a loss of consciousness account for 17%, 13%, and 9% of falls respectively [3]. For elderly people living alone, the number of non-injured fallers is around 47% [4]. Those who remain unhelped for a long time on the floor could die within six months [5]. In these cases, the elderly people describe the fall as loss of balance while doctors lead the search to provide the optimal treatment for this ill health. The definition of a fall is not only difficult but also important.

Reliable fall detection is essential within independent living facilities for the elderly. By continuously monitoring their activities, a fall can be quickly detected and an alarm raised. Various fall detection systems have been proposed using floor vibration sensors [6], acoustic sensors [7], cameras [8], Wi-Fi [9], and wearable accelerometer sensors [10, 11]. Most of these approaches are localized and expensive, suffer from privacy problems, and have a limited range. On the other hand, low-cost wearable sensors can be used everywhere for monitoring the elders.

Algorithms for rapid detection have employed approaches based on thresholds [12] and machine-learning [13, 14]. Hybrid approaches [9] have applied a classification scheme based on falling states consisting of a pre-impact stage, fall impact, horizontal orientation following a fall (post-impact), and static motion after the fall. Most real-time fall detection methods use a wireless tri-axial accelerometer, gyroscope, and magnetometer together with other sensors used for sensing on one or more of these falling states [12-13, 15-21].

Pannurat [12] utilized wireless tri-axial accelerometers placed on different parts of the body to detect the phase of a fall with accuracies of 86.54%, 87.31%, and 91.15% for the pre-impact, impact, and post-impact stages respectively. However, some high impact daily activities were misclassified as falls, and mounting sensors at certain body positions offer less performance than the thigh and ankle in terms of sensitivity. Nevertheless, this work offered insights into how to improve performance in terms of specificity and accuracy.

Kangas [15] has proposed three threshold-based

[1,2,3]The authors are with Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University, HatYai, Songkhla, Thailand., E-mail: theingik6@gmail.com, anant.c@psu.ac.th and nikom.suvonvorn@gmail.com

fall detection algorithms based on multiple falling states. They use tri-axial accelerometer sensors attached to multiple body locations including the waist, head, and wrist. He noted that the start of a fall may be undetected because of a lower velocity than the predefined threshold represented by a person falling first to their knees, and followed by an impact to their waist or hands. Also, there was minimal impact at the waist level for backward falls that ended in a soft landing with a rounded back. However, backward falls were still correctly detected based on head position information.

Bourke [16] developed and tested a threshold-based detection algorithm using tri-axial accelerometer sensors attached to a person's trunk and thigh, and utilizing two falling phases (pre-impact, and impact). Lower impact values were recorded in real life compared to when elderly people lost control which led to their fall, highlighting the differences between real and simulated falls.

Lee [17] proposed a novel fall detection method based on a pre-impact falling phase using vertical velocity and a separate two method comparison for near-falls. The former used accelerometers and gyroscopes, and the latter applied accelerometers only. In certain cases stand-to-sit falls were misidentified as activities of daily living (ADLs) because the vertical velocity did not reach a predefined threshold value. Moreover, stand-to-lie ADL was sometimes mislabeled as falls because of the fast downward velocity of the pre-impact falling state.

Pierleoni [18] used a three-axis accelerometer, a three-axis gyroscope, a three-axis magnetometer, and a barometer sensor, all embedded in a waist-mounted device, which discriminated between falls, falls with recovery, and ADL, by using threshold-based adopted testing protocols based on three falling states (e.g. impact, post-impact (static motion), and body orientation) to solve the problem of identifying the backward falls ending in a sitting position without an orientation change or syncope. This work reduced the number of false negatives by using information about the orientation and altitude signal. In particular, it excluded forward falls involving an attempt to break the fall, and falls involving leaning against a wall and slipping to the floor that end with sitting. This work proves that fusing data from combinations of many sensors leads to higher performance.

Bagala [19] examined thirteen algorithms for detecting falling states, and found that some of the falling states detected in simulated falls did not correspond to real-world falls. Also, the performance of fall detection approaches in real life conditions was lower than that for simulated falls, which was also a conclusion by Bourke [16].

Some of the falling states detected in simulated falls did not correspond to real-world soft falls. This is a reflection of how it is hard to detect these falling states in different situations due to factors such as number of sensors, positions, and number of falling states. However, if the differences between soft falls and daily activities are known, then the fall detection algorithms and test plans can take these differences into consideration to achieve better performance.

No previous study has focused on the occurrence of falling states to detect soft falls with missing falling states. This raises the question: what are the differences between these soft falls and daily activities?

The main contribution of this paper is to focus on the occurrence of falling states to find the differences between ADLs and soft falls where falling states were not detected. Specific considerations are:

(1) To detect the full occurrences and number of both falling states' occurrences so that general falls and ADLs can be differentiated.

(2) To detect the full occurrence of one of the falling states so that soft falls can be classified separately from ADLs.

