A data-driven approach for online pre-impact fall detection with wearable devices

Takuto Yoshida, Kazuma Kano, Keisuke Higashiura, Kohei Yamaguchi,
Koki Takigami, Kenta Urano, Shunsuke Aoki, Takuro Yonezawa,
Nobuo Kawaguchi

Abstract The implementation of wearable airbags to prevent fall injuries depends on accurate pre-impact fall detection and a clear distinction between activities of daily living (ADL) and them. We propose a novel pre-impact fall detection algorithm that is robust against ambiguous falling activities. We present a data-driven approach to estimate the fall risk from acceleration and angular velocity features, and use thresholding techniques to robustly detect a fall before impact. In the experiment, we collect simulated fall data from subjects wearing an inertial sensor on their waist. As a result, we succeeded in significantly improving the accuracy of fall detection from 50.00% to 96.88%, the recall from 18.75% to 93.75%, and the specificity 81.25% to 100.00% over the baseline method.

1 Introduction

It is an important issue to prevent injuries due to falls among the elderly. Falls can lead directly to death, or may cause the elderly to become bedridden that may indirectly lead to death. Psychological trauma and fear of falling can cause anxiety and lack of confidence in performing daily activities, thus preventing independence. One-third to one-half of seniors aged 65 and older experience a fall [1]. About one-third of the elderly who live at home fall at least once a year. About half of the elderly living in nursing homes fall at least once a year. People who have fallen before are more

Takuto Yoshida
Graduate School of Enginnering, Nagoya University, e-mail: takuto@ucl.nuee.nagoya-u.ac.jp

Kazuma Kano, Keisuke Higashiura, Kohei Yamaguchi, Koki Takigami, Kenta Urano,
Takuro Yonezawa, Nobuo Kawaguchi
Graduate School of Enginnering, Nagoya University

Shunsuke Aoki
National Institute of Informatics

likely to fall again. Of those who fall and remain lying on the floor for hours, half will be dead within half a year [2].

Many researchers have study of fall detection and its prevention. The key points of the fall detection task are to reliably detect falls and to distinguish between falls and activities of daily living (ADL). There are three main categories of fall detection methods: ambient device based, vision based and wearable device based [3]. Ambient device based system is a method of recognizing human activity using video, audio, and vibrational data [4, 5]. Vision based system detects the position and posture of a person and estimates their activity based on images from ceiling-mounted cameras [6, 7]. However, those methods are costly and not easy to set up. Additionally, it is difficult to use them for online pre-impact fall detection to prevent fall injuries. On the other hand, wearable-based systems are easy to implement and various methods have been proposed [8, 9, 10, 11, 12, 13, 14, 15, 16]. In most of these methods, acceleration and angular velocity are the key features for detecting falls, and are applied using threshold detection algorithms. In [17, 18, 1], the approaches for online pre-impact fall detection to prevent fall injuries have been proposed. In general, the system should be able to detect a fall 0.3 to 0.4 s before impact, assuming that it is incorporated into a wearable airbag. Most existing methods use thresholds to perform binary classification of fall events (the timing of airbag activation) and ADL events, and achieve high detection rates in experiments with some restricted activities. However, in real life, there are many ambiguous falling activities, such as ADL activities that resemble falls and fall activities that resemble ADLs. For example, sitting on a chair vigorously, jumping, falling with a defensive response and so on. It is difficult to distinguish them by only using instantaneous acceleration and angular velocity values and thresholds. Since falls are life-threatening, there is a need for more robust fall detection algorithms.

We propose the novel pre-impact fall detection algorithm that is robust against ambiguous falling activities to prevent fall injuries. Figure 1 shows an overview of the algorithm for detecting a fall before impact based on acceleration and angular velocity. Our proposed method uses the data-driven human activity recognition (HAR) technique to improve the robustness of fall detection. First, we extract the features of acceleration and angular velocity using a sliding window, and then perform feature reduction using principal component analysis (PCA) or sequential forward feature selection (SFFS). Secondly, we use a machine learning model, bayesian decision making (BDM), the least-squares method (LSM), the k-nearest neighbor algorithm (k-NN), support vector machines (SVM), and artificial neural networks (ANN), to estimate the fall risk, which is a value that models the likelihood of a fall in the time direction. Finally, at the timing when the fall risk exceeds the threshold, the airbag activation signal is output. In our approach, we set the threshold to detect 0.3 s before the fall that is the optimal time to activate the airbag. In the experiment, we collect simulated fall data including ambiguous falling activities from subjects wearing an inertial sensor on their waist. Our contributions are as follows:

- We present a novel pre-impact fall detection algorithm that is robust against ambiguous falling activities.

