# A Comparison of Machine Learning Algorithms for Fall Detection using Wearable Sensors

Nicolas Zurbuchen, Pascal Bruegger
*Institute of Complex Systems (iCoSys)*
*School of Engineering and Architecture of Fribourg Switzerland*
*HES-SO University of Applied Sciences and Arts Western Switzerland*
Fribourg, Switzerland
nicolas.zurbuchen@hes-so.ch, pascal.bruegger@hes-so.ch

Adriana Wilde
*Centre for Health Technologies (CHT)*
*School of Electronics and Computer Science*
*University of Southampton*
Southampton, United Kingdom
agw106@ecs.soton.ac.uk

***Abstract*—The proportion of people 60 years old and above is expected to double globally to reach 22% by 2050. This creates societal challenges such as the increase of age-related illnesses and the need for caregivers. Falls are a major threat for the elderly, often causing serious injuries especially when the fallen person stays on the ground for a long time without assistance.**

**This paper presents the development of a Fall Detection System (FDS) using an accelerometer combined with a gyroscope worn at the waist. Data come from *SisFall*, a publicly available dataset containing records of Activities of Daily Living and falls. We compared five Machine Learning algorithms. We first applied preprocessing and a feature extraction stage before using five Machine Learning algorithms, allowing us to compare them. *Ensemble learning* algorithms such as Random Forest and Gradient Boosting have the best performance, with a Sensitivity and Specificity both close to 99%.**

***Index Terms*—fall detection; wearable sensors; sampling rate; data preprocessing, feature extraction, machine learning**

## I. Introduction

Falls are one of the leading causes of death among the elderly [1]. Every year, 28% to 35% of the elderly fall at least once and this rate increases with age [2]. Falls can have severe physical, psychological and even social consequences. They can also heavily affect the independent quality of living. They can result in bruises and swellings, as well as fractures and traumas [3]. A significant risk is the *long-lie*. This happens when an elderly person remains on the ground for a long duration without being able to call for help. It is associated with death within the next few months following the accident [4]. It also affects the elderly's self-confidence who may develop the *fear of falling* syndrome. It leads to anxiety when performing Activities of Daily Living (ADLs) and can lead to subsequent falls [1].

Therefore, the elderly must continuously be monitored to ensure their safety. Families organize visits but these can be inconvenient and even insufficient. Hiring caregivers or moving into nursing homes are sometimes not affordable options. Recent progresses in technology have enabled the development of Assisted-Living Systems (ALSs) [5]. They can assist the elderly and provide a safer environment through constant monitoring while relieving caregivers' workload. However, ALSs create other challenges such as privacy concerns and acceptability issues that need to be addressed [6].

Fall Detection Systems (FDSs) are part of ALSs. Their goals are to identify falls and notify caregivers so that they can intervene as fast as possible. However, fall recognition is challenging from a computational perspective. Falls can be defined as "the rapid changes from the upright/sitting position to the reclining or almost lengthened position, but it is not a controlled movement" [7]. There is a higher acceleration during falls. Another challenge is that falls can happen in innumerable scenarios. They may occur anywhere at any time [3]. Their starting and ending body posture as well as their direction (e.g. forward, backward) may vary [1]. Hence, FDSs must cover the whole living area. Their reliability must be high while minimizing false alarms, all the while respecting the elderly's privacy.

In this paper, we developed a reliable FDS by the mean of wearable sensors (accelerometer and gyroscope) and various Machine Learning (ML) algorithms. The goal is to compare *lazy*, *eager* and *ensemble learning* algorithms and assess their results. We implemented five algorithms and tested them in the same setup.

The rest of this paper is organized as follows. In section II, we discuss existing FDSs and highlight their distinctive features. Section III covers the employed methodology. Section IV presents and discusses the obtained results. Finally, we conclude with a comment on future work in section V.

## II. Related work

Scientists have employed various approaches to implement FDSs over the past years. They have been classified as presented in Fig. 1. Each of them has its strengths and weaknesses. We focus on wearable technologies since we use this approach. Nevertheless, several survey studies [8], [9] reported the other methods in more depth.