This requires the development of a new type of fall, and the use of false alarms during the detection. Then, analysing the occurrence and number of falling state occurrences, and differences between soft falls and ADLs could provide the prior knowledge needed to design the learning model proposed in a previous study [22] for soft fall detection.

2. SYSTEM DESIGN AND IMPLEMENTATION

When a fall occurs, four falling states can be identified as loss of balance, unbalanced, posture changes, and motionless. Loss of balance state occurs at the start of the fall, before impact. Unbalanced state appears when a body impacts the ground, and the posture changes from vertical to horizontal. After the fall, the motionless state begins.

The lack of falling state occurrences in real falls [15-17], and their appearance in ADLs, such as jumping, causes the fall detection rate to decrease. In this work, only the first two falling states were analyzed.

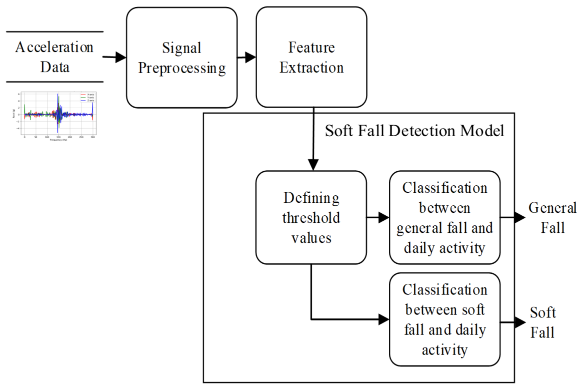

Fig. 1: *Fall detection using falling states.*

Fig. 1 shows the offline training, testing, and analysis phases of our detection model. In the training phase, data from the fall dataset is read, preprocessed, and features are extracted. In addition, the first threshold values for the state occurrence and the number of falling state occurrences used to discriminate general falls from ADLs are defined. Secondary threshold values are used to separate soft falls from ADLs. During the testing phase, the falling states will be detected.

2.1 Acceleration Data

Although the related methods were applied to data from different wearable sensors, the acceleration data collected by the smartphones in a pocket were only applied to find the differences among general falls, soft falls, and ADLs.

2.1.1 The Real-world Daily Activity Database

The tFall dataset [23] consists of acceleration signal samples collected from sensors embedded in smartphones located in pockets and handbags, at 50Hz. Daily activities were recorded under real-life conditions. Eight falls from ten participants, were collected with each one lasting 6 seconds, and repeated three times. We utilize the pocket position records consisting of 7816 and 503 samples of ADLs and falls respectively. These were reclassified into two types of falls: as general falls, and syncope (soft fall). This allowed the tFall dataset to be applied to detect soft falls in our work.

2.1.2 MobiFall

The MobiFall dataset [24] is made up of accelerometer and gyroscope phone data sampled at 87 Hz and 200 Hz. 630 examples from 11 volunteers are classified into falls and ADLs. Even though soft falls were not included in this dataset, we utilized this dataset to find the differences between high impact ADLs, such as jogging, and general falls, and verify whether ADLs may be confused with soft falls or not.

2.2 Signal Preprocessing

We determined that a sample file of duration 6 seconds would be sufficient to detect falling states during a fall. As a consequence, MobiFall sample files were segmented into that length centered around the highest peak. Before the adopted features were extracted, the data was passed through a median filter of length 3, and high-pass and low-pass filtered (25Hz) with a digital second order Butterworth filter. This removed noise and extracted static and dynamic motions [25].

2.3 Feature Extraction

After the noise had been filtered from the raw tri-axial acceleration data, a signal magnitude vector (SMV) was extracted within each 1 second time window. Its computation was adopted from the study [26] and is expressed as:

$$\mathrm{SMV} = \sqrt{x^2 + y^2 + z^2} \tag{1}$$

This SMV is made up of acceleration data taken from the x-, y-, and z-axes. The resulting SMV values for syncope type falls are shown in Fig. 2.

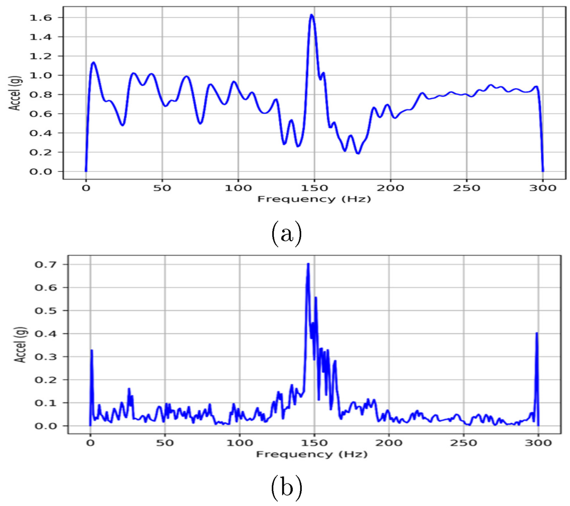

Fig.2: *Examples of different SMV values from low-pass filters (a), and high-pass filters (b) for a syncope fall with an impact and a duration of around 6 seconds.*

This SMV feature is related to a Loss of Balance (LOB), which occurs during the fall. It occurs during the descending phase, before an impact, and is related to the lower fall peak. A SMV value was calculated from the low-pass filter data as shown in Fig. 2(a). The value decreased to an acceleration of under 0.5 g before impact, and its minimum, called SMVmin, was used to detect the LOB state for each ADL or fall.