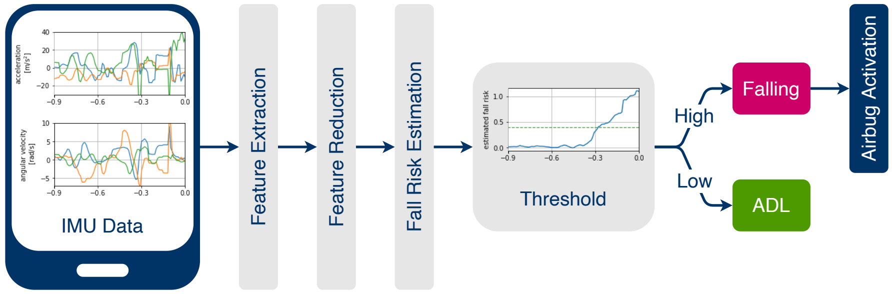


Fig. 1: The implementation flow of our pre-impact fall detection algorithm

- We collect simulated fall data including ambiguous falling activities from subjects wearing an inertial sensor on their waist.
- We have achieved improved robustness of fall detection against ambiguous falling activities.

The paper is structured as following. We describe the related work in the Section 2. We explain the detailed proposed method in the Section 3. We conduct the experiment to search for optimal hyperparameters and regression models in the Section 4. We evaluate the baseline and the proposed method in terms of accuracy and timing of fall detection in the Section 5. Finally, the Section 6 concludes the paper with discussing future directions of the research.

2 Related Work

Research on pre-impact fall detection has been conducted to prevent falls injuries in the elderly. Wu et al. in [17] has proposed that motion analysis techniques can be used to detect a fall before impact. Wu has showed that by thresholding the horizontal and vertical velocity profiles of the trunk, falls could be detected 0.3 to 0.4 s before impact. Wu et al. in [18] separated fall and ADL using a threshold detection method with the magnitude of the vertical velocity of the inertial frame as the main variable. The algorithm enable to detect all fall events at least 0.07 s before the impact. By setting the threshold for each individual subject, all falls can be detected and no false alarms occurred. Tamura et al. in [1] have developed a wearable airbag that incorporates a fall detection system. The fall detection algorithm is implemented using a thresholding technique based on accelerometers and gyroscopes. The algorithm is able to detect a signal 0.3 s before the impact. This signal triggers the airbag to inflate up to 2.4 liters, preventing physical impact that could lead to injury. These methods have achieved a high fall detection rate by using raw data of acceleration and angular velocity and a threshold. Their validation data certainly includes a various ADL events and fall events, but does not include data that is ambiguous about the distinction between falls and ADLs. In order to apply to real life, it is very important to validate with this ambiguous fall data and achieve a high fall detection rate.

In the field of HAR, the data-driven approach has been successfully used to improve classification accuracy and robustness. Matsuyama et al. in [19] classified dance figures based on acceleration, angular velocity, and posture extracted from images. Matsuyama have achieved high accuracy in classification by using Long short-term memory (LSTM) as a classifier. Wan et al. in [20] have proposed a accelerometer-based architecture for real-time HAR. He utilizes Convolutional Neural Network (CNN), LSTM, Bidirectional LSTM (BLSTM), Multilayer perceptron (MLP) and SVM models. Altun et al. in [21] have compared different techniques for HAR using small inertial and magnetic sensors attached to the body. Altun uses BDM, LSM, k-NN, DTW, SVM, and ANN as estimators and employs PCA and SFFS for feature reduction. These gave us the insight to run a general machine learning algorithm based HAR. Furthermore, Yoshida et al. in [22] and Chen et al. in [23] have successfully applied the HAR technique to regression analysis. Yoshida has developed a method for estimating walking speed from acceleration using deep neural network (DNN). Chen has proposed a DNN framework to estimate pedestrian speed and heading from acceleration and angular velocity. In fall detection, these suggests that the fall risk over time can be estimated from acceleration and angular velocity.

3 Proposed Method

We propose a data-driven pre-impact fall detection method. In this method, we train machine learning models with feature values and fall risk that represents the potential of fall, estimate current fall risk by regression and then detect pre-impact by threshold. In this section, to begin with, we explain what features to extract and how to reduct the number of them. Next, we define fall risk to use as the target value at machine learning. Then, we train several machine learning models with sets of the feature values and fall risk calculated according to the definition. Finally, we estimate current fall risk with the trained models and detect 0.3 seconds before the impact by using threshold.

3.1 Feature Extraction and Reduction

We extract 256 features from acceleration and angular velocity for each sliding window. We apply the idea of sliding window to the model for fall-potential estimation since potential to fall at a certain time should be estimated considering the sensor data in the last certain period of time. With this model, fall-potential at a time t is estimated by regressing the feature values of the window that is period from $t - W$

Table 1: 32 features extracted from each of the eight data (X-axis data, Y-axis data, Z-axis data, and norm data for acceleration and angular velocity, respectively)

classification	features extracted from each Window	data type	number of features
(1)	minimum value, maximum value, median value, mean value, standard deviation, total value, average of differences of neighboring elements, 0.05, 0.10, 0.25, 0.75, 0.90, 0.95 quantile, max/min value, max-min value	original value	15
(2)	skewness, kurtosis	original value	2
(1′)	same as (1)	absolute value	15

to t (t: time at right edge of the window, W: window size). The slide width when learning is half of the window size, and when estimation is set to 1.

