Fall Detection Using Smart Floor Sensor and Supervised Learning*

Ludovic Minvielle[1], Mounir Atiq[2], Renan Serra[4], Mathilde Mougeot[3], Nicolas Vayatis[5]

***Abstract*— Falls are a major risk for elderly people's health and independence. Fast and reliable fall detection systems can improve chances of surviving the accident and coping with its physical and psychological consequences. Recent research has come up with various solutions, all suffering from significant drawbacks, one of them being the intrusiveness into patient's life. This paper proposes a novel fall detection monitoring system based on a sensitive floor sensor made out of a piezoelectric material and a machine learning approach. The detection is done by a combination between a supervised Random Forest and an aggregation of its output over time. The database was made using acquisitions from 28 volunteers simulating falls and other behaviours. Unlike existent fall detection systems, our solution offers the advantages of having a passive sensor (no power supply is needed) and being completely unobtrusive since the sensor comes with the floor. Results are compared with state-of-the-art classification algorithms. On our database, good performance of fall detection was obtained with a True Positive Rate of 94.4% and a False Positive Rate of 2.4%.**

I. Introduction

Among aged people, falls are a major cause of fatal injury [1]. In fact, it has been shown that 10% to 15% of falls will cause serious physical injury to old people, and more than 30% of those above 65 y.o. fall at least once a year [2]. As elderly population is growing, the demand for surveillance system (and especially fall detection) has increased within the healthcare industry. For this purpose, numerous solutions have been proposed. In [3], fall detection systems are divided in three categories: wearable devices, ambience sensor and camera based systems. Mostly based on accelerometers, wearable devices systems are the cheapest and give good results [4] [5], but must always be attached to the person, making them a poor choice for elderly care as they can forget or simply refuse to wear them. Despite their high accuracy concerning fall detection [6], camera based systems have the disadvantage to be very intrusive in the patient's life, causing privacy issues. *Ambiance* based devices gather systems using microphones [7], vibrational data [8] or fusion between different sensors, for example camera and accelerometer [9] or sound, passive infrared (PIR) and vibration sensors [10]. As we aim at detecting falls for elderly in their daily life, the system has to be as discrete as possible.

Therefore we believe that floor sensor present a good interest as they are unobtrusive, do not suffer from occlusion (which is the case for camera or PIR based systems), and do not need the patient to wear anything. Previous work [11] showed multiple kind of solutions, including systems based on capacitive, resistive, binary switch, piezoelectric and piezoresistive sensors. When regarding floor sensors, specific attention was given to the scalability (can it be set up on a large area), ease of installation (cables, electronic), price, fragility over time, water resistance, resilience to the concrete beneath or the flooring above, heavy mass acceptance. Considering this guidance, the Tarkett engineering team came up with a piezoelectric solution. We tested different supervised methods and made our choice out of different criteria. Using our database we then trained the selected algorithm, built a macro-decision for fall detection and as a result of this work a patent was filed [12].

The work presented herein is the results of two objectives: building a fall detection algorithm from a research point of view, and making it implementable in an embedded system. Section II describes the process from signal generation to feature extraction. Section III presents the framework for fall detection and the operational work is given in section IV. Results and comparison with different algorithms are presented and discussed in section V, and last section gives perspectives.

II. Materials

A. Sensor and signal

The sensor is made out of a piezoelectric polymer. A piezoelectric material is a material that emits an electric field when mechanically stressed, and reversely, is subject to mechanical strain under electrical field. The sensor comes in the form of a thin film, directly put onto the concrete and covered by a flooring solution. We use *bands* of 60 cm wide and the length is adapted to the installation. Each band emits its own signal. In a classical installation, there are three zones equipped with several bands as shown in Fig. 1. Each band's signal is individually *pre-processed*, going through a low-pass filter and an offset correction. After being pre-processed, all signals are summed, meaning that the only

