# Assistive Smart Cane (ASCane) for Fall Detection: First Advances

Pedro Mouta[1], Nuno Ferrete Ribeiro[1], Rui Moreira[1], and Cristina P. Santos[1]

[1]Center for MicroEletroMechanical Systems (CMEMS), University of Minho, 4800-058, Guimarães, Portugal,
pmfmouta@gmail.com,nuno.ferrete.ribeiro@gmail.com,
a71335@alunos.uminho.pt and cristina@dei.uminho.pt

**Abstract.** The development of fall detection systems with the capability of real-time monitoring is necessary considering that a large amount of people die and suffer severe consequences from falls. Due to their advantages, daily life accessories can be a solution to embed fall-related systems, and canes are no exception. In this paper, it is presented a cane with fall detection abilities. The ASCane is instrumented with an inertial sensor which data will be tested with three different fixed multi-threshold fall detection algorithms, one dynamic multi-threshold and machine learning methods from the literature. They were tested and modified to account the use of a cane. The best performance resulted in a sensitivity and specificity of 96.90% and 98.98%, respectively.

**Keywords:** Fall Detection, Machine Learning, IMU, Activities of Daily Living

## 1 Introduction

Falls are the second main reason of death by accident worldwide [1]. The estimated medical costs in the United States of America (U.S.A.) concerning fatal and nonfatal falls, in 2015, were approximately $32 billion, and the annual average cost of treating individuals due to injuries from a fall is approximately $20.000 [2, 3]. By 2020, expenses linked to injuries from falls to senior citizens are expected to cost $43.8 billion [4].

Researchers have proposed several different solutions regarding fall-related systems. Most of the developed projects focus on fall detection and employ methods supported by vision, wearable and environmental approaches. Commonly, different sensors are attached to the subject's body. However, the wearable system weights on the individual and hinder its flexibility. On the contrary, both optical and environmental device approaches free the subject of sensors, but require a pre-built infrastructure restricting the subject movements [5].

Populations with motor impairment normally use assistive devices that auxiliary their gait. Nowadays, more than 4 million people in the U.S.A. use a cane and its usage will increase with the elderly population growth. The size is another advantage to choose canes as fall-related systems [6–8].

Bourke et al. [9], developed a threshold based algorithm using exclusively accelerometry data from the trunk or thighs for the computation of the acceleration Sum Vector Magnitude. If the upper or the lower individual threshold is surpassed, a fall is detected with an accuracy of 100%.

Bourke et al. [10], also identified three gyroscopic features from the trunk, namely the Sum Vector Magnitude ($\omega_{res}$), the resultant angular acceleration ($\theta_{res}$) and the resultant change in trunk-angle ($\alpha_{res}$). When their thresholds are exceeded, a fall is also detected with an accuracy of 100%.

The algorithm introduced by Kangas et al. [11] is based on the analysis of 5 acceleration parameters from the wrist, head or waist. With the combination of the Sum Vector Magnitude, the Dynamic Sum Vector, $SV_{TOT}$, the Vertical acceleration, $Z_2$, the $SV_{maxmin}$ and the final posture, which is detected 2 seconds after the impact, a sensitivity of 97.5% and a specificity of 100% was obtained for the waist.

Fixed threshold-based algorithms can be insufficient to achieve the main goal of fall-related systems due to inter and intra-variability of subjects, and limited sample size [9–12]. Thus, these methods should be adaptive and account for variability. Otanasap et al. [13] developed a dynamic threshold algorithm by the means of accelerometry data. A Fixed Threshold (FT) is computed based in the data acquired from the subject while performing Activities of Daily Living (ADL), $\text{ADL}_{acc}$. Secondly, the Dynamic Threshold (DT) is formulated by the FT added by a standard deviation calculated with the data gathered in the last second. The algorithm outputs a percentage which discriminates the possibility of a fall, reaching results of 97.4%, 99.5% and 95.3% for accuracy, sensitivity and specificity, respectively.

Xu et al. [5] reviewed and compared fall detection algorithms on the most cited works. It was found an increase of machine learning algorithms in recent years, namely the Support Vector Machine(SVM) and the Decision Tree. The accuracy of the algorithms are relatively high since most accuracies are above 90%, ranging between 79.6% and 100%.

The main objective of this work is to develop a cane able to detect falls. This manuscript focus on applying the described algorithms over the data collected on a cane being used as an assistive device for eleven healthy users. The algorithms are modified to account for the use of a cane. The remainder of this paper is organized as follows. Section II provides a complete system overview which includes the hardware installed into the cane, its purpose, the different fall detection algorithms considered and the experimental protocol for data acquisition. In section III, the results attained with the different fall detection algorithms are presented. Finally, in section IV, a discussion of the achieved results is accomplished and the paper is concluded in section V.

## 2 Methods and Materials

### 2.1 System Overview

The ASCane, Fig. 1., uses an inertial sensor located on the upper part, an 9-axis Motion Processing Unit (MPU-9250), which is widely embedded in systems related to human motion [3, 5, 12, 14, 15]. Chen et al. [16] studied acceleration readings in different places of a cane under the same falling process and concluded that the amplitudes of the acquired data in the upper part of the device were higher than the other locations. The higher the variation, the easier is to observe discriminative characteristics of the signal.

As full scale range, it was used $\pm 8g$ for the accelerometer and $\pm 300^o$/s for the gyroscope [15, 17]. When the cane is being used, all its sensors are operating at 200Hz, which is higher then the sampling rate in other cane related works [3, 5, 16]. The acquired data was saved in a memory card and the microcontroller STM32F303K8 was used.

