MIPRO 2018, May 21-25, 2018, Opatija Croatia

Optimal threshold selection for threshold-based fall detection algorithms with multiple features

D. Razum, G. Seketa, J. Vugrin and I. Lackovic
University of Zagreb, Faculty of Electrical Engineering and Computing, Zagreb, Croatia

***Abstract* - As people get older, their bodies go through multiple changes that make them more fragile and susceptible to falls. The population of elderly people living alone is increasing worldwide, and this imposes a risk that a potential fall may happen without receiving prompt attention of a healthcare provider or caregiver. To solve this problem, various solutions for automatic fall detection have been proposed that recognise when a person falls and send alarms to someone that could provide quick help. One group of automatic fall detectors use wearable sensors attached to a person's body to measure body accelerations and then to distinguish falls from normal activities of daily living (ADLs) with some of the threshold or machine learning based algorithms. In threshold-based algorithms, features are calculated from measured accelerations and they are evaluated with a set of rules to check whether a fall has happened. The choice of fixed thresholds is thereby important for the overall efficiency of the algorithm. In our previous works in the field of fall detection, we have analysed methods for the determination of appropriate threshold levels for algorithms based on one acceleration-based feature. In this paper, we present a method for setting optimal thresholds in algorithms that use multiple acceleration-derived features. We demonstrate the efficiency of algorithms with thresholds set according to the newly presented method when tested on our dataset of accelerations measured during simulated falls and ADLs.**

Keywords - fall detection, acceleration, threshold, Receiver Operating Characteristics

I. INTRODUCTION

The percentage of elderly people (aged over 65 years) in the overall world population steadily grows and is expected to reach 28% by 2050 [1]. A lot of effort is put today in the research of means to improve the life quality of that population group. Of particular interest are a better healthcare service and solutions that could assist in the independent living of elderly people.

Unintentional falls are one of the major problems of the elderly people. The risk of experiencing an unintentional fall increases with age and according to the statistics, 28-35% of people over 65 years, experience a fall every year [2]. These falls have both negative physical and psychological consequences, leading to a decrease of a person's independence due to injuries or a fear of falling again. After a person experiences a fall, the negative consequences can be reduced by a prompt reaction of a caregiver or healthcare provider, and for this purpose various systems for automatic fall detection have been researched and developed [1]. The main goal of these systems is to automatically detect when a person falls without the need of any additional intervention and to alert caregivers and/or healthcare providers.

A broad range of technologies has so far been proposed for the automatic fall detection purposes [3, 4, 5]. Generally, systems for fall detection can be divided into three group depending on the technologies they are using: ambient based, vision-based and wearable. In vision-based systems, a video camera is used to record a person's movements and image processing algorithms are used to classify falls from regular daily activities. Ambient systems rely on measurements of various sensors set in a person's environment to analyze whether a fall has happened. In wearable systems, a person carries a device attached to his/her body and data is gathered from sensors embedded on the device. Algorithms are then used to detect possible falls from sensor measurements.

In wearable systems for fall detection, mostly accelerometric sensors are used, although the use of other sensors like gyroscopes, magnetometers and barometers have also been explored in the literature [6, 7, 8]. This study focuses only on wearable fall detection systems that utilize acceleration sensors.

In wearable, acceleration based fall detection systems, sensor measurements are fed to an algorithm that detects fall events. The aim of fall detection algorithms is to correctly detect falls and not to generate false alarms during activities of daily living (ADL). The algorithms proposed in the literature can mainly be divided into two categories: threshold-based algorithms and machine learning algorithms [9]. In threshold-based algorithms, features are calculated from acceleration measurements and they are compared to threshold values to determine if a fall has happened.

The choice of threshold values is essential for the algorithm's efficacy. The influence of various methods used to select algorithm's thresholds have not yet been thoroughly explored in the literature. In our previous work [10] we proposed a method for the determination of thresholds for various single features used for fall detection. In this work, we analyze the threshold selection process for two algorithms that use two features for fall detection. For this purpose, two threshold selection methods were implemented and their performance was tested with a dataset of acceleration measurements assessed with a wearable device during ADLs and simulated falls. The performance of the algorithms is mutually compared with thresholds selected using two different methods.

This paper is organized into five main sections. Section II presents a description of algorithms and methods used for choosing optimal threshold values. Section III describes results of the evaluation of proposed algorithms. Discussion and conclusion are given in sections IV and V respectively.