### A. Choice of sensors and sampling rate

Several types of sensors including accelerometers, gyroscopes, magnetometers, and tilt sensors have been used to detect falls. Based on the fall characteristics, most studies, such as [10]–[13], employed only acceleration measurements. Only Bourke and Lyons [14] used a single biaxial gyroscope and measured changes in angular velocity, angular acceleration, and body angle. Wang *et al.* [15] employed a heart rate monitor and discovered that the heart rate increases by 22% after a fall in people over 40 years old. This demonstrates that physiological data can be used in such a system. Across these papers, the sensors' sampling rate varied within a range from 20 to 1000 Hz. This variation is not small, one having 50 times more samples than the other, seemingly arbitrary.

### B. Sensing position

The sensor placement highly affects the detection performance. Previous studies [12], [16] demonstrated that better results are achieved when sensors are placed along the longitudinal axis of the body (e.g. head, chest, waist) when compared to other placements (e.g. thigh, wrist). The movement of this axis during a fall is more consistent and steady. However, this requires to wear a dedicated device on uncommon body parts which consequently creates inconveniences. For this reason, other studies [10], [17], [18] used commodities (e.g. smartphones carried by the thigh, smartwatches worn on the wrist). These usually do not disturb the users since they already wear them. However, people tend to take these devices off when they are at home which makes the FDS useless. Another method is to combine various sensing positions. Özdemir *et al.* [19] developed a system consisting of six wearable devices that are all used together. The problem is that the elderly already have acceptability issues with one device, let alone six.

### C. Algorithms

There are two categories of algorithms: *threshold-based* and *ML-based*. Threshold algorithms simply define limit values, outside of which, a fall is detected.. They have often been sufficient but they tend to produce false alarms especially with fall-like activities such as sitting abruptly [14]. To compensate, these studies [11], [20] added simple posture and pattern recognition algorithms that detect changes in body posture and level of activity. This improves the detection's robustness while keeping a low computational complexity. However, it may still fail during specific falls and ADLs. For example, Sucerquia *et al.* [21] used a threshold-based classification over their dataset *SisFall*, achieving 96% accuracy.

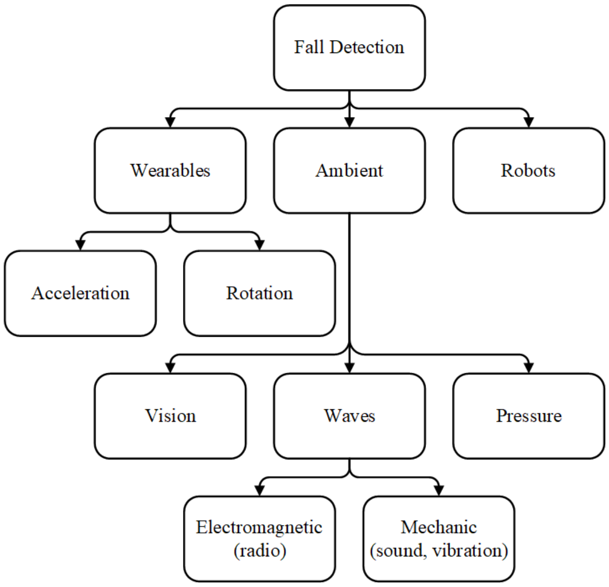

Fig. 1. Classification of Fall Detection System approaches.

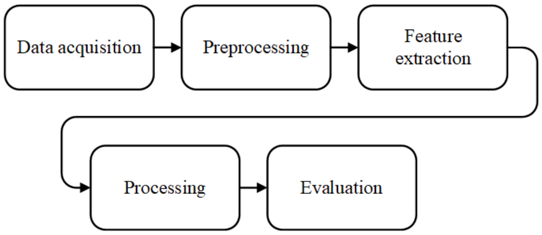

Fig. 2. General architecture of Fall Detection Systems.

ML algorithms automatically learn patterns based on data, and very commonly include feature extraction. They require more computational power and are complex to optimize but produce improved results. Most of the studies such as [10], [11] employed a supervised learning technique. Common algorithms are k-Nearest Neighbor [19], Support Vector Machine [19] and Artificial Neural Network [10], [19]. Yuwono *et al.* [13] used unsupervised learning which works with clusters. This is a compelling solution because it does not require labelled data. Deep Learning algorithms are nowadays very popular and achieve promising results in various fields. Musci *et al.* [23] employed Recurrent Neural Networks to detect falls. They used a publicly available dataset [21] and outperformed the paper's results.

### D. Strengths and weaknesses

Wearable technologies have several advantages. They are relatively inexpensive, can operate anywhere and require little computational power all of it with minimal intrusion compared to other approaches, such as environmental monitoring [16]. They can also identify the wearer and get precise measurements. However, they may create discomfort due to their size and intrusiveness. The main disadvantage is their human dependency. These sensors must have enough battery and be worn to work properly. Furthermore, the elderly may have a cognitive impairment and thus, may forget to wear the sensor.