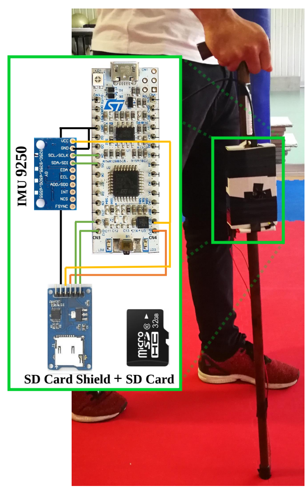


**Fig. 1.** Set-up of the ASCane and its location in a healthy user.

### 2.2 Fall Detection Algorithms

The complete fixed threshold algorithms are exposed in Fig. 2 and the original threshold values for each feature, body location and study are summarized in Table 1. The algorithm introduced by Otanasap et al. [13], which will also be tested and adapted to the ASCane, is presented in Fig.3.

A search was conducted in order to uncover which machine learning algorithms authors use to classify falls. As result, 4 articles were selected [18–21], and their corresponding features are presented in Table 2. Data were then divided in two different classes: Fall and ADL samples. Afterwards, 70% of each data were used to train the classifier and 30% to test it as indicated in Fig.4.

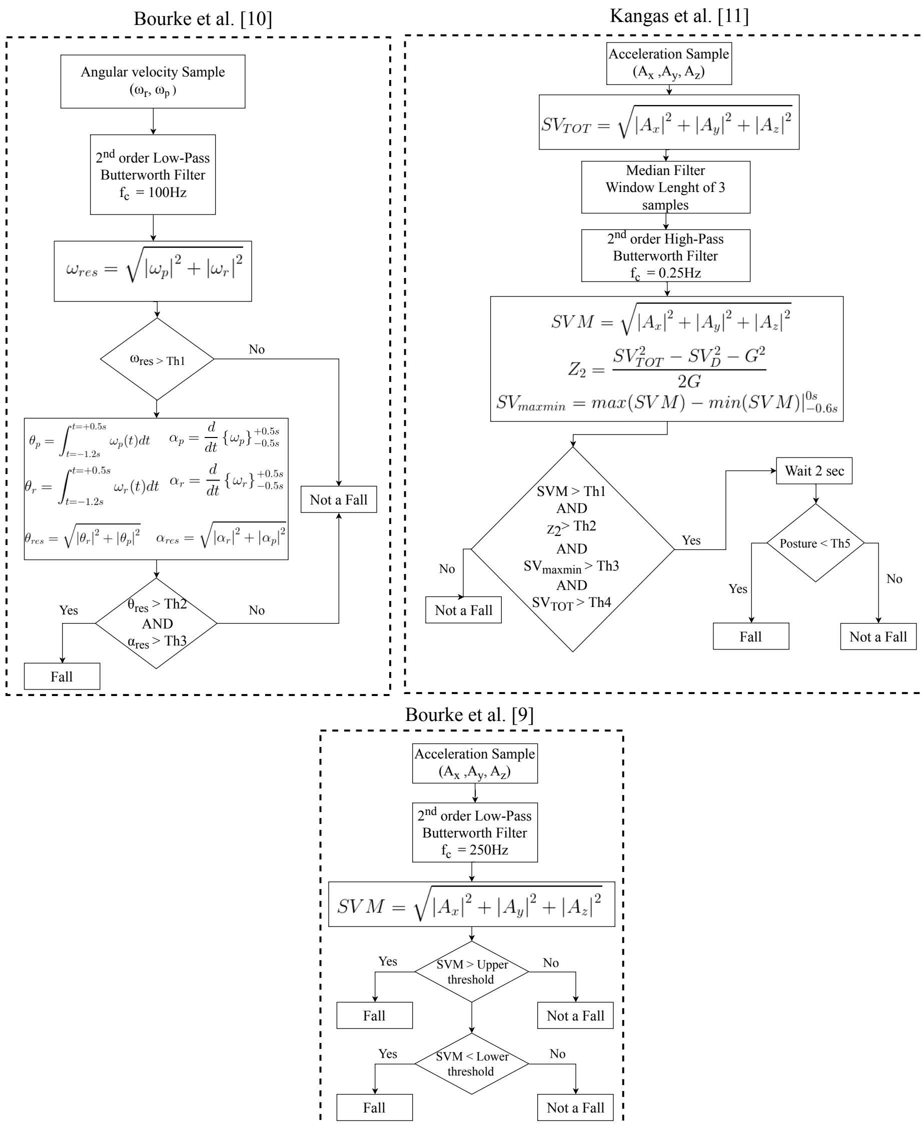


**Fig. 2.** Three different fixed threshold algorithms implemented into the ASCane.

**Table 1.** Threshold values for the different fixed threshold fall detection algorithms [9–11].

| Study | Parameter | Location | Value | Type of Threshold* |
|---|---|---|---|---|
| Bourke et al. [9] | SVM (g) | Trunk | 3.52 | UFT |
| | | | 0.41 | LFT |
| | | Thigh | 2.74 | UFT |
| | | | 0.60 | LFT |
| Bourke et al. [10] | $\omega_{res}$ (rads/s) | Trunk | 3.1 | UFT |
| | $\alpha_{res}$ (rads/s$^2$) | | 0.05 | |
| | $\theta_{res}$ (rad) | | 0.59 | |
| Kangas et al. [11] | SVM (g) | Waist | 2.0 | |
| | | Head | 2.0 | |
| | | Wrist | 5.2 | |
| | $SV_D$ (g) | Waist | 1.7 | |
| | | Head | 1.2 | |
| | | Wrist | 5.1 | |
| | $Z_2$ (g) | Waist | 1.5 | |
| | | Head | 1.8 | |
| | | Wrist | 3.9 | |
| | $SV_{MaxMin}$ (g) | Waist | 2.0 | |
| | | Head | 1.7 | |
| | | Wrist | 6.5 | |