As for features, 32 features are calculated from each of the eight data sets (X-axis data, Y-axis data, Z-axis data, and norm data for acceleration and angular velocity, respectively), resulting in 256 features for each single sliding window. 32 features are listed in Table 1. The 15 features in (1) are also computed in a window that took the absolute value of all the data, and made into a new features (1′). All features are normalized to the interval [0,1] to be used for machine learning in Sect. 3.3.

The 256 features obtained do not all have the same amount of information, and there is a potential that models cannot learn well due to their large feature dimension. Therefore, in order to obtain features that can calculate the fall risk more efficiently, we examine two methods: principal component analysis (PCA) for feature extraction and sequential feature forward selection (SFFS) for feature selection. PCA is a method for extracting more meaningful features by setting a new axis in the direction where the variance of the data is maximized. We use this method to reduce 256 features to 40 features with a cumulative contribution rate of about 90%. SFFS is different from PCA in that it selects features as they are. By sequentially selecting the best features, SFFS achieves feature reduction while maximizing the prediction capability of the model. The number of features to be selected is set to five because of the large computational cost.

3.2 Definition of Fall Risk

Some actions like standing or walking are classified with sufficient precision in conventional HAR research. In these general activity recognition task, the timing of recognition does not matter much. When it comes to real-time fall detection task, however, the timing of recognition needs to be carefully considered due to its time tightness. It is difficult to find the earliest time to be able to judge whether the subject is actually about to fall or not after he/she started falling. In fact, that the subject is falling does not get suddenly confirmed at some point. For example, in case a subject had stumbled on a step and eventually hit his/her hips, no one knows the exact time when that he/she was falling got confirmed. He/She might not have fallen if he/she had balanced.

Thus, we model the action of fall as its potential increases gradually, and estimate it by regression. It is expected that the index of fall potential is a value that meets following requirements:

- Takes 0 at ADL.
- Increases gradually toward end of fall.
- Takes 1 just before end of fall.

Therefore, we introduce a new index fr (fall risk) defined by following formula:

$$fr(t) = \begin{cases} 0 & (t < T_{fs}, t > T_{fe}) \\ \dfrac{t - T_{fs}}{W} & (T_{fs} \leq t \leq T_{fs} + W) \\ 1 & (T_{fs} + W < t \leq T_{fe}) \end{cases}$$

fr : fall risk
t : time at right edge of window
T_{fs} : fall start time
T_{fe} : fall end time
W : window size

Figure 2 shows the relationship between fr and t. We refer to period before T_{fs} as ADL phase and period from T_{fs} to T_{fe} as falling phase. fr is equivalent to the proportion of the area included in falling phase to the whole window. It takes 0 while the whole window is included in ADL phase. Then, it increases as the window moves and the proportion of the area included in falling phase increases. Finally, it takes 1 after the whole window gets included in falling phase.

3.3 Training of Machine Learning Model

We train machine learning models using the features obtained in Sect. 3.1 and fall risk calculated according to the definition in Sect. 3.2. T_{fe} for calculating fall risk are set by labels. In our approach, we use five regression techniques : Bayesian decision making (BDM), the least-squares method (LSM), the k-nearest neighbor algorithm (k-NN), support vector machines (SVM), and artificial neural networks (ANN). The accuracy of them is evaluated by mean square error (MSE). BDM is a method of determining the most likely weights by introducing noise that follows a normal distribution to the predictions. In this case, we also added a ridge regression term to prevent oveefitting. LSM is a well-known method used in linear regression analysis. It determines the plane so that the sum of the squared errors for each data and approximation plane is minimized. The k-NN and SVM are one of the most commonly used methods of regression analysis and classification. In ANN, we use a three-layer perceptron regressor. It is one of the simplest neural networks.