## III. Methodology

Our FDS is based on a common pipeline (Fig. 2) which has been seen in the literature [19]. This pipeline is a common practice when working with ML algorithms. We first acquire raw data using various sensors and convert them into discrete values. We then preprocess the raw data to remove measuring errors which can badly affect the performance. Afterwards, we construct and extract meaningful information in a vector. Finally, we train and evaluate our ML algorithm to distinguish falls from ADLs.

TABLE I
DETAILS OF THE ACTIVITIES OF DAILY LIVING AND FALLS CONTAINED IN THE *SisFall* DATASET [21].

| **Activity** | **Duration [s]** |
|---|---|
| Walking slowly | 100 |
| Walking quickly | 100 |
| Jogging slowly | 100 |
| Jogging quickly | 100 |
| Walking upstairs and downstairs slowly | 25 |
| Walking upstairs and downstairs quickly | 25 |
| Slowly sit and get up in a half-height chair | 12 |
| Quickly sit and get up in a half-height chair | 12 |
| Slowly sit and get up in a low-height chair | 12 |
| Quickly sit and get up in a low-height chair | 12 |
| Sitting, trying to get up, and collapse into a chair | 12 |
| Sitting, lying slowly, wait a moment, and sit again | 12 |
| Sitting, lying quickly, wait a moment, and sit again | 12 |
| Changing position while lying (back-lateral-back) | 12 |
| Standing, slowly bending at knees, and getting up | 12 |
| Standing, slowly bending w/o knees, and getting up | 12 |
| Standing, get into and get out of a car | 25 |
| Stumble while walking | 12 |
| Gently jump without falling (to reach a high object) | 12 |
| Fall forward while walking, caused by a slip | 15 |
| Fall backward while walking, caused by a slip | 15 |
| Lateral fall while walking, caused by a slip | 15 |
| Fall forward while walking, caused by a trip | 15 |
| Fall forward while jogging, caused by a trip | 15 |
| Vertical fall while walking, caused by fainting | 15 |
| Fall while walking with damping, caused by fainting | 15 |
| Fall forward when trying to get up | 15 |
| Lateral fall when trying to get up | 15 |
| Fall forward when trying to sit down | 15 |
| Fall backward when trying to sit down | 15 |
| Lateral fall when trying to sit down | 15 |
| Fall forward while sitting, caused by fainting | 15 |
| Fall backward while sitting, caused by fainting | 15 |
| Lateral fall while sitting, caused by fainting | 15 |

### A. Dataset

We used a publicly available dataset named *SisFall* [21]. We selected this dataset over others because of its high quality. We assessed this quality with various criteria, namely the size of the dataset and the diversity of subjects in terms of age, gender, weight, and height. We also took into account the number of falls and ADLs performed by each subject. In the *SisFall* dataset, two triaxial accelerometers (ADXL345 and MMA8451Q) and a triaxial gyroscope (ITG3200) were used at a sampling rate of 200 Hz. We decided not to use the data of the second accelerometer (MMA8451Q) because usual setups only have a single accelerometer.

Twenty-three young people (19 to 30 years old) performed 15 types of falls and 19 types of ADLs including fall-like activities. Fifteen elderly people (60 to 75 years old) also performed the same ADLs for more authenticity. There were five trials per activity except for the walking and jogging activities, each of which had only one trial (See Table I). Hence, *SisFall* contains a total of 4505 records including 2707 ADLs and 1798 falls, making it unbalanced. A total of 38 people including 19 women and 19 men participated. Table I lists the falls and ADLs and their duration.

### B. Data preprocessing

The *SisFall* dataset required minimal preprocessing. We started by equalizing the duration of each record, by equally cutting (*top and tail* in equal measure) reducing the length to 10 seconds. We chose 10 seconds to remove outliers induced by the fall experiment, whilst preserving the fall within each record.