*UFT – Upper Fall Threshold. LFT – Lower Fall Threshold

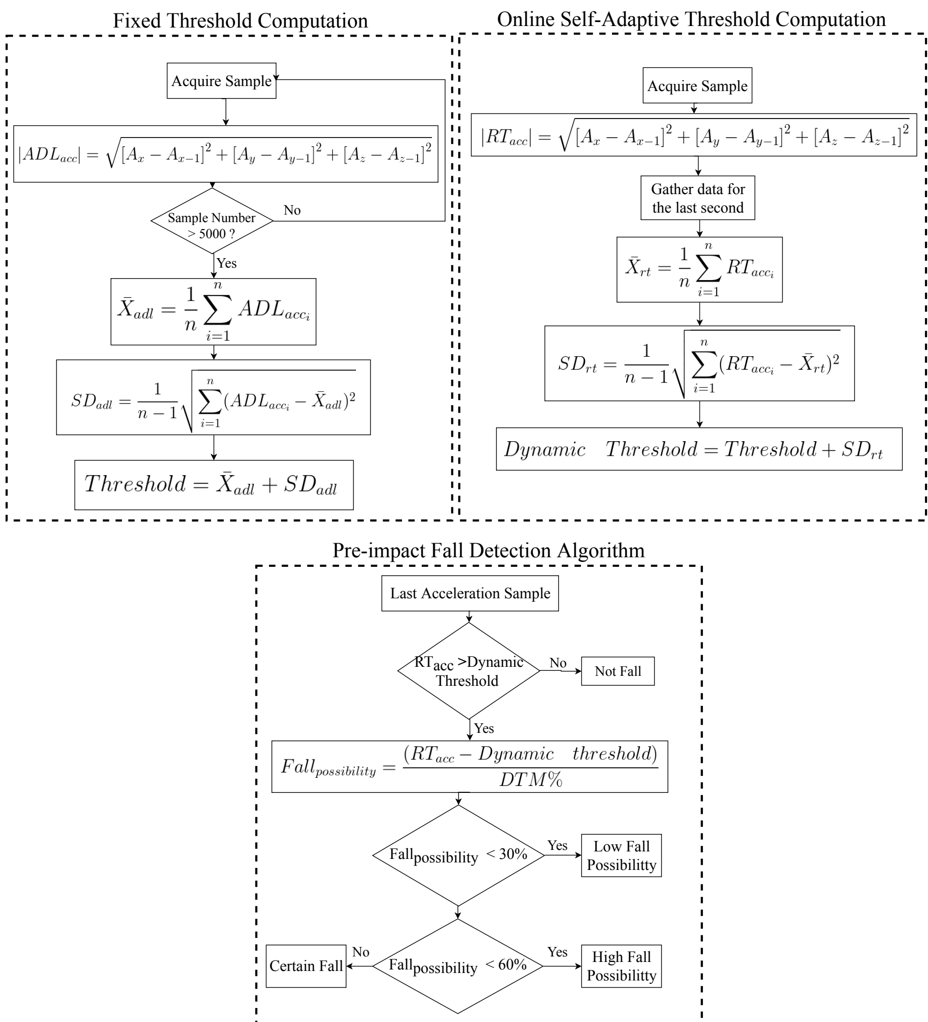


**Fig. 3.** Self Adaptive Threshold Algorithm presented by Otanasap et al. [13]

Different tests were accomplished by varying the kernel type and proportion of samples between classes in the SVM Classifier. However, the best set of parameters was determined by enabling the "OptimizeHyperparameters" option in MATLAB.

**Table 2.** Summary of the features which may correlate with falls-risk in the selected fall detection algorithms [18–21]

| Study | Feature Name |
|---|---|
| Shibuya et al. [18] | Range of angular velocity for each individual axis ($R_{\omega.x}$ . $R_{\omega.y}$ and $R_{\omega.z}$) |
| | Range of acceleration for each individual axis ($R_{A.x}$ . $R_{A.y}$ and $R_{A.z}$) |
| Liu et al. [19] | Sum Vector Magnitude (SVM) |
| | Fast Changed Vector ($CV_{Fast}$) |
| | Vertical Acceleration ($Z_2$) |
| Chen et al. [20] | Sum Vector Magnitude (SVM) |
| | Rotation angle (RA) |
| | Slope (SL) |
| | The acceleration in the xy – plane ($A_{xz}$) |
| Putra et al. [21] | Sum Vector Magnitude (SVM) |
| | Maximum Sum Vector Magnitude ($Max_{SVM}$) |
| | Minimum Sum Vector Magnitude ($Min_{SVM}$) |
| | Average Sum Vector Magnitude ($Avg_{SVM}$) |
| | Root mean square of the acceleration vector magnitude ($RMS_{SVM}$) |
| | Acceleration exponential moving average (EMA) |
| | Signal magnitude area (SMA) |

Features regarding ADL and falls were labeled using the parameter $CV_{Fast}$ to mark the falling range [19]. The maximum $CV_{Fast}$ of each fall trial was calculated and multiplied by 0.87. The samples higher than $0.87CV_{Fast}$ were considered a fall and labeled as 1.