In addition, an unbalanced (UB) state occurs when a body hits the ground during a fall, and is related to the upper fall peak; its SMV value was calculated from the high-pass filter data, as shown in Fig. 2 (b). The value increased during the fall and was around $\sim$0g before and after the fall. Its maximum, called SMVmax, was used to detect the UB state.

2.4 The Soft Fall Detection Model

Two falling states are detected in the soft fall detection model with threshold values.

2.4.1 Defining Threshold Values for checking the falling state occurrences

The calculated SMVmin and SMVmax were compared with the threshold values to detect both LOB and UB states during general fall detection. For LOB state monitoring, the SMVmin values for both fall and ADL groups were utilized: the mean of each group was calculated, and their average used as the threshold LOB_th. As a consequence, if a signal was less than the LOB_th, then the LOB state occurred. Otherwise, it did not.

For UB state detection, the mean SMVmax values from both the fall and ADL groups were averaged to produce the threshold UB_th. If the signal was greater than the UB_th, then the UB state occurred, otherwise, it did not.

2.4.2 Defining Threshold Values for counting the number of falling state occurrences

Both falling states' occurrence conditions were considered for general fall detection, but high-impact ADLs, such as jogging satisfy these conditions. To discriminate between high-impact ADLs and general falls, the number of falling states, numFallState, was counted in the windows, in a way similar to the analysis of high-impact ADLs [27].

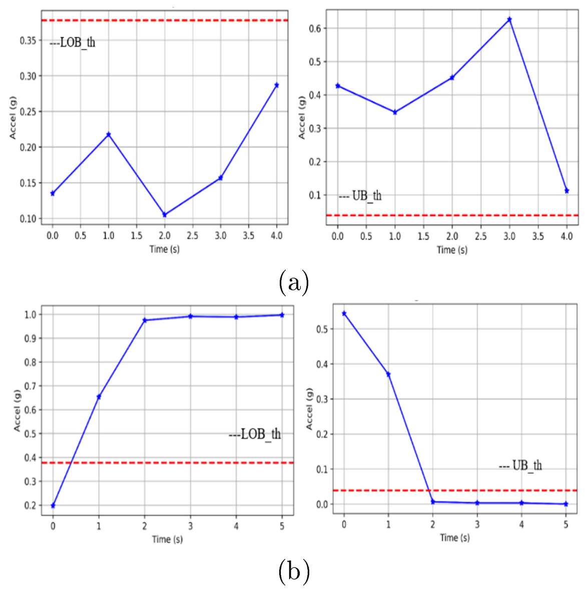

Fig.3: *SMV examples in the LOB and UB states for a) high-impact ADL (jogging), b) a sideward-lying fall.*

Fig. 3 (a, b) gives SMV values for both states of a high impact ADL involving jogging and a sideward-lying fall. Both states occur in all the windows for jogging with the numFallState (numLOB and numUB) at 100%. However, for the fall, the numFallState is less than 50%. Even though both falling states' occurrence conditions can be found in both activities, the numFallState value can be used to discriminate between them.

SMV values for both states for a chair-sitting ADL and a back sitting chair fall are shown in Fig. 4 (a, b), with the LOB state disappearing and the UB state occurring in both activities. The same numUB value (40%) was calculated for all the windows. Therefore, if both falling states did occur, then the number of state occurrences would be compared with the thresholds, where the maximum number of falls was set at the thresholds. The thresholds were calculated as follows:

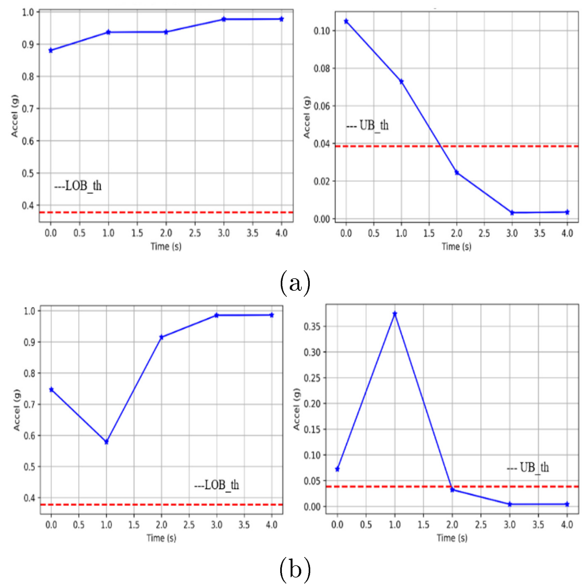

Fig.4: *SMV examples in the LOB and UB states for a) a chair-sitting ADL, and b) a back sitting chair fall.*