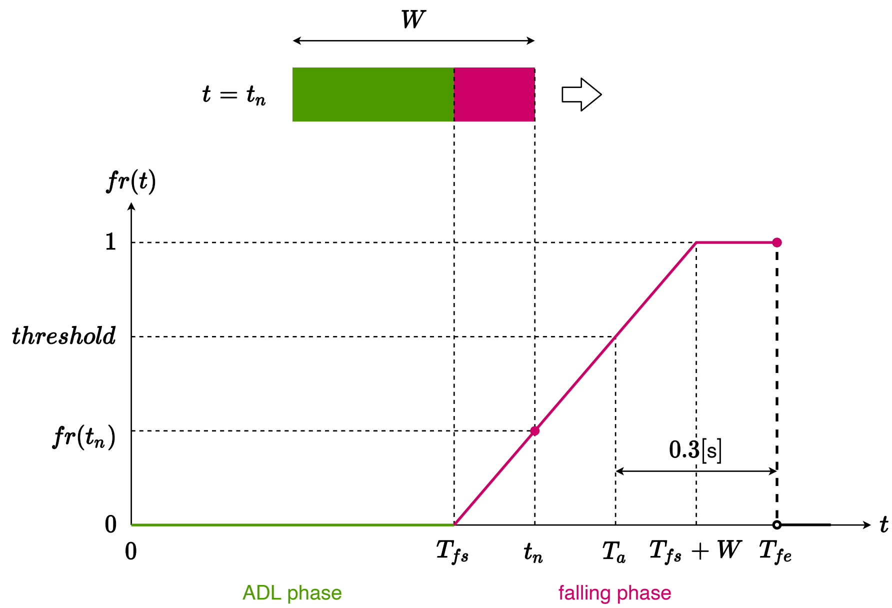


Fig. 2: Relationship between fall risk and window position

3.4 Estimation of Fall Risk and Fall Detection with Threshold

fr (fall risk) will be estimated by inputting the feature values of the current window to the model trained in Section 3.3. If estimated fr exceeds a certain threshold value, the system recognizes that the subject is falling, and outputs signal to activate the airbug which prevents him/her from getting injured. Threshold values to detect any seconds before end of fall can be given as the theoretical value of fr at that time. In this paper, we aim to detect 0.3 seconds before end of fall because it is generally considered that the model should be able to detect 0.3 to 0.4 seconds before the impact in order to activate the airbug in advance. As such, the threshold value corresponding to 0.3 seconds before the impact will be calculated by following equation:

$$threshold = fr(T_a)$$

$$T_a = T_{fe} - 0.3[\mathrm{s}] \ : \text{airbug activation time}$$

The relationship between the threshold value and T_a is shown in Figure 2. The pair of parameters (T_{fs}, W) needs to satisfy following condition so that the slope of $fr(t)$ does not take 0 around $t = T_a$:

$$T_a - T_{fs} \leq W \leq T_{fe} - T_{fs}$$

4 Search for Hyperparameters and Regression Models

We conduct experiments to find a machine learning model that can estimate fall risk with high accuracy. Initially, we collect data from four subjects to train and evaluate the machine learning model. Then we train and cross-validate the machine learning model using the collected data to select the optimal window size, feature reduction method, and regressor.

4.1 Experimental Setup

We collect training data for the proposed method from three subjects with an average age of 21, a weight of 58.0±7.0 kg, and a height of 172.0±9.4 cm and test data from one subject with an age of 24, a weight of 58.2 kg, and a height of 165.5 cm. Figure 3 shows a simulated falling experiment. Three of them fall to 11 times front, 12 times back, 11 times right-side and 5 times left-side and perform daily activities such as walking 39 times, a total of 78 times for training data. One of them fall to 4 times front, 4 times back, 4 times right-side and 4 times left-side and perform daily activities such as walking 16 times, a total of 32 for test data. They wear the smartphones with inertial sensors (XPERIA Z5 SO-01H) that is attached to the back of the waist and the right side of the waist. The y-axis of the smartphone is aligned with the subject's foot-to-head axis, and the back of the smartphone is in close contact with the body. In other words, the z-axis of the smartphone is perpendicular to the plane of the body and points from the inside of the body to the outside. At the moment of impact, the recorder presses a button on the M5stack to record the timing of the impact of the fall. The inertial sensor data and the impact timing data are sent to the same server in real time, thus that the time is synchronized accurately.

Our data simulates real-life falls, including stumbling, losing balance, and bending the knee in defense. It is more difficult to distinguish between fall events and ADL events in our data than in data from existing studies because there are many variations in the falling events in these data, including defensive responses. Therefore, our data set is more suitable to verify the robustness of the method against ambiguous activities.

4.2 The Result of Estimation Accuracy

Through a number of simulated fall experiments, we have found that setting T_{fs} to 0.5 s before impact is optimal for modeling fall risk well. Therefore, we search for W, reduction methods, and regressor that can estimate the fall risk with the highest accuracy when $T_{fs} = T_{fe} - 0.5$ s. Table 2 shows the results of the evaluation using mean squared error (MSE). As a result, when a window size of 0.5 s, a reduction

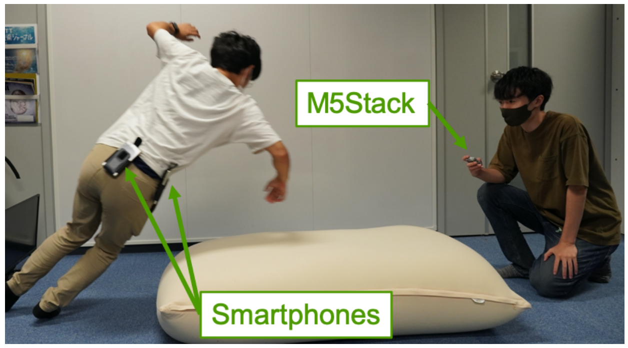


Fig. 3: A simulated falling experiment: The person wearing a smartphone on his waist falls and inertial sensor data is collected. The person holding the M5Stack presses a button at the timing of the impact of the fall and records it.