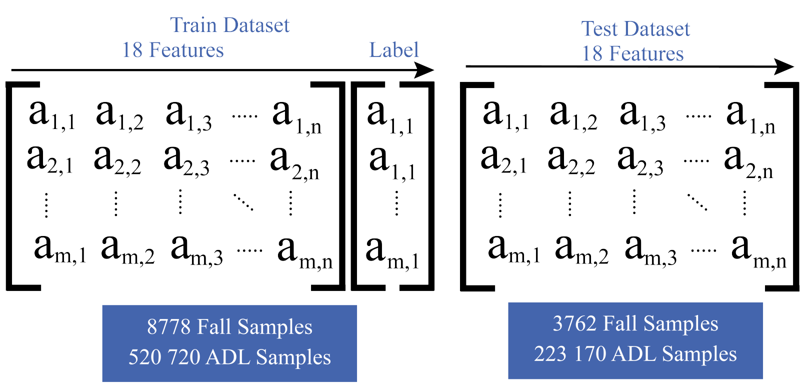


**Fig. 4.** Data structures achieved for each class (Fall + ADL) after differentiating the percentage for train and test of the classifier.

### 2.3 Experimental Protocol

A set of activities (Table 3 and Fig. 5) was executed by eleven volunteers which ranged from 22 to 29 years (24.2 ±2.6 years), with a body mass between 52 and 80 kg (70.8 ±8.23 Kg) and a height of 1.51 to 1.83 m (1.73 ±0.086m). All participants provided their written consent. Each activity was performed three times. A total of 132 simulated falls were recorder with 66 combining the subject and cane (Activities 6 and 7) and 66 only with the cane (Activities 4 and 5). Also, 99 ADL were registered (Activities 1, 2 and 3). The algorithms were implemented offline using the Matlab 2017b version.

**Table 3.** Activities simulated with the ASCane Prototype

| Activity No. | Description |
|---|---|
| 1 | Walking at Normal Speed and 180º rotation (Subject + Cane) |
| 2 | Walk forward and turn right (Subject + Cane) |
| 3 | Walk forward and turn left (Subject + Cane) |
| 4 | Free Falling (Cane) |
| 5 | Thrown out (Cane) |
| 6 | Falling Forward (Subject + Cane) |
| 7 | Falling Sideways (Subject + Cane) |

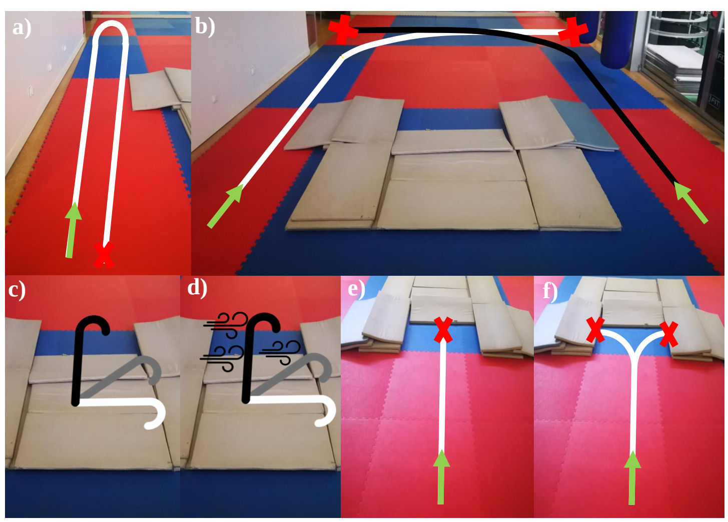


**Fig. 5.** Activities performed for data acquisition: a) Activity 1; b) Activities 2 and 3; c) Activity 4; d)Activity 5; e)Activity 6; f)Activity 7.

### 2.4 Evaluation of Classification Performance

Several performance indicators were calculated such as accuracy [18–20], precision [21], sensitivity [18–21], specificity [18–20], cohen kappa [22], and Matthews Correlation Coefficient (MCC) [22] to compare the different fall detection algorithms. The True Negatives (TN) correspond to the number of ADL correctly classified. True Positives (TP) are the falls proper classified. False Positives (FP) are the ADLs not right classified. Finally, False Negatives (FN) are the falls incorrectly classified.

## 3 Results

### 3.1 Original Algorithms

The algorithms were tested with the acquired data using the original thresholds. The results of the different performance indicators are summarized in Table 4.