*This work was supported by Tarkett GDL S.A., Luxembourg, the Centre de Mathématiques et Leurs Applications (CMLA), École Normale Suprieure (ENS) de Cachan, CNRS, Université Paris-Saclay, 94235 Cachan, France, and the Cognition and Action Group (Cognac-G), CNRS, Université Paris Descartes, SSA, Paris, France

[1,2]L. Minvielle and M. Atiq are with Tarkett GDL S.A., Luxembourg, CMLA, ENS Cachan, Cachan, France and Cognac-G, CNRS, Université Paris Descartes, SSA, Paris, France . minvielle@cmla.ens-cachan.fr mounir.atiq@tarkett.com

[4] R. Serra is with Tarkett GDL S.A., Luxembourg renan.serra@tarkett.com

[3]M. Mougeot is associate professor in statistics and data-mining at Université Paris-Diderot, France. mathilde.mougeot@univ-paris-diderot.fr

[5]N. Vayatis is with CMLA, ENS Cachan, Cachan, France and Cognac-G, CNRS, Université Paris Descartes, SSA, Paris, France vayatis@cmla.ens-cachan.fr

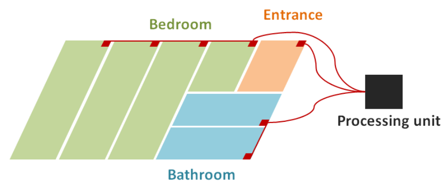


Fig. 1. Classical system installation where bands are used to cover three zones: the entrance, the bedroom and bathroom. The bands of a same zone are linked and each zone is connected to the processing unit which includes the fall detection system.

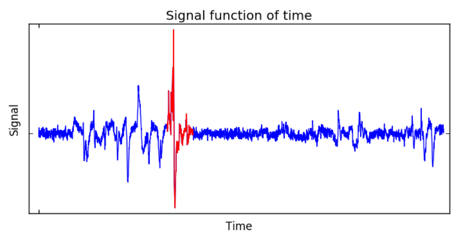


Fig. 2. Example of a fall acquisition signal. For confidentiality reasons, the signal was deliberately noised and is presented without axes units. The volunteer arrives onto the smart floor, walks a few steps, falls, stays put for a while, gets up and leave the floor. The red part is the ground truth of the fall location.

spatial information we have is the zone (three zones in the case of a classical room).

B. Database

A database was created using 28 volunteers aged from 25 to 45 years old. Following a protocol, people were falling starting from specific positions towards different directions. The non-fall events were made as varied as possible, recording walks (with one or more persons), some with a cane, movements with a chair, sitting, jumping, running, picking objects. 742 acquisitions were made, including 409 falls (i.e. 55% of the acquisitions). Recorded signals last about 20 seconds in average. All acquisitions were simultaneously video-recorded for labelling. The signal is sampled at $f_s =$ 100 Hz. As an example, a fall acquisition signal is shown Fig. 2. In our database, we observe that the falls last 1.2 seconds in average (*i.e.* 120 samples at 100 Hertz).

C. Feature extraction and database extension

For the classification problem, we consider a window of size W samples over the signal, with W previously set to 250 (*i.e.* 2.5 seconds at 100 Hertz). Instead of dealing with the signal itself, we chose to represent instances in a *feature* space. Indeed, the signal itself contains a lot of information (here it would have been represented in a W-dimension space), and part of it is not relevant for the classification. Besides, using each sample as a dimension would lead to both redundancy between dimensions, and great variance within them. Therefore, we compute 17 features over the signal itself and two other transformations of the signal:

- the first derivative of the signal, as not only the pressure variations but also their speed may matter in discriminating such events
- the Fourier transform, assuming that frequency response between a fall and other events are different

It creates then a 51-dimension feature space. These features are basic measures such as minimum or maximum values, average, variance, etc.