Table 2: The results of cross-validation with MSE when using each window size, feature reduction, and regressor (Target fall risk when $T_{fs} = T_{fe} - 0.5$)

W	BDM		LSM		k-NN		SVM		ANN	
	PCA	SFFS	PCA	SFFS	PCA	SFFS	PCA	SFFS	PCA	SFFS
0.20	0.0157	0.0158	0.0157	0.0183	0.0133	0.0229	0.0173	0.0230	0.0183	0.0148
0.30	0.0125	0.0127	0.0120	0.0151	0.0095	0.0158	0.0142	0.0196	0.0127	0.0114
0.40	0.0096	0.0100	0.0095	0.0121	0.0064	0.0104	0.0103	0.0159	0.0104	0.0082
0.50	0.0068	0.0072	0.0067	0.0084	**0.0044**	0.0083	0.0073	0.0104	0.0078	0.0072

method of PCA, and a regressor of k-NN, the MSE of fall risk is minimized. We use this setting as our method at evaluation in the Section 5.

5 Evaluation

We evaluate the baseline and the proposed method in terms of accuracy and timing of fall detection. Firstly, we introduce the baseline method. Secondly, we define the evaluation metrics. Finally, we show the evaluation results and discuss them.

5.1 Baseline Method

We evaluate the performance of pre-impact fall detection for the proposed method and the baseline method. We select [1] as baseline. It is threshold-based pre-impact fall detection algorithm using only current acceleration and angular velocity. It assumes that the acceleration signal during the fall is similar to free fall. Additionally, it suggests that pitch angular velocity signals can also be used for fall detection. Therefore, it judges a fall when the acceleration is less than ±3 m/s^2 and the pitch angular velocity exceeds 0.52 rad/s.

5.2 Evaluation Metrics

The results of running the fall detection algorithm on the test data can be categorized into the following four groups:

- True Positive (TP) that falls are detected in the falling phase.
- True Negative (TN) that falls are not detected in the ADL phase.
- False Positives (FP) that falls are detected in the ADL phase.
- False Negative (FN) that falls are not detected in the falling phase.

The high percentage of TP and TN indicates that the fall detection system works well to prevent falling injuries. If the percentage of FP is high that the airbag will malfunction during the ADL phase, causing inconvenience to the user. If the percentage of FN is high, the airbag will not work during the falling phase and the user will be injured by the fall. Therefore, it is ideal that FP and FN are zero. The time of airbag activation is also an important evaluation indicator to prevent fall injuries. Hence we calculate the mean and standard deviation of $T_{fe} - \hat{T}_a$. $\hat{T}_a$ denotes the estimated airbag activation time. This indicator is desirable to have a mean value of 0.3 and a standard deviation close to 0.

5.3 Evaluation results

Table 3 shows the number of TP, TN, FP, and FN counts at each sensor position when the baseline and the proposed method detection algorithm are executed on the test data. In common with both methods, there are almost no difference in accuracy by sensor position. The results of the baseline method shows that 3 for TP, 13 for TN, 3 for FP, and 13 for FN, giving an accuracy of 50.00%, a recall of 18.75% and a specificity of 81.25%. On the other hand, the results of the proposed method shows that 15 for TP, 16 for TN, 0 for FP, and 1 for FN, giving an accuracy of 96.88%, a recall of 93.75% and a specificity of 100.00%. It can be seen that the accuracy, recall and specificity are higher in the proposed method than in the baseline. The results show that the proposed method is more accurate in detecting falls and is able to make a distinction between ADL and falling more clearly than the baseline methods.

Table 4 shows the evaluation result of airbag activation timing. The overall $T_{fe} - \hat{T}_a$ of baseline has a mean of 0.265 s and a standard deviation of 0.150 s. The overall $T_{fe} - \hat{T}_a$ of the proposed method has a mean of 0.154 and a standard deviation of 0.0965. Considering that the target $T_{fe} - \hat{T}_a$ is 0.3 s, we can see that the baseline is more accurate. On the other hand, when we focus on the standard deviation, we can see that the proposed method is more stable with less variation.