II. Materials And Methods

A. Dataset

For the algorithm performance evaluation, a dataset containing acceleration measurements obtained with a wearable sensor unit was used. The dataset was acquired from 16 young subjects (15 to 44 years) that wore the sensor unit attached to their waist while performing ADLs and simulated falls in a safe laboratory environment. A commercially available sensor unit Shimmer3 (Shimmer Sensing, Dublin, Ireland) was used in the experiments, with the acceleration sensor set to ±8g range and 204.8 Hz sampling rate. In total, subjects performed 15 tasks: 12 ADL and 3 falls, as described in our previous paper [10].

B. Features

Each file in the dataset consisted of 3-axial acceleration measurements of a single task performed by one subject in one experiment trial. From acceleration measurements, two features were calculated: :

-Sum Vector Magnitude (*SVM*)

$$SVM(n) = \sqrt{a_x(n)^2 + a_y(n)^2 + a_z(n)^2} \quad (1)$$

-Euler angle (*THETA*) between the direction of Earth gravitational field and the device's vertical axis (y):

$$THETA\ (n) = \text{atan2}\left(\frac{\sqrt{a_x(n)^2+a_z(n)^2}}{a_y(n)}\right) \quad (2)$$

where $a_x(n)$, $a_y(n)$ and $a_z(n)$ represent the *n*-th sample of acceleration in the corresponding sensor axes. Atan2 is an inverse tangent function that returns values in the [-π, π] interval.

C. Algorithms

Two simple threshold-based algorithms, based on the ideas from [11], were implemented: THETA&SVM and SVM&THETA. The first algorithm (THETA&SVM) consists of observing whether the value of the angle feature *THETA* surpasses the threshold value assigned to the *THETA* feature (*theta_threshold*) within a time interval of 0.5 seconds. If the THETA value exceeds the *theta_threshold*, then the *SVM* values in the same time interval window are compared to the threshold set for the *SVM* feature (*svm_threshold*). Finally, if a *SVM* value is found that is greater than the *svm_threshold*, a fall is detected. The reasoning behind this approach is that a person would first experience the angle change when a fall starts, detected with a THETA threshold and followed by an impact that would cause a peak in the SVM feature.

The second algorithm (SVM&THETA) also relies on comparing feature values with their corresponding thresholds. First the *SVM* values for every sample are compared with the *svm_threshold*. If the *svm_threshold* value is surpassed, then the THETA feature is analysed in the following samples. The idea behind this approach is to first detect the impact with the ground and after that the lying orientation of the falling person. If an THETA angle greater than *theta_threshold* is detected, an additional condition is checked to reduce the number of false positives. In a time frame of 1s around the sample at which the THETA angle is found to be greater than the threshold, the mean value of SMV is calculated. If this value is less than 1.2, indicating that a person is at rest, a fall is detected.

D. Performance Evaluation

The algorithms were tested with measurements contained in our dataset to evaluate their performance. Confusion matrices (error matrix, contingency table [12]) were generated by varying the threshold values and testing the algorithms' success in detecting and distinguishing fall events from ADL. Thresholds were chosen from a limited set of values containing: for THETA feature, all values from 0° to 175° in increments of 5°; for SVM feature, all values from 0.1g to 12g in increments of 0.1g. For every combination of thresholds the appropriate numbers in the confusion matrix were updated:

- the number of true positive events (TP) – fall happened, algorithm detected fall
- the number of false negative events (FP) – no fall happened, algorithm detected a fall
- the number of true negative events (TN) – no fall happened, algorithms did not detect a fall
- false negative events (FN) – fall happened, algorithm did not detect a fall

The results of the algorithms' evaluation contained in the confusion matrix were used to calculate three performance metrics as follows:

$$sensitivity = \frac{TP}{TP+FN} \quad (3)$$

$$specificity = \frac{TN}{TN+FP} \quad (4)$$

$$meanSESP = \sqrt{SE * SP} \quad (5)$$

E. Optimal Threshold Selection

The optimal threshold values for the algorithms were selected from the Receiver Operating Characteristics (ROC) curve. The ROC curve is obtained by plotting *sensitivity* (true positive rate) versus 1-*specificity* (false positive rate) values. Optimal thresholds were selected from the ROC curve as values that minimize the distance between the curve and then point (0,1) which represents the ideal case of 100% sensitivity and specificity rates. Two methods of choosing the two threshold values for the algorithms were analyzed.