**Table 4.** Performance Indicators of fall detection algorithms

| Type | Study | Details | Accuracy | Precision | Sensitivity | Specificity | MCC | Kappa |
|---|---|---|---|---|---|---|---|---|
| Fixed Threshold | Bourke et al. [9] | [1]Trunk | 0.5746 | 0.5708 | 1 | 0.0202 | 0.1074 | 0.023 |
| | | [1]Thigh | 0.5658 | 0.5658 | 1 | 0 | NaN[6] | 0 |
| | Bourke et al. [10] | [1]Trunk | 0.8114 | 0.9388 | 0.7132 | 0.9394 | 0.6534 | 0.6296 |
| | Kangas et al. [11] | [1]Waist | 0.5789 | 0.5740 | 0.9922 | 0.0404 | 0.1105 | 0.0367 |
| | | [1]Head | 0.5658 | 0.5658 | 1 | 0 | NaN[6] | 0 |
| | | [1]Wrist | 0.5789 | 0.9714 | 0.2636 | 0.9899 | 0.3485 | 0.2282 |
| Dynamic Threshold | Otanasap N. [13] | [2]0.0740 | 0.5658 | 0.5658 | 1 | 0 | NaN[6] | 0 |
| Machine Learning | Support Vector Machine | [3]1:60 [4]RBF | 0.9913 | 0.9744 | 0.4863 | 0.9998 | 0.6852 | 0.6449 |
| | | [3]1:1.6 [4]RBF | 0.9154 | 0.9390 | 0.8347 | 0.9660 | 0.8211 | 0.8178 |
| | | [3]1:1.6 [4]Linear | 0.9105 | 0.9329 | 0.8273 | 0.9627 | 0.8106 | 0.8070 |
| | | [3]1:1.6 [5]Optimized | 0.9121 | 0.9358 | 0.8289 | 0.9643 | 0.8141 | 0.8105 |

[1]Location; [2]Fixed threshold; [3]ADL:Fall Proportion; [4]Kernel Function; [5]Optimized with MATLAB; [6]Not a Number

The algorithm introduced by Bourke et al. [9] presented similar results for the two sets of thresholds described (Table 4). It detected a fall in 100% of the cases. However, all or almost all the ADLs performed were also considered a fall with a Specificity of 0 and 2.02% for the thighs and trunk, respectively.

With the method presented by Kangas et al. [11], the results are similar to the ones reached by Bourke et al. [9] in the three different sets of thresholds (Table 4). Nevertheless, while with the waist and head thresholds a fall is detected in 99.22% and 100% of the cases, respectively, the thresholds for the wrist detected only 26.36% of falls. Using the algorithm from Bourke et al. [10], it resulted in an overall higher performance compared to the remaining fixed threshold algorithms (Table 4), achieving an accuracy of 81.14%. Similar to Bourke et al. [9] and Kangas et al. [11], with the dynamic algorithm proposed by Otanasap et al. [13], a fall was spotted 100% of the cases, yet, the entirely ADL dataset was also assessed as a fall (Table 4). With the machine learning approach, the best set of parameters achieved an accuracy of 91.54% , sensitivity of 83.47% and specificity of 96.60%. The results for all accomplished tests are revealed in Table 4.

### 3.2 Modified algorithms

Both falls and ADLs present a similar acceleration maximum as identified in Table 5 and Fig. 6, which explains why the algorithm by Bourke et al. [9] was not able to detect ADLs. Thus, the algorithm was tested with a single lower threshold. The corresponding results are presented in Table 6. On the contrary, the ($\omega_{res}$) does not exhibit the same behavior as the acceleration.

**Table 5.** Maximum, minimum, mean and standard Deviation of the acceleration Sum Vector Magnitude and the angular velocity for the intentional falls and ADL trials

| Feature | Type of Activity | Maximum | Minimum | Mean | Standard Deviation |
|---|---|---|---|---|---|
| SVM (g) | ADL | 13.8357 | 0.1351 | 1.0557 | 0.3427 |
| | Fall | 13.8980 | 0.0681 | 3.8644 | 3.8296 |
| $\omega_{res}$ (rad/s) | ADL | 3.5636 | 0 | 0.6711 | 0.5440 |
| | Fall | 12.6706 | 0 | 2.7512 | 1.89002 |

The algorithm present by Otanasap et al. [13] was also not able to detect ADLs. An analysis of the features behaviour throughout the trials was accomplished, Fig. 7, and the algorithm was tested with several different FT which results are indicated in Table 7.

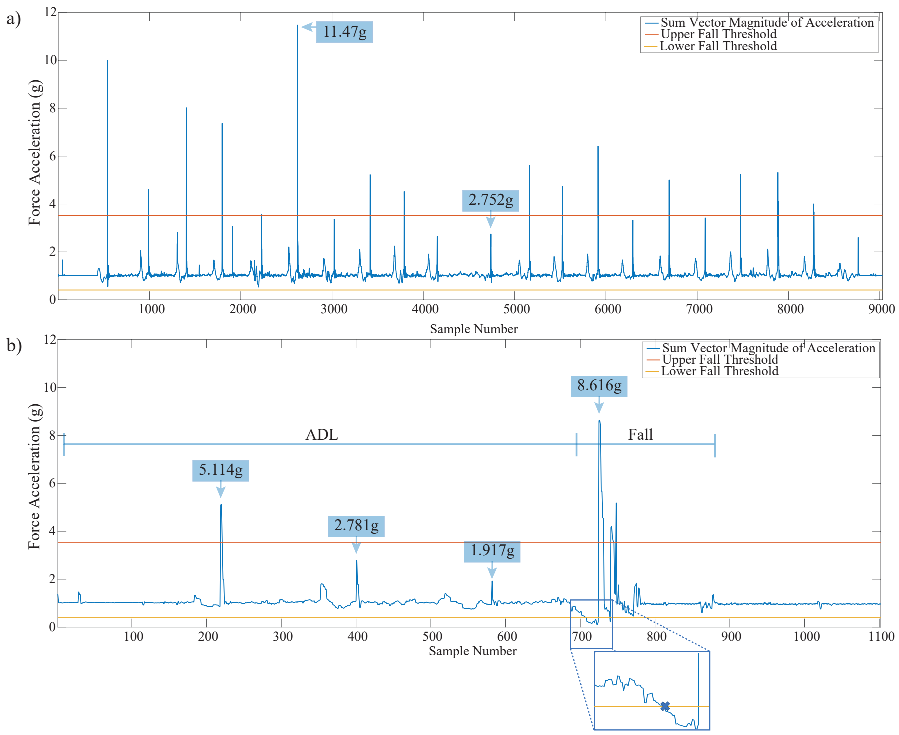


**Fig. 6.** Sum Vector Magnitude for: a) One ADL trial; b) One intentional fall trial with the corresponding fall detection as a result of the lower threshold of 0.41g.