As we have to select a window *position* on which we calculate the features, we decided to add a random aspect to it, hence extending the database. Indeed, extracting a window out of a fall acquisition makes us naturally choose the window in which the sub-signal of the fall is *centred*. However, doing so would train our algorithm to detect centred falls in the window. Therefore, in order to improve robustness of our solution, we decided to select r windows randomly chosen over time, making sure that if we are dealing with a fall acquisition, the fall signal is encompassed in the different windows. Choosing r was done by training and testing our classifier over values of r ranging from 1 to 10.

Following sections present the adopted solution for implementation in the embedded monitoring system. State-of-the-art algorithms were tested and the choice has been done according to performance of the classifier, ease of implementation and ease of interpretation. The solution finally adopted is provided by a combination of Random Forest algorithm [13] and time aggregation of classifier outputs.

III. Supervised Learning

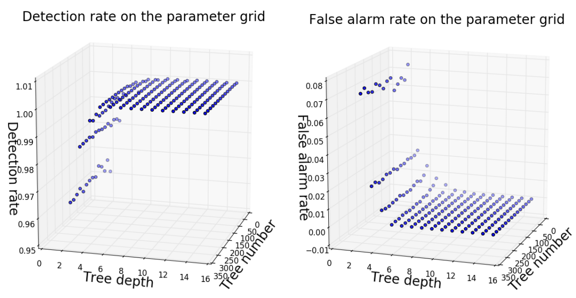
A. Random forest

A Random Forest (RF) is an aggregation of several decision trees with two major principles. First, each tree is grown from a dataset obtained by *bootstrap* of the initial training dataset. This process, called *bagging* aims at reducing the variance of the prediction function. Second, when growing a tree, each split is done on a *random subset* of m features rather than the whole set. This aims at decorrelating the trees.

The m value is commonly set at $\sqrt{Q}$, where Q is the total number of features [14]. Concerning number of trees N_T and maximum tree depth T_D, we divided the database into a training and a testing part and for each couple of parameters (N_T, T_D), we constructed multiple RFs and calculated the average Detection Rate and False Alarm Rate . This operation was done for several divisions of the database and results were averaged. These results are visible at Fig. 3.

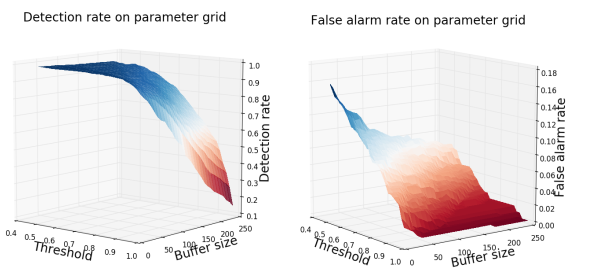
B. Macro-decision construction

As said before, our signal is sampled at frequency f_s and we selected a W-sample long window as the instance to be classified. Every $1/f_s$ second, we compute the 51 features on the window. The result is passed trough the RF giving as an output N_T *votes* (we can consider a tree output as a vote for the class *Fall*). Let $N_{active}(t)$ denote the number of trees voting for the *Fall* class at timestep t. Considering solely this output over time may be unsuccessful in terms of false alarms as a raise of $N_{active}(t)$ can trigger an alarm, as short as the raise duration may be. For that reason, we



(a) (b)

Fig. 3. Detection (a) and false alarm rate (b) on a grid of varied N_T and T_D. Above $N_T \simeq 50$ the number of trees does not have influence on the scores, while letting $T_D > 8$ gives us the best results.



(a) (b)

Fig. 4. Detection (a) and false alarm rate (b) on a grid of varied B_s and T_h. Increasing the buffer size or the threshold lowers both detection and false alarm rate. Hence, there is a trade-off between detection rate and false alarm rate that can be controlled by those two parameters.

encompass these successive micro-decisions into a macro-decision. The results of the RF are gathered over time and we construct our decision as following. We introduce two new parameters: a buffer size Bs and a threshold Th, with $0 < Th < 1$. We collect the RF votes during Bs samples, meaning that we have now $Bs \times N_T$ votes at each timestep t. Let us denote the function g as following:

$$g(t) = \frac{\sum_{i=t-B_s+1}^{t} N_{active}(i)}{Bs \times N_T} \tag{1}$$

Then we define our new binary classification function:

$$d(t) = \begin{cases} 1, & \textit{if } g(t) > T_h \\ 0, & \textit{otherwise} \end{cases} \tag{2}$$

Adding this *buffer* notion gives us control over the required detection duration and prevents short peaks in N_{active} (caused for example by falling objects) from being detected as falls. We tested the influence of these two parameters and gave an illustration of the results on Fig. 4.