In the first method (MET1), optimal thresholds for each feature used in the algorithms were selected separately and set to values determined in our previous work (optimal thresholds determined for *theta* and *SVM*) [10]. With the *svm_threshold* and *theta_threshold* chosen in this way, the performances of both algorithms were evaluated and the performance metrics were calculated.

For the second method (MET2), confusion matrix was generated for all combinations of the two threshold values. Thus, a single ROC curve was plotted for each of the algorithms and by the optimal threshold selection

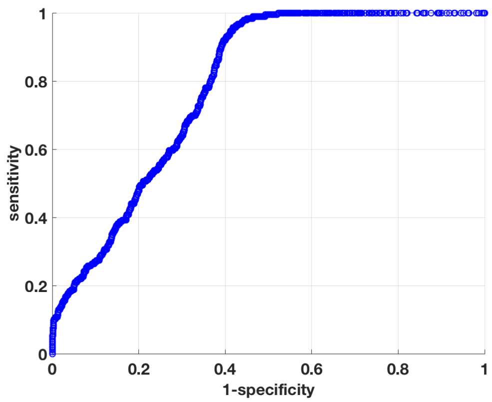

Figure 1. ROC curve of the THETA&SVM algorithm obtained by analyzing all threshold pairs

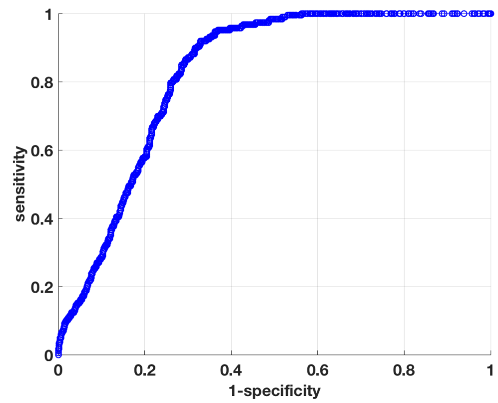

Figure 2. ROC curve of the SVM&THETA algorithm obtained by analyzing all threshold pairs

procedure described above, a pair of optimal threshold values (*svm_threshold, theta_threshold*) was sought.

III. Results

The two implemented algorithms, THETA&SVM and SVM&THETA, were evaluated using our dataset and corresponding confusion matrices were generated. From the confusion matrices, performance metrics were calculated and ROC curves were plotted. ROC curves obtained for both algorithms by analyzing all threshold value pairs (MET2) are given in Fig. 1 and Fig. 2.

Optimal threshold values for both algorithms were chosen by the two methods described in the previous section. Values of the selected thresholds along with the performance metrics calculated for them are given in TABLE I.

IV. Discussion

Among the analyzed algorithms and threshold selection methods in this study, the algorithm SVM&THETA with thresholds set according to MET2 showed the best results with the sensitivity of 86.0%, specificity 80.1% and *meanSESP* 83%.

In our previous work [10], we calculated the performance metrics (sensitivity, specificity) for a set of features used individually for fall detection. From the analyzed features, the best results were achieved with the *GSVM* feature. *GSVM* is calculated as a normalized product of *THETA* and *SVM* for a given sample. With the threshold set at 3g, the GSVM feature was tested with the same dataset as used in this study and sensitivity of 90.3% and specificity of 88.5% were achieved (*meanSESP* 89.4%). Comparing these results to the ones obtained with the algorithms tested in this study, in all cases fall detection based on the GSVM feature outperforms the tested multi-feature algorithms.

The THETA and SVM feature were also among the features analysed individually in [10]. For the THETA feature, sensitivity of 94% and specificity of 63.5% were achieved (*meanSESP* 77.2%) and for the SVM feature sensitivity of 88.7% and specificity of 67.8% were achieved (*meanSESP* 77.5%). Thus, the algorithm SVM&THETA with thresholds set with MET2, outperformed the single features of THETA and SVM in fall detection tasks. On the contrary, the THETA&SVM algorithm performed similarly or even slightly worse than THETA and SVM individually. This showed once more the potential benefits of combining multiple features in the algorithm and the importance of the methods used to combine the features.