**Table 6.** Performance indicators of the fall detection algorithm proposed by Bourke et al. [9] tested only with a single lower threshold

| Lower Threshold | Accuracy | Precision | Sensitivity | Specificity | MCC | Kappa |
|---|---|---|---|---|---|---|
| 0.41 | 0.9190 | 0.8815 | 0.9917 | 0.8222 | 0.8406 | 0.8312 |
| 0.2 | 0.9781 | 0.9920 | 0.9690 | 0.9898 | 0.9559 | 0.9555 |

**Table 7.** Performance Indicators of the fall detection algorithm proposed by Otanasap et al. [13] tested with different FT

| FT | Accuracy | Precision | Sensitivity | Specificity | MCC | Kappa |
|---|---|---|---|---|---|---|
| 7 | 0.8478 | 0.8444 | 0.9157 | 0.7455 | 0.6796 | 0.6756 |
| 7.2 | 0.8229 | 0.8488 | 0.9125 | 0.7679 | 0.6945 | 0.6914 |
| 7.4 | 0.8636 | 0.8750 | 0.8974 | 0.8148 | 0.7167 | 0.7163 |
| 7.6 | 0.8837 | 0.9155 | 0.8784 | 0.8909 | 0.7648 | 0.7639 |
| 7.8 | 0.8819 | 0.9104 | 0.8714 | 0.8947 | 0.7633 | 0.7624 |
| 8 | 0.8810 | 0.9206 | 0.8529 | 0.9138 | 0.7643 | 0.7619 |

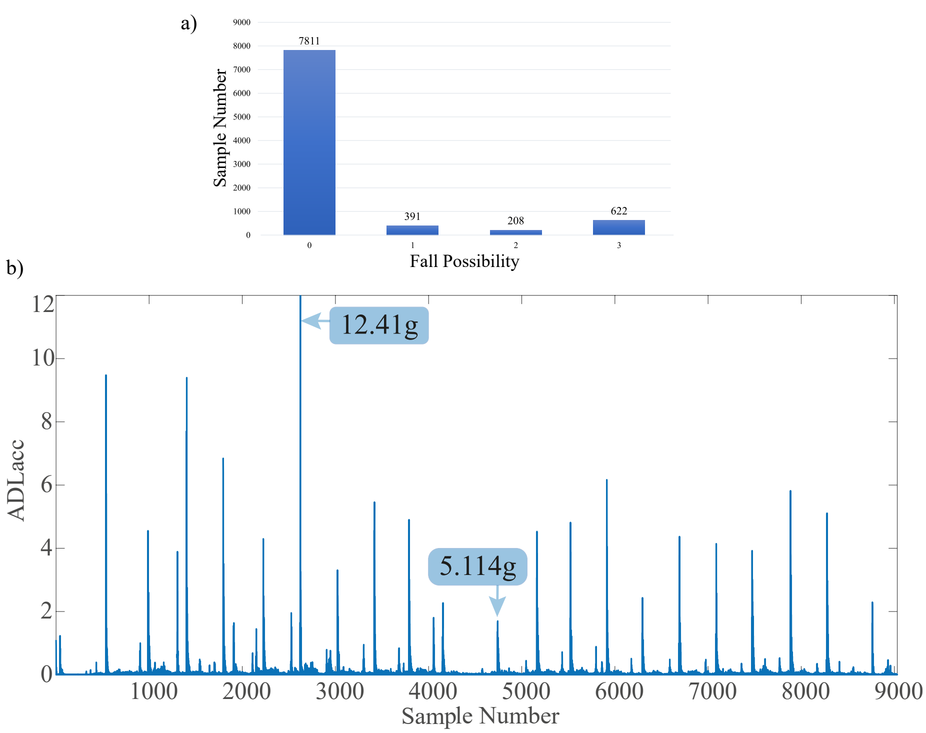


**Fig. 7.** a) Fall possibility computed by the algorithm proposed by [13] during an ADL trial b)$\mathrm{ADL}_{acc}$ of the same trial.

## 4 Discussion

The algorithm introduced by Bourke et al. [9] considered a fall in almost all ADL trials. This indicates that the original thresholds are not appropriate or adapted to canes considering that when the cane hits the ground, there is a substantial increase in the Sum Vector Magnitude, Fig. 6 a), similarly to the trials of falls, Fig. 6 b). Since the upper threshold is frequently surpassed when the cane hits the ground, contrarily to the lower threshold, Fig. 6 a), the algorithm was tested with different lower thresholds. Consequently, the performance was significantly higher, Table 6. Thus, the upper thresholds with Sum Vector Magnitude on canes are not recommended due to the aforementioned problem. This feature is also directly related to the force applied to the cane for each strike with the floor, and it is different for every gait cycle (Fig. 6 a)).