Regarding the two walking and two jogging activities, which only have one trial (Table I), we extracted 5 times 10 seconds for each record. We did this to have the same number of trials per activity. We selected five windows with no overlap along each record as follows:

1) From 5 to 15 seconds
2) From 25 to 35 seconds
3) From 45 to 55 seconds
4) From 65 to 75 seconds
5) From 85 to 95 seconds

### C. Feature extraction

We then extracted meaningful information from the preprocessed data. This process helps extracting information that better characterize each activity. A common practice, when working with time series, is to extract time and frequency domains features [19]. A time-domain feature extracted widely in the literature [10], [18], [19] is the *norm* of a sample (1).

$$Norm = \sqrt{X^2 + Y^2 + Z^2} \tag{1}$$

We calculated the *norm* of acceleration and rotation measures. However, this feature alone is not sufficient to allow a robust fall detection. In the case of fall-like activities (e.g. faster movement), this feature would probably be misleading. Thus, we also extracted time-domain features such as the *variance*, *standard deviation*, *mean*, *median*, *maximum*, *minimum*, *delta*, *25th centile*, and *75th centile*. Additionally, we extracted frequency-domain features, using a Fast Fourier Transform (FFT) and we extracted two features: the *power spectral density* and the *power spectral entropy*.

Table II summarizes the selected features. We extracted them for each axis of each sensor (3 axes, 2 sensors) but also for each sensor *norm*. This results in a feature vector of 88 features per record.

Finally, we normalized the extracted features to rescale the data to a common scale. This gives more influence to data with small values which can be neglected depending on the employed algorithm. In this work, we used the common *min-max* normalization (2) which scales the values between 0 and 1 included.

$$min\_max[0,1] \to x' = \frac{x - min(x)}{max(x) - min(x)} \tag{2}$$

### D. Classification algorithm

We selected five different types of ML algorithms among the most widely used ones. The following list shortly describes them:

TABLE II
LIST OF EXTRACTED TIME AND FREQUENCY DOMAINS FEATURES.

| Feature | Domain |
|---|---|
| Variance | Time |
| Standard deviation (STD) | Time |
| Mean | Time |
| Median | Time |
| Maximum | Time |
| Minimum | Time |
| Delta (peak-to-peak) | Time |
| 25th Centile | Time |
| 75th Centile | Time |
| Power Spectral Density (PSD) | Frequency |
| Power Spectral Entropy (PSE) | Frequency |

- *k-Nearest Neighbour (KNN)* is a simple and popular yet effective algorithm. Data are classified by a majority vote with the class most represented among its k-closest neighbours. It is a *lazy learning* type of algorithm because almost no work is done until a prediction. KNN has previously been used for fall detection and produced promising results [19].
- *Support Vector Machine (SVM)* is also commonly used in various tasks. It tries to find the best hyperplane which maximises the margins between each class. It is an *eager learning* algorithm because it works a lot during the training stage, building a model, in this case, a hyperplane. It has often been used to detect falls either with wearable, [19] where it has also produced promising results, or video-based systems.
- *Decision Tree (DT)* is tree shaped. Each node is a decision leading to a more precise category of a data for classification. It is also an *eager learning* algorithm because the tree is constructed during the training process. DTs are easy to interpret but may produce overfitting.
- *Random Forest (RF)*, as its name suggests, uses multiple DTs in parallel. Each DT is trained on a subset of data and their results are then merged to determine the most likely class. This allows reducing the overfitting generated by DTs. It is an *ensemble learning* type of algorithm because it uses multiple other algorithms.
- *Gradient Boosting (GB)* also uses multiple DTs but this time in a sequence. Each DT learns iteratively on the errors made by its predecessor. It is also an *ensemble learning* kind of algorithm. It can perform better than RF but potentially has overfitting issues.

### E. Evaluation

The performance evaluation of our FDS under the selected classifiers was done using *k-folds cross-validation*. This required splitting the dataset into $k$ sets. $k-1$ sets are used as training and 1 as testing. The process is repeated $k$ times with a different set as the test one. Given that FDSs must be able to detect falls for new people (e.g. unseen data), the test set should not contain people data that the algorithm has been trained on.

We chose a value of $k = 5$. This creates a training set of 80% and a test set of 20%. We filtered the *SisFall* to only keep subjects that performed all activities. Thus, we removed all data created by elderly people because they have not performed simulated falls. We also removed three young people data due to missing records. This leaves us with data of 20 subjects that perfectly fit the split in five folds. Consequently, we have 3400 records consisting of 1900 ADLs and 1500 falls, making the whole data more balanced.

During the evaluation of ML algorithms, each prediction falls in one of the following categories:

- *True Negative (TN)*: Correct classification of a negative condition, meaning a reject.
- *False Positive (FP)*: Incorrect classification of a negative condition, meaning a false alarm.
- *False Negative (FN)*: Incorrect classification of a positive condition, meaning a missed.
- *True Positive (TP)*: Correct classification of a positive condition, meaning a hit.