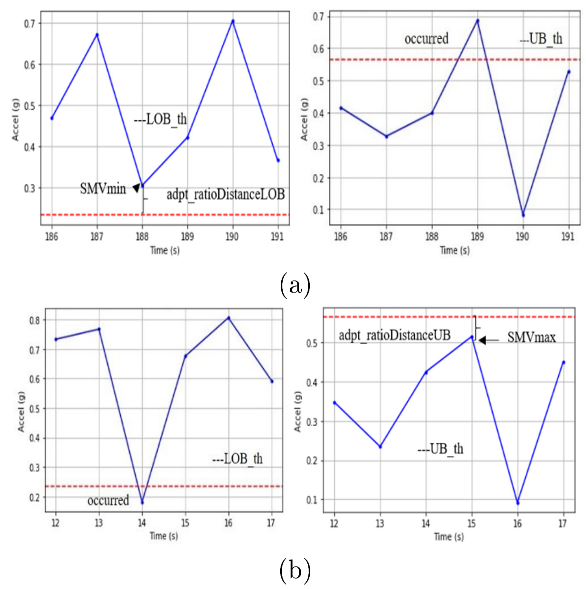

Fig.5: *Computation distance between LOB_th and UB_th for a) the LOB state, and b) the UB state. (Dashed lines show the threshold values for each state).*

$$\text{numLOB\_Percent} = \text{numLOB}/\text{totalWindow} \quad (2)$$

$$\text{numUB\_Percent} = \text{numUB}/\text{totalWindow} \quad (3)$$

$$\text{numLOBTh} = \max(\text{numLOB\_Percent\_fall}) \quad (4)$$

$$\text{numUBTh} = \max(\text{numUB\_Percent\_fall}) \quad (5)$$

The totalWindow value is the total number of windows, and numLOB_Percent_fall is the numLOB_Percent value for all the fall signals.

2.4.3 Defining Adaptive Threshold Values for checking the falling state occurrences

The adaptive ratio of distances was used to identify soft falls, as shown in Fig. 5. Fig. 5 (a) shows how the syncope fall signal with a disappearing LOB state and a UB state occurrence were found. In Fig. 5 (b), a fall signal includes a LOB state and the UB state disappears. In these cases, a state disappears when its signal does not reach the threshold. The adaptive distance ratio based on the threshold was calculated for each state, making it possible to distinguish between soft falls and ADL signals.

Some soft falls would be undetected if the minimum fall distance was chosen as a threshold, so the maximum distance of fall groups was selected instead, and the signals were checked using adaptive distance thresholds. The equations employed to calculate the adaptive ratios of the distance and adaptive distance thresholds for each state were defined as follows:

$$\text{adpt\_ratioDistanceLOB} = ((\text{SMVmin} - \text{LOB\_th})/\text{LOB\_th}) \times 100 \quad (6)$$

$$\text{adpt\_LOB\_Th} = \max(\text{adpt\_ratioDistanceLOB\_fall}) \quad (7)$$

$$\text{adpt\_ratioDistanceUB} = ((\text{UB\_th} - \text{SMVmax})/\text{UB\_th}) \times 100 \quad (8)$$

$$\text{adpt\_UB\_Th} = \max(\text{adpt\_ratioDistanceUB\_fall}) \quad (9)$$

The adpt_ratioDistanceLOB_fall and adpt_ratio DistanceUB_fall are the adaptive distance ratios for the LOB and UB states for a fall group.

2.5 CLASSIFICATION

Two Soft Fall Detection algorithms based on the Occurrence of falling States (SFDOS) (a, b) were employed in our model, as shown in Fig. 6. SFDOS (a) was used to discriminate between general falls and ADLs, where SFDOS (b) distinguished general falls and soft falls from ADLs. The LOB and UB states each have possible values for falling state occurrence and disappearance, allowing four possible conditions to be defined, as shown in Fig. 7.