Regarding the study from Kangas et al. [11], none of the set of thresholds are suitable to canes. Both waist and head thresholds detect falls in almost ADL trials and the wrist thresholds only detects a fall in 26% of the cases (Table 3). Considering that the five features used to evaluate the trial are accelerometry based, all of them will be affected when the cane hits the ground. Therefore, using this algorithm with the original thresholds is inefficient. Proposing a new set of threshold requires a more complex analysis of the data.

Due to the fact that peak values of $\omega_{res}$ for the recorded ADLs and falls are different (5), the first threshold of 3.1 rad/s ($\omega_{res}$) is hardly ever surpassed as can be seen in Fig. 8 for one trial. Thus, the algorithm described by Bourke et al. [10] presented the best results among the fixed threshold fall detection algorithms. However, when using a single lower acceleration threshold of 0.2g, the accuracy increased to 97.81%, which is better than the results attained by the aforementioned algorithms.

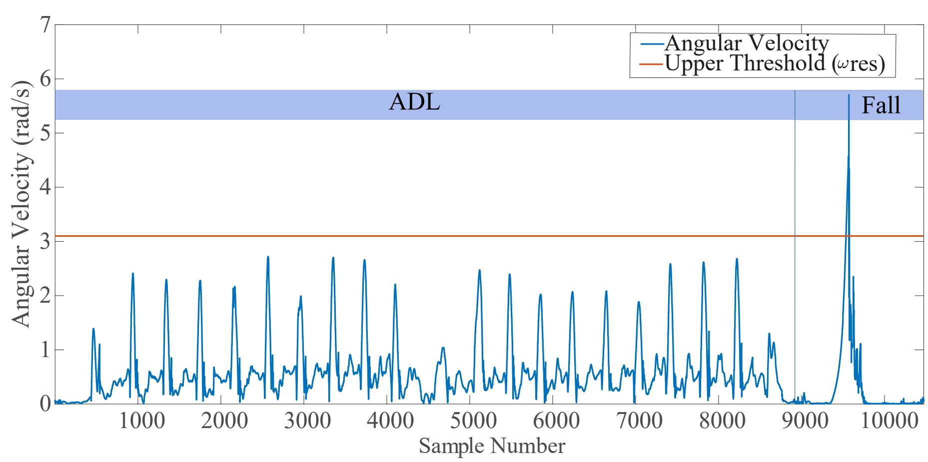


**Fig. 8.** Angular Velocity of an ADL trial versus a simulated fall trial.

Since the algorithm introduced by [13] is mainly based upon the $\text{ADL}_{acc}$, it is expected a lower performance compared to the results stated in this study because this feature is accelerometry based.

As seen in Fig. 7. b), during an ADL trial, the $\text{ADL}_{acc}$ surpasses the fixed threshold numerous times as proven by the fall possibility computed and plotted in Fig. 7. a). Thus, this method is not optimized for cane systems with the original FT. Consequently, the dynamic algorithm was tested with several different FT (Table 7). The best performance was achieved by a new FT of 7.6g.

Class imbalance is a commonly problem faced in data mining due to imbalanced datasets [21]. In this situation, the amount of samples from ADL is immensely larger than the number of fall samples with a proportion of 60:1. From Table 4, when the classifier was trained with an imbalanced dataset, it achieved an accuracy of 99.13%. However, the classifier is overfitting the data. Afterwards, when the classifier was trained with a proportion of 1:1.6 (Table 4), the sensitivity improved by almost 40% in the three other cases. However, when using the RBF kernel, the best result in this domain was achieved with a specificity and sensitivity of 96.60% and 83.47%, respectively.

Comparing the MCC and Kappa values from the implemented algorithms, the embedment of a single lower threshold of 0.2g is more desirable (MCC = 95.59%; Kappa = 95.55%). This method surpasses the values of the machine learning implementation which has a range of MCC between 0.68 and 0.82 and a Kappa between 0.69 and 0.82.

## 5 Conclusions

This paper describes and analyses the results of five fall detection algorithms implemented in the ASCane. When using the original fixed thresholds, falls were not detected effectively because accelerometry based features were affected by the impact of the cane on the ground on each gait cycle. This event has similar acceleration values to a fall and justifies the low accuracy results. The dynamic threshold method was also inefficient in detecting ADLs since it always considered them as a fall. The application of Support Vector Machine achieved great results when compared to the dynamic and fixed threshold algorithms. After class balancing, a sensitivity, specificity and MCC of 83.47%, 96.60% and 82.11% were obtained, respectively. However, the best performance was achieved by the algorithm proposed by Bourke et al. [9] that was modified by the authors. With a single lower threshold of 0.2g, values of sensitivity, specificity and MCC were 96.90%, 98.98% and 95.59%, respectively. Results obtained from the machine learning classifier were lower when compared to the proposed method by the authors likely because of the sample labeling method used, the $CV_{Fast}$. This method could be inappropriate for data acquired with a cane, and may need to be improved.

## 6 Acknowledgment

This work is supported by the FCT - Fundação para a Ciência e Tecnologia - with the scholarship reference PD/BD/141515/2018, with the reference project UID/EEA/04436/2013, by FEDER funds through the COMPETE 2020 - Programa Operacional Competitividade e Internacionalização (POCI) - with the reference project POCI-01-0145-FEDER-006941.

### Conflict of Interest

The authors declare that they have no conflict of interest.