Each prediction is added to the count of its category which allows then to calculate various metrics such as the accuracy. A usual representation of these categories is a confusion matrix.

In fall detection, two metrics are especially important: Sensitivity (SE) (3) and the Specificity (SP) (4) [7]. The SE (or *recall*) corresponds to how many relevant elements are actually selected. This is basically the detection probability meaning how many falls have actually been detected. The SP corresponds to how many non-relevant elements are selected. It means how many non-falls are actually non-falls.

$$Sensitivity = \frac{TP}{TP + FN} \tag{3}$$

$$Specificity = \frac{TN}{TN + FP} \tag{4}$$

Additionally, we calculated the accuracy and the Area Under the Receiver Operating Characteristics Curve (AUROC). The AUROC is used to evaluate classifiers performance which is used in pattern recognition and ML [22]. In simple terms, an AUROC close to the value of one is indicative of a well-performing algorithm, with high true-positive and true-negative rates consistently.

## IV. Results and Discussion

Table III presents the results of the evaluation of our Fall Detection System (FDS) under the selected five Machine Learning (ML) algorithms, showing that we successfully developed a reliable FDS. Both the Sensitivity (SE) and Specificity (SP) surpassed 98% with the and the Gradient Boosting algorithm, outperforming results reported by [21]. In general, *ensemble learning* algorithms achieved better performance than others. This is because they use multiple ML algorithms, though the improvement in performance is at the expense of more resources. Support Vector Machine (SVM) had more difficulties to distinguish the activities. However, by tuning some hyperparameters, its results would likely improve. In an

FDS, it is desirable to detect every fall to ensure the elderly's safety (i.e. a perfect SE). We could then use a threshold to improve the SE, even though it would reduce the SP and raise more false alarms.

These results were unexpected especially prior optimization. We infer that Activities of Daily Living and falls in the *SisFall* dataset are discriminating by default, similar to [14]. Thus, any algorithm can perform very well. However, in real-life conditions, the SE and SP would very likely drop because of the falls heterogeneity as highlighted by Krupitzer *et al.* [16]. The difficulty of obtaining real falls data is the main shortcoming in FDS studies, given that it is challenging to capture them in realistic settings with the elderly.

TABLE III
FALL DETECTION RESULTS OF THE MACHINE LEARNING ALGORITHMS.

| Algorithm | Sensitivity [%] | Specificity [%] | Accuracy [%] | AUROC [%] |
|---|---|---|---|---|
| KNN | 97.26 | **99.31** | 98.41 | 99.45 |
| SVM | 87.93 | 93.78 | 91.20 | 96.43 |
| DT | 96.60 | 97.26 | 96.97 | 96.93 |
| RF | 98.00 | 98.94 | 98.52 | 99.90 |
| GB | **98.06** | 99.21 | **98.70** | **99.93** |

## V. CONCLUSIONS AND FUTURE WORK

In this paper, we successfully developed a Fall Detection System (FDS) using wearable technologies. Our results are an improvement over those reported by [23] and [21], with a final Sensitivity and Specificity over 98%. The system is reliable as we were able to test it on a large dataset containing several thousands of Activities of Daily Living (ADLs) and falls. We obtained these results using various Machine Learning (ML) algorithms which we were able to compare. We observed that *ensemble learning* algorithms perform better than *lazy* or *eager learning* ones.

There is scope for future work. With the high computation resources available nowadays, it would be interesting to explore Deep Learning (DL) algorithms. There is a study [23] using Recurrent Neural Networks but there are other algorithms available such as Convolutional Neural Networks with the advantage of automatic feature extraction from time series [24]. This reduces the number of steps to implement and removes the question of how many and which features are needed to be extracted. The *SisFall* dataset allows plenty of experiments. However, the lack of falls data availability in realistic settings is a common challenge in FDS studies, which also affected our study.