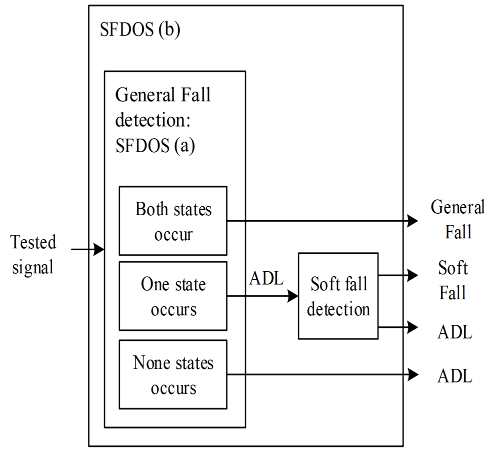

Fig. 6: *Soft Fall Detection.*

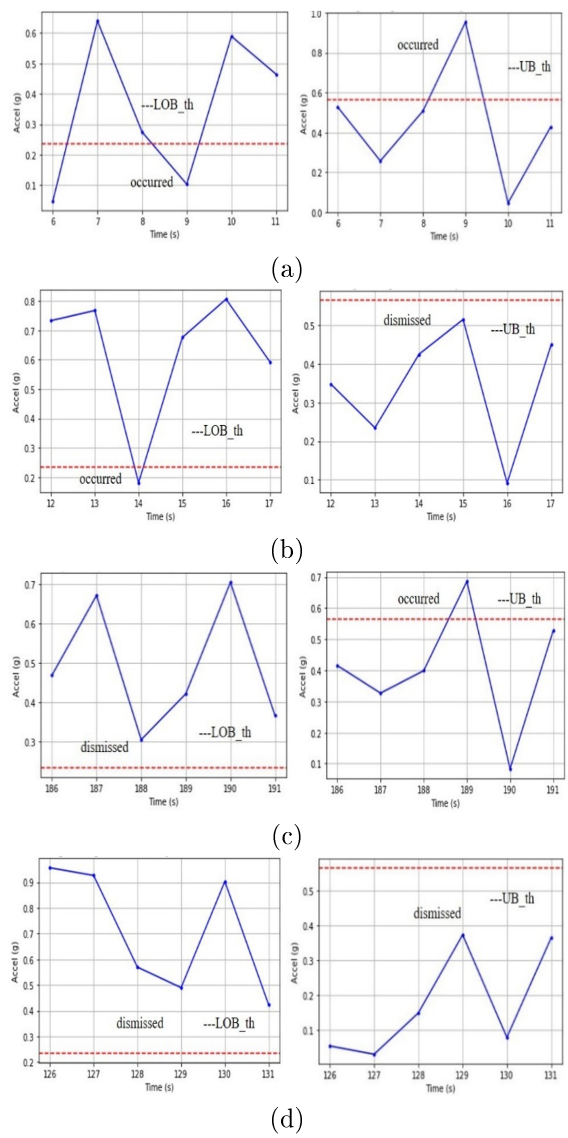

Fig. 7: *Examples of four patterns found during syncope fall.*

```
SFDOS (b)

If SMVmin < LOB_th and SMVmax > UB_th Then
        fall = detected
Else If SMVmin < LOB_th and SMVmax < UB_th Then
        Call function calculateRatioDistanceUB with input SMVmax, UB_th
        calculateRatioDistanceUB returns adpt_ratioDistanceUB
        If adpt_ratioDistanceUB < adpt_UB_Th Then
                fall = detected
        End If
Else If SMVmin > LOB_th and SMVmax > UB_th Then
        Call function calculateRatioDistanceLOB with input SMVmin, LOB_th
        calculateRatioDistanceLOB returns adpt_ratioDistanceLOB
        If adpt_ratioDistanceLOB < adpt_LOB_Th Then
                fall = detected
        End If
End If
```

Fig.8: *Soft Fall Detection Algorithm.*

As seen in Fig. 6, SFDOS (a) is based on a consideration of occurrences of both falling states. In general fall detection, a signal like the one shown in Fig. 7 (a) displays occurrences of both falling states. In the other cases, this signal will be identified as ADL. The signals shown in Fig. 7 (b, c) display just a single falling state occurrence, and so are considered to be soft fall. Signals with no falling state occurrences, as in Fig. 7 (d), are identified as ADLs. SFDOS (b) detects a soft fall by adding in a consideration of one falling state occurrence to SFDOS (a). If the ratios of the distances of the signals in Fig. 7 (b, c) are less than the adaptive distance thresholds adpt_LOB_Th and adpt_UB_Th, then they are identified as soft falls. Otherwise, these signals will be classified as ADLs. The pseudo code for SFDOS (b) is described in Fig. 8.

3. RESULTS AND DISCUSSION

To distinguish general and soft falls from ADLs based on the LOB and UB states, two conditions were considered. The first condition discriminates between general falls and ADLs by using both falling states' occurrences in SFDOS (a). The second condition separates soft falls from undetected falls and ADLs based on only single falling state occurrence. Both conditions are included in SFDOS (b).

All of the signals from both datasets were analysed in terms of falling state occurrences. In order to study the effect of new dataset, the average performance was calculated after eight-fold cross-validation. Seven falls and 90% of ADL samples were utilized in the training, and one fall and 10% of ADL samples of the tFall dataset and all samples of the MobiFall dataset were utilized in the testing.