Table 3: The number of TP, FP, and FN counts when the proposed method and the baseline fall detection algorithm are executed on the test data

	Sensor position	TP	TN	FP	FN
Baseline	Right	1	7	1	7
	Back	2	6	2	6
	Overall	3	13	3	13
Proposal	Right	8	8	0	0
	Back	7	8	0	1
	Overall	15	16	0	1

Table 4: The mean and standard deviation of $T_{fe} - \hat{T}_a$ s that is time difference between falling impact and the airbag activation

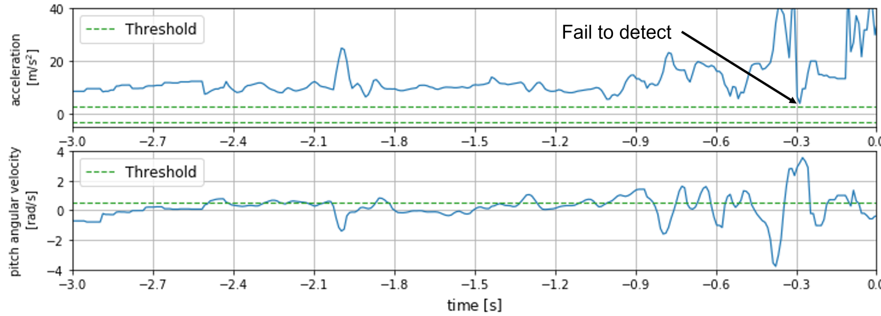
	Sensor position	$T_{fe} - \hat{T}_a$ Mean	Std
Baseline	Right	0.334	—
	Back	0.231	0.174
	Overall	0.265	0.150
Proposal	Right	0.168	0.097
	Back	0.137	0.094
	Overall	0.154	0.097

5.4 Comparison of our Proposed with the Baseline

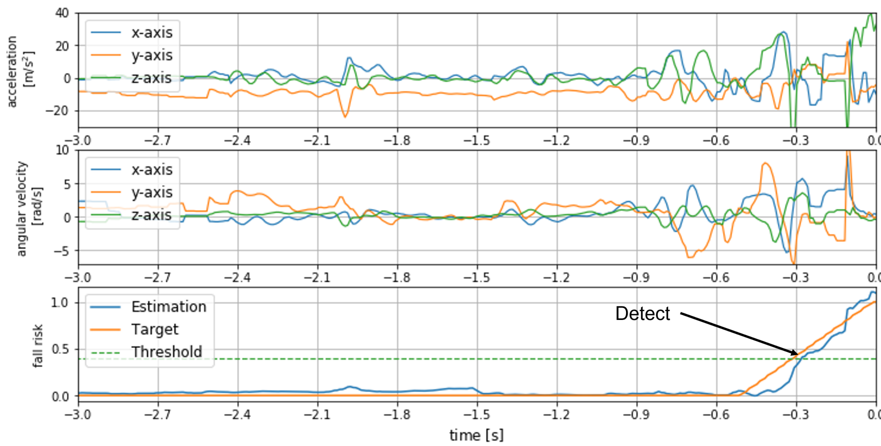
Table 3 shows that there is a large difference in the recall, which indicates the fall detection rate, between the proposed method (93.75%) and the baseline (18.75%). Figure 4 shows a typical example of fall detection for baseline and the proposed method when the subject has an inertial sensor attached to the right side of the waist and falls to the right side. Figure 4a shows a example of baseline. The signal is not reaching within 0.3 m/s^2 that is the threshold for acceleration. All of the test data simulated realistic falls, and included behaviors such as stumbling, losing balance, and defensive reactions. There are few such data that can detect a clear free fall. As a result, the threshold-based method fails to detect it. In addition, there are some data that cannot detect a fall because the angular velocity in the free fall phase is small due to the defensive reaction of bending the knee and falling from the knee that is also mentioned in the discussion of [1]. This experiment has shown that it is difficult to implement a threshold-based system that works reliably in real-world fall accidents and does not cause the airbag to malfunction during the ADL phase.

Figure 4b shows the results of pre-impact fall detection based on proposed method that uses a threshold and fall risk estimation with a window size of 0.5 s, a reduction method of PCA, and a regressor of k-NN. k-NN have successfully estimated the fall risk we defined in the Section 3 from the acceleration and angular velocity. This result supports the validity of our assumed fall risk model.

We can see that we have succeeded in detecting the fall 0.3 s before the impact from the time of the crossing point between a threshold of 0.4 and the fall risk. This result shows that improving the accuracy of fall risk estimation contributes to improving the accuracy and robustness of pre-impact fall detection and reducing the



(a) Baseline method



(b) Proposed method

Fig. 4: The example of fall detection for baseline and proposed methods

number of false fall detections. In other words, we only need to focus on improving the accuracy of fall risk to improve the performance of pre-impact fall detection.