Since both algorithms use some fixed time intervals or thresholds (time window of 0.5 s in THETA&SVM, time window of 1 s and threshold for mean SVM in SVM&THETA), the performance of the algorithms should also be tested with various values for these variables. In our

TABLE I. PERFORMANCE OF THE ANALYZED ALGORITHMS

Algorithm	Method	Optimal threshold	*sensitivity*	*specificity*	*meanSESP*
THETA&SVM	MET1	*svm_threshold*=4,3g *theta_threshold*=120°	61.3%	69.3%	65.2%
	MET2	*svm_threshold*=4,4g *theta_threshold*=45°	86.6%	65.6%	75.4%
SVM&THETA	MET1	*svm_threshold*=4,3g *theta_threshold*=120°	20.4%	89.9%	42.8%
	MET2	*svm_threshold*=4,1g *theta_threshold*=70°	86.0%	80.1%	83.0%

future work, we plan to set these variables to optimal values and assess the algorithms' performances to see whether in this way their efficacy could be improved.

V. Conclusion

In both tested algorithms, the optimal thresholds and thus the performance metrics changed with the change of methods used to determine the thresholds. The algorithms performed better with the thresholds set combined (MET2) than with the thresholds set separately (MET1). The proposed method (MET2) for optimal threshold selection in multiple feature algorithms may therefore be beneficial for the overall algorithm performance compared to the standard method of choosing the threshold individually (MET1), however further experiments must be carried out to make stronger conclusion.

In conclusion, we implemented two algorithms for fall detection based on the analysis of two acceleration derived features. We analyzed their performance with the use of our dataset for fall detection comprised of acceleration measurements. For each algorithm, we examined the effects of using two different methods for the selection of optimal threshold levels. The best results were obtained for the SVM&THETA algorithm with thresholds set according to the method MET2: *sensitivity* of 86.0%, *specificity* 80.1% and *meanSESP* 83%.

References

[1] Lapierre N, Neubauer N, Miguel-Cruz A et al. The state of knowledge on technologies and their use for fall detection: A scoping review. International Journal of Medical Informatics. Vol. 111, pp 58-71, 2018

[2] Casilari E, Santoyo-Ramon J, Cano-Garcia J. Analysis of public datasets for wearable fall detection systems. Sensors (Switzerland). Vol 17, issue 7, 2017.

[3] Madhubala S, Umamakeswari A, Jenita A. A survey on technical approaches in fall detection systems. National Journal of Physiology, Pharmacy and Pharmacology. Vol. 5, Issue 4, 2015.

[4] King R, Villeneuve E, White R et al. Application of data fusion techniques and technologies for wearable health monitoring. Medical Engineering and Physics. Vol. 42, pp 1-12, 2017.

[5] Koshmak G, Loutfi A, Linden M. Challenges and issues in multisensory fusion approach for fall detection: Review paper. Journal of Sensors. Vol 2016, 2016.

[6] Khowaja S, Setiawan F, Prabono A et al. An effective threshold based measurement technique for fall detection using smart devices. International Journal of industrial Engineering: Theory Applications and Practice. Vol. 23, Issue 5, pp 332-348, 2016.

[7] Sabatini A, Ligorio G, Mannini A et al. Prior-to and post-impact fall detection using inertial and barometric altimeter measurements. IEEE Transactions on Neural Systems and Rehabilitation Engineering. Vol 24, Issue 7, pp 774-783, 2016.

[8] Pierleoni P, Belli A, Maurizi L et al. A Wearable Fall Detector for Elderly People Based on AHRS and Barometric Sensor. IEEE Sensors Journal. Vol 16, Issue 17, 2016.

[9] Pannurat N, Thiemjarus S, Nantajeewarawat E. Automatic fall monitoring: A review. Sensors (Switzerland). Vol 14, Issue 7, 2014.

[10] Seketa, G., Vugrin, J., Lackovic, I. Optimal threshold selection for acceleration-based fall detection (2018). 3rd International Conference on Biomedical and Health Informatics, Thessaloniki; Greece. IFMBE Proceedings, 66, pp. 151-155.

[11] Garret B. An Accelerometer Based Fall Detector: Development, Experimentation, and Analysis. Summer Undergraduate Program in Engineering at Berkeley (SUPERB) 2005.

[12] Fawcett T. An Introduction to ROC Analysis. Pattern Recognition Letters. 27 (8): 861-874.