In tFall, the number of signals having occurrences of both falling states is the lowest for ADLs, and the highest for general and soft falls. Therefore, both falling states' occurrences were considered in SFDOS (a). In contrast, the number of signals having no falling state occurrences is the highest for ADLs, and lowest for general and soft falls. Therefore, no falling state occurrence was not considered in soft fall detection, leaving only signals with one falling state occurrence. However, in the MobiFall dataset, a single UB state occurrence is mostly found in ADLs, which shows how considering just falling state occurrence can confuse ADLs and falls. This issue was solved by applying numFallState because jogging ADLs often register UB state occurrences, as shown in Fig. 3. When both falling states' occurrence condition is met, and the numFallState value for tested signals is lower than the thresholds, then these signals are identified as falls.

To identify differences between undetected soft falls and ADLs, the ratios of the distances for all the activities were calculated. The maximum, minimum, and mean distance values were extracted to test the condition that the tested signals had one falling state occurrence, as shown in Tables 1 and 2.

The minimum ratio distances for soft falls in the tFall dataset are the highest among all the activities, and the minimum ratio distances of the ADLs are the lowest. This leads to confusion between soft falls and ADLs because the minimum distance ratio for falls is higher than the maximum distance ratio of ADLs. In contrast in MobiFall, the maximum and mean ratios distance of ADLs are higher than those of falls, which leads to confusion between general falls and high-impact ADLs.

Table 1: *Ratio of distance from UB_th of signals having a LOB state occurrence.*

Dataset	Activity	Maximum distance (%)	Minimum distance (%)	Mean distance (%)
tFall	GF	43.45	0.06	3.68
	SF	34.33	1.7	15.88
	ADL	0.06	0.01	0.03
MobiFall	GF	20.7	0	0.07
	ADL	88.5	0	0.59

GF = general fall, SF = soft fall, ADL = activity of daily living

Table 2: *Ratio of distance from LOB_th of signals having a UB state occurrence.*

Dataset	Activity	Maximum distance (%)	Minimum distance (%)	Mean distance (%)
tFall	GF	149	0.24	3.9
	SF	179	30.2	74.6
	ADL	0.24	1.37e-5	0.08
MobiFall	GF	149	0	21
	ADL	167	0	53.6

As a consequence, if the maximum fall values were set as thresholds, some of the ADLs would be wrongly classed as falls, and if the minimum fall values were used instead then some falls would be missed. This led to our use of adaptive distance threshold values stored in adpt_LOB_Th and adpt_UB_Th as the maximum fall values for each state, which are shown in Tables 1 and 2. This meant that a LOB fall state having a UB state occurrence was detected using adpt_LOB_Th, and UB fall states having a LOB state occurrence were observed by applying adpt_UB_Th. In general, if one falling state disappears and the distance ratio of the tested signal is less than the threshold, then a fall will be identified.

To determine the pattern of falling states for activities, the distance ratios of the tested signals were compared with the adaptive distance thresholds. Fig. 9 lists the comparison of all the falls to indicate how the testing of the new dataset effects the performance. It shows that the performance of SFDOS (a) could be degraded when different types of falls occur in real life. When the proposed soft fall detection model SFDOS (a, b) was combined with the numFallState value, soft falls were better detected.

Fig. 10 presents information about the percentage of false alarms generated in ADLs, and the sensitivity (SE) rates before and after evaluating SFDOS with falling state occurrences by utilizing testing samples from the tFall dataset and all samples from the MobiFall datasets. Overall, the results indicate that the average SE is increased by up to 66.5%. Although the number of falls detected is increased, the average accuracy is increased by up to 71%, and the average number of false alarms also increases by up to 28%. It shows that SFDOS (b) increases the sensitivity, and the number of false alarms for one falling state occurrence. However, SFDOS (a, b) combined with the numFallState value reduces the number of false alarms for both falling states' occurrences. Although SFDOS (a) combined with the numFallState value has fewer false alarms than SFDOS(b) combined with the numFallState value as shown in the Fig. 10(b), the SFDOS (b) combination is better in real-life conditions when different falls and ADLs are detected.

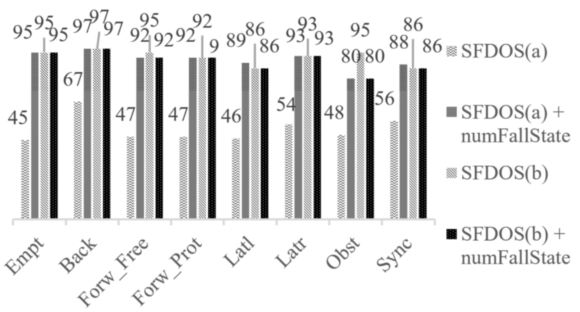

Fig.9: *Sensitivity (%) of the algorithms to study the effect of different falls.*