## References

1. World Health Organization: Falls - WHO (2018). URL http://www.who.int/news-room/fact-sheets/detail/falls. (Date last accessed 01-Nov-2018)
2. Burns, E.R., Stevens, J.A., Lee, R.: The direct costs of fatal and non-fatal falls among older adults United States. Journal of Safety Research **58**, 99–103 (2016). DOI 10.1016/j.jsr.2016.05.001
3. Lachtar, A., Val, T., Kachouri, A., Lachtar, A., Val, T., Kachouri, A.: 3DCane : a monitoring system for the elderly using a connected walking stick. International Journal of Computer Science and Information Security **14**(8), 1–8 (2017)
4. Cates, B., Sim, T., Heo, H.M., Kim, B., Kim, H., Mun, J.H.: 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) **18**(4) (2018). DOI 10.3390/s18041227
5. Xu, T., Zhou, Y., Zhu, J.: New Advances and Challenges of Fall Detection Systems: A Survey. Applied Sciences **8**(3), 418 (2018). DOI 10.3390/app8030418
6. Shi, G., Chan, C.S., Li, W.J., Leung, K.S., Zou, Y., Jin, Y.: Mobile human airbag system for fall protection using mems sensors and embedded SVM classifier. IEEE Sensors Journal **9**(5), 495–503 (2009). DOI 10.1109/JSEN.2008.2012212
7. Bateni, H., Maki, B.E.: Assistive devices for balance and mobility: Benefits, demands, and adverse consequences. Archives of Physical Medicine and Rehabilitation **86**(1), 134–145 (2005). DOI 10.1016/j.apmr.2004.04.023
8. Luz, C., Bush, T., Shen, X., Pruchno, R.: Do canes or walkers make any difference? nonuse and fall injuries. Gerontologist **57**(2), 211–218 (2017). DOI 10.1093/geront/gnv096
9. Bourke, A.K., O'Brien, J.V., Lyons, G.M.: Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. Gait and Posture **26**(2), 194–199 (2007). DOI 10.1016/j.gaitpost.2006.09.012
10. Bourke, A.K., Lyons, G.M.: A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. Medical Engineering and Physics **30**(1), 84–90 (2008). DOI 10.1016/j.medengphy.2006.12.001
11. Kangas, M., Konttila, A., Lindgren, P., Winblad, I., Jämsä, T.: Comparison of low-complexity fall detection algorithms for body attached accelerometers. Gait and Posture **28**(2), 285–291 (2008). DOI 10.1016/j.gaitpost.2008.01.003

12. Ashfak Habib, M., Mohktar, M.S., Bahyah Kamaruzzaman, S., Seang Lim, K., Maw Pin, T., Ibrahim, F.: Smartphone-based solutions for fall detection and prevention: Challenges and open issues. Sensors (Switzerland) **14**(4), 7181–7208 (2014). DOI 10.3390/s140407181
13. Otanasap, N.: Pre-impact fall detection based on wearable device using dynamic threshold model. Parallel and Distributed Computing, Applications and Technologies, PDCAT Proceedings pp. 362–365 (2017). DOI 10.1109/PDCAT.2016.083
14. Yan, Q., Huang, J., Luo, Z.: Human-robot coordination stability for fall detection and prevention using cane robot. 2016 International Symposium on Micro-NanoMechatronics and Human Science, MHS 2016 (1) (2017). DOI 10.1109/MHS.2016.7824171
15. Bouten, C.V., Koekkoek, K.T., Verduin, M., Kodde, R., Janssen, J.D.: A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity. IEEE Transactions on Biomedical Engineering **44**(3), 136–147 (1997). DOI 10.1109/10.554760
16. Chen, P.H., Li, Y.H., Chiou, C.W., Lee, C.Y., Lin, J.M.: A smart safety cane for human fall detection. International Journal of Ad Hoc and Ubiquitous Computing **20**(1), 49–65 (2015). DOI 10.1504/IJAHUC.2015.071662
17. Özdemir, A.T.: An analysis on sensor locations of the human body for wearable fall detection devices: Principles and practice. Sensors (Switzerland) **16**(8) (2016). DOI 10.3390/s16081161
18. Shibuya, N., Nukala, B.T., Rodriguez, A.I., Tsay, J., Nguyen, T.Q., Zupancic, S., Lie, D.Y.: A real-time fall detection system using a wearable gait analysis sensor and a Support Vector Machine (SVM) classifier. 2015 8th International Conference on Mobile Computing and Ubiquitous Networking, ICMU 2015 pp. 66–67 (2015). DOI 10.1109/ICMU.2015.7061032
19. Liu, S.H., Cheng, W.C.: Fall detection with the support vector machine during scripted and continuous unscripted activities. Sensors (Switzerland) **12**(9), 12301–12316 (2012). DOI 10.3390/s120912301
20. Chen, K.H., Yang, J.J., Jaw, F.S.: Accelerometer-based fall detection using feature extraction and support vector machine algorithms. Instrumentation Science and Technology **44**(4), 333–342 (2016). DOI 10.1080/10739149.2015.1123161. URL http://dx.doi.org/10.1080/10739149.2015.1123161
21. Putra, I.P.E.S., Brusey, J., Gaura, E., Vesilo, R.: An event-triggered machine learning approach for accelerometer-based fall detection. Sensors (Switzerland) **18**(1), 1–18 (2018). DOI 10.3390/s18010020
22. Misiunas, A., Meskauskas, T., Samaitien, R.: Accuracy of Different Machine Learning Type Methodologies for EEG Classification by Diagnosis pp. 441–448 (2019). DOI 10.1007/978-3-030-10692-8_50