5.5 The Importance of Features

We use the SFFS results to discuss the features that are important in estimating the fall risk. We have found three features that are important for estimating fall risk in k-NN in the process of feature extraction by SFFS. The features are the 0.05 quantile of y-axis acceleration (feat-1) and the kurtosis of y-axis acceleration (feat-2) and average of the differences of neighboring elements in absolute value of z-axis acceleration (feat-3). Figure 5 shows an example of these features and the target fall

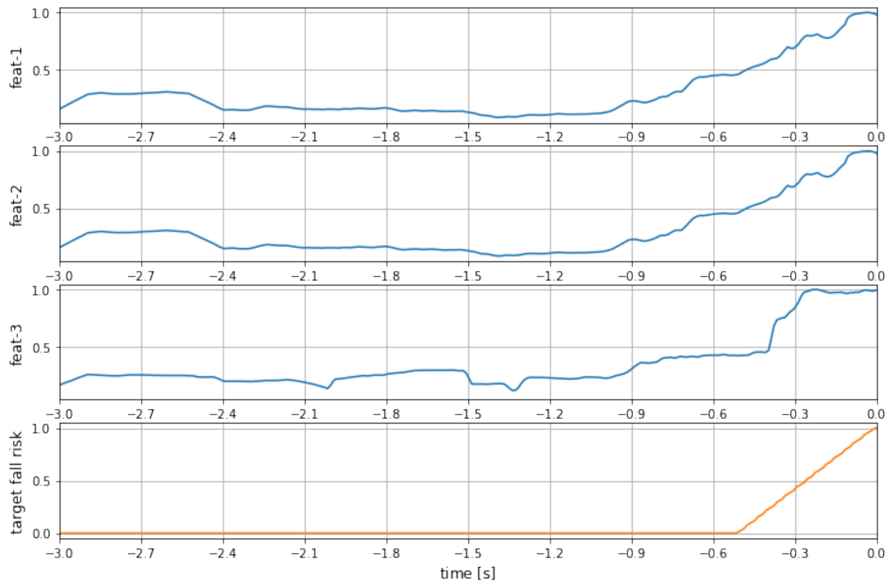


Fig. 5: The example of three features that are important for estimating fall risk in k-NN that is selected by SFFS

risk. Each features are shown as feat-1, feat-2, and feat-3 in order. It can be seen from the Figure 5 that there is a strong correlation between target fall risk and these features. The values of feat-1 and feat-2 rise faster than the target fall risk, however the values from -0.9 to -0.6 s are close to the values around -2.7 s, therefore it is difficult to use them as a basis for activating the fall risk. However, after -0.6 s, the value of feat-1,2 exceeds 0.5, and after -0.4, the value of feat-3 increases significantly that determines the increase in falling risk.

5.6 Evaluation of Airbag activation time

The $T_{fe} - \hat{T}_a$ of the proposed method, 0.15, is smaller than the target of 0.3. This means that the time to activate the airbag is shortened that needs to be improved. The reason is the delay in the rise of the fall risk estimated by the proposed method. Figure 6 shows an example of the delay in the rise of the fall risk. To solve this problem, we can increase the number of training data or improve the learning model. These are future issues.

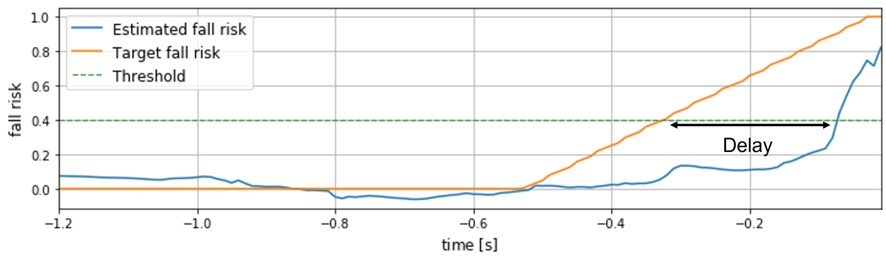


Fig. 6: The example of the delay in the rise of the fall risk estimated by our proposed method

6 Conclusion

We have developed a online pre-impact fall detection algorithm for airbags to prevent fall injuries. Our algorithm uses a data-driven approach to calculate the risk of falling based on acceleration and angular velocity features, and uses a thresholding technique to robustly detect a fall before impact. As a result of evaluating the algorithm on data simulating real-life falls, we succeeded in significantly improving the accuracy of fall detection from 50.00% to 96.88% and the recall from 18.75% to 93.75% over the baseline method. However, the fall detection time of the proposed method is on average 0.15 s before the impact that is shorter than 0.3 s that is enough time to activate the airbag. As a solution, we consider a method to increase the training data to improve the accuracy of fall risk estimation and reduce the delay of airbag activation. In addition, the current evaluation dataset does not have enough variation and quantity in terms of age, gender, etc., therefore these need to be increased.

Acknowledgements This work is supported by JSPS KAKENHI Grant Number JP17H01762, JST CREST and NICT.

References

1. Toshiyo Tamura, Takumi Yoshimura, Masaki Sekine, Mitsuo Uchida, and Osamu Tanaka. A wearable airbag to prevent fall injuries. *IEEE Transactions on Information Technology in Biomedicine*, 13(6):910–914, 2009.
2. Graham Mulley. Falls in older people. *Journal of the Royal Society of Medicine*, 94(4):202–202, Apr 2001. PMC1281399[pmcid].
3. Muhammad Mubashir, Ling Shao, and Luke Seed. A survey on fall detection: Principles and approaches. *Neurocomputing*, 100:144–152, 2013. Special issue: Behaviours in video.
4. Xiaodan Zhuang, Jing Huang, Gerasimos Potamianos, and Mark Hasegawa-Johnson. Acoustic fall detection using gaussian mixture models and gmm supervectors. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 69–72, 2009.