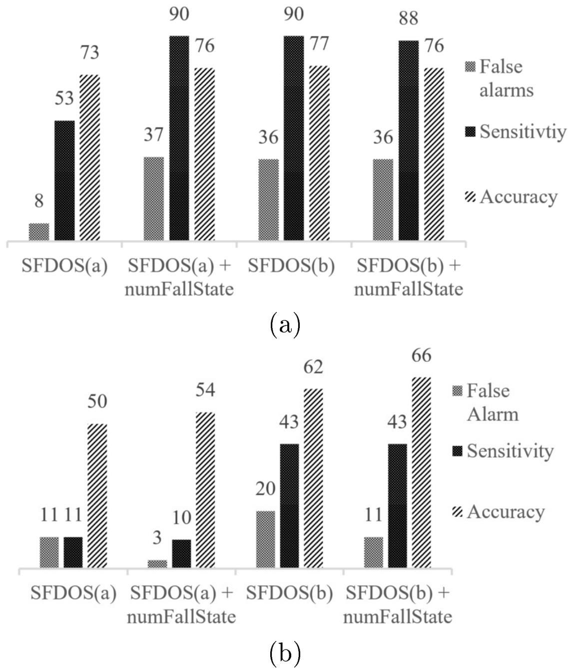

Fig.10: *Performance (%) using the a) tFall, and b) MobiFall datasets.*

Although our work increases the performance of SFDOS in terms of sensitivity and accuracy, it only employs two falling states. If other falling states were added, the detection performance could be increased further in terms of specificity, as considered in [12]. Also, knowing the types of daily activities, and including more types of falls could better explain the differences between falls (general falls and soft falls) and ADLs.

Our work uses acceleration data from a single sensor, while some other approach [17] utilized multiple sensors to handle syncope falls. These soft falls can be detected by adding two features utilizing a single sensor with a higher performance. Therefore, our work suggests that decreasing the number of sensors may not be a disadvantage.

4. CONCLUSIONS

The main contribution of this paper is to develop a technique to identify soft falls even when some of their associated falling states were not detected in real life. Two classification techniques were tested upon the tFall and MobiFall datasets.

This study showed that SFDOS (b) combined with the numFallState value increased fall detection accuracy by up to 71%. We recommend the testing of the algorithms for analysing falling state occurrences, especially SFDOS (b) combined with the numFallState value, which should be trained and tested with real soft falls and activities. Also, more falling states should be detected with the aim of reducing the similarity between them. To the best of our knowledge, there is no analysis on falling state occurrences for fall detection algorithms under real-life conditions.

As a result, a single sensor may not efficient enough to present sufficient data about soft falls, although our approach does not affect a user's daily life. Therefore, a multi-sensors fusion technique will be utilized in our future work. The results from this current work could act as prior knowledge to build the learning model.

ACKNOWLEDGMENT

This work was supported by the Higher Education Research Promotion and the Thailand's Education Hub for Southern Region of ASEAN Countries Project Office of the Higher Education Commission.

References

[1] World Health Organization, "*WHO Global Report on Falls Prevention in Older Age.*," Community Health (Bristol)., Geneva: World Health Organization, 2007.

[2] A. Stinchcombe, N. Kuran, and S. Powell, "*Seniors' Falls in Canada*," Public Health Agency of Canada, Vol. 34, No. 2–3, pp. 171-174, Jul. 2014.

[3] S. R. Lord and C. Sherrington, "Falls in older people risk factors and strategies for prevention," *Aging and Society*, vol. 21, no. 5, pp. 667-675, 2001.

[4] M. E. Tinetti, W. L. Liu, and E. B. Claus, "Predictors and prognosis of inability to get up after falls among elderly persons," *JAMA J. Am. Med. Assoc.*, vol. 269, no. 1, pp. 65-70, 1993.

[5] D. Wild, U. S. L. Nayak, and B. Isaacs, "How dangerous are falls in old people at home?," *Br. Med. J. (Clin. Res. Ed).*, Vol. 282, No. 6260, pp. 266-268, 1981.

[6] D. Litvak, I. Gannot, "Detection of falls at home using floor vibrations and sound," in *2008 IEEE 25th Convention of Electrical and Electronics Engineers in Israel (Ieeei 2008)*, pp. 514-518, 2008.

[7] M. Cheffena, "Fall detection using smartphone audio features," *IEEE J. Biomed Health Inform*, vol. 20, no. 4, pp. 1073-1080, 2016.

[8] M. Kepski and B. Kwolek, "Fall detection using ceiling-mounted 3D depth camera," in *Int. Conf. on Computer Vision Theory and Applications (VISAPP)*, vol. 2, pp. 640-647, 2013.

[9] X. Yang, F. Xiong, Y. Shao, and Q. Niu, "WmFall: WiFi-based multistage fall detection with channel state information," *Int. J. Distrib. Sens. Networks*, vol. 14, No. 10, 2018.

[10] A. T. Ozdemir and B. Barshan, "Detecting falls with wearable sensors using machine learning techniques," *Sensors*, vol. 14, no. 6, pp.10691-10708, 2014.

[11] O. Kerdjidj, N. Ramzan, K. Ghanem, A. Amira, and F. Chouireb, "Fall detection and human activity classification using wearable sensors and compressed sensing," *J. Ambient Intell. Humaniz. Comput.*, pp. 1-13, 2019.