## REFERENCES

[1] L. Z. Rubenstein, "Falls in older people: epidemiology, risk factors and strategies for prevention," Age and Ageing, vol. 35, pp. ii37-ii41, September 2006

[2] World Health Organisation, "WHO global report on falls prevention in older age," 2008

[3] S. Sadigh, A. Reimers, R. Andersson, and L. Laflamme, "Falls and Fall-Related Injuries Among the Elderly: A Survey of Residential-Care Facilities in a Swedish Municipality," Journal of Community Health, vol. 29, pp. 129-140, April 2004

[4] D. Wild, and U. S. Nayak, and B. Isaacs, "How dangerous are falls in old people at home?," Br Med J (Clin Res Ed), vol. 282, pp. 266-268, January 1981

[5] P. Rashidi, and A. Mihailidis, "A Survey on Ambient-Assisted Living Tools for Older Adults," IEEE Journal of Biomedical and Health Informatics, vol. 17, pp. 579-590, May 2013

[6] H. Hawley-Hague, E. Boulton, A. Hall, K. Pfeiffer, and C. Todd, "Older adults' perceptions of technologies aimed at falls prevention, detection or monitoring: A systematic review," International Journal of Medical Informatics, vol. 83, pp. 416-426, June 2014

[7] N Noury, A. Fleury, P. Rumeau, A.K. Bourke, G. O. Laighin, V. Rialle, and J. E. Lundy, "Fall detection - Principles and Methods," 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 1663-1666, August 2007

[8] M. Mubashir, L. Shao, and L. Seed, "A survey on fall detection: Principles and approaches," Neurocomputing, vol. 100, pp. 144-152, January 2013

[9] X. Yu, "Approaches and principles of fall detection for elderly and patient," HealthCom 2008 - 10th International Conference on e-health Networking, Applications and Services, pp. 42-47, July 2008

[10] S. Abbate, M. Avvenuti, F. Bonatesta, G. Cola, P. Corsini, and A. Vecchio, "A smartphone-based fall detection system," Pervasive and Mobile Computing, vol. 8, pp. 883-899, December 2012

[11] A. M. Chan, N. Selvaraj, N. Ferdosi, and R. Narasimhan, "Wireless patch sensor for remote monitoring of heart rate, respiration, activity, and falls," 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 6115-6118, July 2013

[12] 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, pp. 285-291, August 2008

[13] M. Yuwono, B. D. Moulton, S. W. Su, B. G. Celler, and H. T. Nguyen, "Unsupervised machine-learning method for improving the performance of ambulatory fall-detection systems," BioMedical Engineering OnLine, vol. 11, pp. 9, February 2012

[14] A. K. Bourke, and G. M. Lyons, "A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor," Medical Engineering & Physics, vol. 30, pp. 84-90, January 2008

[15] J. Wang, Z. Zhang, B. Li, S. Lee, and R. S. Sherratt, "An enhanced fall detection system for elderly person monitoring using consumer home networks," IEEE Transactions on Consumer Electronics, vol. 60, pp. 23-29, February 2014

[16] C. Krupitzer, T. Sztyler, J. Edinger, M. Breitbach, H. Stuckenschmidt, and C. Becker, "Hips Do Lie! A Position-Aware Mobile Fall Detection System," 2018 IEEE International Conference on Pervasive Computing and Communications (PerCom), pp. 1-10, March 2018

[17] E. Casilari, and M. A. Oviedo-Jiménez, "Automatic Fall Detection System Based on the Combined Use of a Smartphone and a Smartwatch," PLOS ONE, vol. 10, pp. 1-11, November 2015

[18] T. Vilarinho, B. Farshchian, D. G. Bajer, O. H. Dahl, I. Egge, S. S. Hegdal, A. Lønes, J. N. Slettevold, and S. M. Weggersen, "A Combined Smartphone and Smartwatch Fall Detection System," 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, pp. 1443-1448, October 2015

[19] A. T. Özdemir, and B. Barshan, "Detecting Falls with Wearable Sensors Using Machine Learning Techniques," Sensors, vol. 14, pp. 10691-10708, June 2014

[20] J. Y. Hwang, J.M. Kang, and 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," The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 2204-2207, September 2004

[21] A. Sucerquia, J. D. López, and J. F. Vargas-Bonilla, "SisFall: A Fall and Movement Dataset," Sensors, vol. 17, January 2017

[22] T. Fawcett, "An introduction to ROC analysis," Pattern Recognition Letters, vol. 27, pp. 861-874, June 2006

[23] M. Musci, D. De Martini, N. Blago, T. Facchinetti, M. and Piastra, "Online Fall Detection using Recurrent Neural Networks," arXiv preprint arXiv:1804.04976, April 2018

[24] F. J. Ordóñez, and D. Roggen, "Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition," Sensors, vol. 16, pp. 115, January 2016