5. M. Alwan, P.J. Rajendran, S. Kell, D. Mack, S. Dalal, M. Wolfe, and R. Felder. A smart and passive floor-vibration based fall detector for elderly. In *2006 2nd International Conference on Information Communication Technologies*, volume 1, pages 1003–1007, 2006.
6. H. Nait-Charif and S.J. McKenna. Activity summarisation and fall detection in a supportive home environment. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 4, pages 323–326 Vol.4, 2004.
7. Tracy Lee and Alex Mihailidis. An intelligent emergency response system: Preliminary development and testing of automated fall detection. *Journal of telemedicine and telecare*, 11:194–8, 02 2005.
8. J.Y. Hwang, J.M. Kang, Y.W. Jang, and H.C. Kim. Development of novel algorithm and real-time monitoring ambulatory system using bluetooth module for fall detection in the elderly. In *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 1, pages 2204–2207, 2004.
9. A. Diaz, M. Prado, L.M. Roa, J. Reina-Tosina, and G. Sanchez. Preliminary evaluation of a full-time falling monitor for the elderly. In *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 1, pages 2180–2183, 2004.
10. Alan Bourke, J.V. O'Brien, and Gearóid ÓLaighin. Evaluation of threshold-based tri-axial accelerometer fall detection algorithm. *Gait posture*, 26:194–9, 08 2007.
11. A.K. Bourke and G.M. Lyons. A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. *Medical Engineering Physics*, 30(1):84–90, 2008.
12. Fabio Bagalà, Clemens Becker, Angelo Cappello, Lorenzo Chiari, Kamiar Aminian, Jeffrey M. Hausdorff, Wiebren Zijlstra, and Jochen Klenk. Evaluation of accelerometer-based fall detection algorithms on real-world falls. *PLOS ONE*, 7:1–9, 05 2012.
13. Mohammad Ashfak Habib, Mas S. Mohktar, Shahrul Bahyah Kamaruzzaman, Kheng Seang Lim, Tan Maw Pin, and Fatimah Ibrahim. Smartphone-based solutions for fall detection and prevention: Challenges and open issues. *Sensors*, 14(4):7181–7208, 2014.
14. Jin Wang, Zhongqi Zhang, Bin Li, Sungyoung Lee, and R. Simon Sherratt. An enhanced fall detection system for elderly person monitoring using consumer home networks. *IEEE Transactions on Consumer Electronics*, 60(1):23–29, 2014.
15. Bijan Najafi, Kamiar Aminian, François Loew, Yves Blanc, and Robert Prince. Measurement of stand-sit and sit-stand transitions using a miniature gyroscope and its application in fall risk evaluation in the elderly. *IEEE transactions on bio-medical engineering*, 49:843–51, 08 2002.
16. Tong Zhang, Jue Wang, Liang Xu, and Ping Liu. Using wearable sensor and nmf algorithm to realize ambulatory fall detection. In Licheng Jiao, Lipo Wang, Xinbo Gao, Jing Liu, and Feng Wu, editors, *Advances in Natural Computation*, pages 488–491, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
17. G Wu. Distinguishing fall activities from normal activities by velocity characteristics. *Journal of biomechanics*, 33(11):1497—1500, November 2000.
18. Ge Wu and Shuwan Xue. Portable preimpact fall detector with inertial sensors. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 16(2):178–183, 2008.
19. Hitoshi Matsuyama, Shunsuke Aoki, Takuro Yonezawa, Kei Hiroi, Katsuhiko Kaji, and Nobuo Kawaguchi. Deep learning for ballroom dance: A performance classification model with three-dimensional body joints and wearable sensor. *IEEE Sensors Journal*, pages 1–1, 2021.
20. Shaohua Wan, Lianyong Qi, Xiaolong Xu, Chao Tong, and Zonghua Gu. Deep learning models for real-time human activity recognition with smartphones. *Mobile Networks and Applications*, 25(2):743–755, Apr 2020.
21. Kerem Altun and Billur Barshan. Human activity recognition using inertial/magnetic sensor units. volume 6219, pages 38–51, 08 2010.
22. Takuto Yoshida, Junto Nozaki, Kenta Urano, Kei Hiroi, Takuro Yonezawa, and Nobuo Kawaguchi. Gait dependency of smartphone walking speed estimation using deep learning. pages 641–642, 06 2019.
23. Changhao Chen, Xiaoxuan Lu, Andrew Markham, and Niki Trigoni. Ionet: Learning to cure the curse of drift in inertial odometry. 01 2018.