[12] N. Pannurat, S. Thiemjarus, and E. Nantajeewarawat, "A hybrid temporal reasoning framework for fall monitoring," *IEEE Sens. J.*, vol. 17, no. 6, pp. 1749-1759, 2017.

[13] D. Genoud and V. Cuendet, "Soft fall detection using machine learning in wearable devices," in *2016 IEEE 30th Int. Conf. on Advanced Information Networking and Applications (AINA)*, pp. 501-505, 2016.

[14] F. Hussain, F. Hussain, M. Ehatisham-Ul-Haq, and M. A. Azam, "Activity-Aware Fall Detection and Recognition Based on Wearable Sensors," *IEEE Sens. J.*, vol. 19, no. 12, pp. 4528–4536, 2019.

[15] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jämsä, "Comparison of low-complexity fall detection algorithms for body attached accelerometers," *Gait Posture*, vol. 28, no. 2, pp. 285-291, 2008.

[16] 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, 2007.

[17] J. K. Lee, S. N. Robinovitch, and E. J. Park, "Inertial sensing-based pre-impact detection of falls involving near-fall scenarios," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 23, no. 2, pp. 258-266, 2015.

[18] P. Pierleoni et al., "A wearable fall detector for elderly people based on AHRS and Barometric sensor," *IEEE Sens. J.*, vol. 16, no. 17, pp. 6733-6744, 2016.

[19] F.Bagala, C. Becker et al., "Evaluation of accelerometer-based fall detection algorithms on real-world falls," *PLoS One*, vol. 7, no. 5, pp, 1-9, 2012.

[20] P. V. Er and K. K. Tan, "Non-intrusive fall detection monitoring for the elderly based on fuzzy logic," *Meas. J. Int. Meas. Confed.*, vol. 124, July 2017, pp. 91–102, 2018.

[21] P. Pierleoni, A. Belli, L. Palma, M. Pellegrini, L. Pernini, and S. Valenti, "A High Reliability Wearable Device for Elderly Fall Detection," *IEEE Sens. J.*, vol. 15, no. 8, pp. 4544–4553, 2015.

[22] T. Kyaw, A. Choksuriwong, N. Suvonvorn, "Soft Fall Detection using Multi-sensors," *Asia Pacific Conference on Robot IoT System Development and Platform 2018 (APRIS2018)*, pp. 52–53, 2018.

[23] C. Medrano, R. Igual, I. Plaza, and M. Castro, "Detecting falls as novelties in acceleration patterns acquired with smartphones," *PLoS One*, vol. 9, no. 4, 2014.

[24] G. Vavoulas, M. Pediaditis, E. G. Spanakis, and M. Tsiknakis, "The MobiFall dataset: An initial evaluation of fall detection algorithms using smartphones," *13th IEEE Int. Conf. Bioinformatics and Bioengineering*, pp. 1–4, 2013.

[25] M. Kangas, A. Konttila, I. Winblad, and T. Jämsä, "Determination of simple thresholds for accelerometry-based parameters for fall detection," *29th Annu. Int. Conf. IEEE Engineering in Medicine and Biology (IEEE EMBS)*, pp. 1367–1370, 2007.

[26] E. Casilari, J. A. Santoyo-Ramón, and J. M. Cano-García, "Analysis of public datasets for wearable fall detection systems," *Sensors (Switzerland)*, vol. 17, no. 7, 2017.

[27] B. Cates, T. Sim, H. M. Heo, B. Kim, H. Kim and J. H. Mun, "A novel detection model and its optimal features to classify falls from low- and high-acceleration activities of daily life using an insole sensor system," *Sensors (Switzerland)*, vol. 18, no. 4, 2018.

Thein Gi Kyaw was born in Yenangyaung, Magway, Myanmar in 1986. She received B.C.Sc and B.C.Sc(Hons:) degrees in computer science from the Computer University, Magway, Myanmar in 2005, 2006, and the M.C.Sc degree from Computer University, Maubin, Myanmar, in 2010. She is a Ph.D candidate in Prince of Songkla University (PSU), Hat Yai, Songkhla, Thailand. Her research interest includes Machine learning.

Anant Choksuriwong received Diploma, Master, Ph.D. degrees in 2000 (PSU), 2003 (UJF), 2004 (INPG) and 2008 from the School of Engineering in ENSI de Bourges. He is researcher at iSys Laboratory of Computer Engineering, Prince of Songkla University (PSU), Songkhla, Thailand, and lecturer at department of Computer Engineering, PSU, who teaches courses in Advance Image Processing, Machine Learning and Principle of Robotics.

Nikom Suvonvorn obtained DEA form l'Université de Paris Sud (XI) in 2003, and received PhD in Computer Science from l'Université de Paris Sud (XI), Orsay, France in 2006. He is currently a lecturer at Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University, Hatyai, Thailand. His research concerns the computer vision, image processing, and its related applications.