IV. Feature selection for operational design

To this day, the algorithm is implemented in a embedded system with a specific architecture, hence bringing limitations in terms of both memory and computing power. Therefore, as timeframe between each diagnosis was not to be modified, a study of feature selection was done.

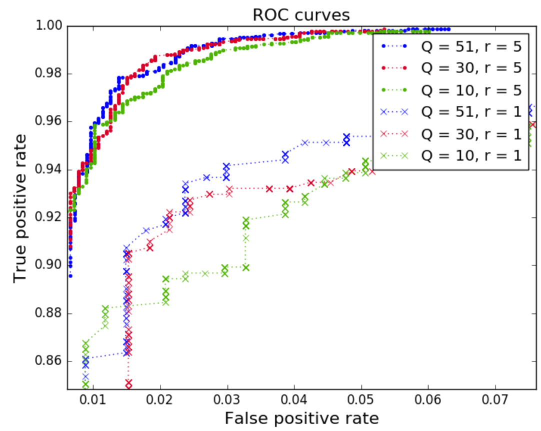


Fig. 5. ROC curves of RF classifiers trained with the whole pool of features, the 30 *best* and the 10 *best* according to our method. These RFs were also trained with $r = 1$ and $r = 5$ for comparison purpose (r is the number of randomly chosen windows over the signal). It appears that decreasing the feature space dimension while keeping the best variables still gives good results. Increasing r up to certain level gives better results, but we remind the reader that it is to be taken with precaution, as it brings redundancy into the database, especially regarding *fall* instances.

Random Forests come with an interesting measure called *variable importance* [13]. Given a variable (or feature) X_q, the importance is defined as the sum of the weighted impurity decreases for all nodes in which X_q is used, averaged over all N_T trees. It is then defined as follows:

$$Imp(X_q) = \frac{1}{N_T} \sum_{T} \sum_{t \in T : v(s_t) = X_q} p(t) \Delta i(s_t, t) \tag{3}$$

where $p(t)$ is the proportion of samples reaching the node t, $v(s_t)$ is the variable used in split s_t and $\Delta i(s_t, t)$ is the decrease of impurity $i(t)$. Therefore, ranking features according to their importance highlights those that are the most used and efficient for separating the classes. Variable reduction was done as following:

1) Using the pool of Q features, several trainings are done while recording the variable importances.
2) The average of importances over trainings is computed and we select the variable X_q* such that $Imp(X_q*)$ is minimum.
3) X_q* is removed from the pool of features and we go back to 1).

Scores of the *reduced* RFs were computed with Q ranging from 51 to 2 and ROC curves of certain of them are shown Fig. 5. Those results were provided to embedded system department, hence contributing to the feature subset selection.

V. Results and Discussion

The RF algorithm was compared to others state-of-the-art methods. This was done on our database of 742 events with $r = 5$, resulting in a database of 3710 instances. Classifiers were trained and tested using k-fold cross-validation with $k = 5$, combining both micro-decision and time aggregation. Table I gives Accuracy and F1 score for each methods we

TABLE I
RESULTS

	Accuracy	F1	FPR max (%)	FPR min (%)
LR	95.7 ± 0.6	96.1 ± 0.5	34.0	10.2
LDA	93.5 ± 0.5	94.0 ± 0.5	49.0	21.0
QDA	84.0 ± 3.3	87.2 ± 2.3	55.5	33.3
SVM linear	96.0 ± 1.2	96.3 ± 1.1	38.8	12.3
SVM (Polyn.)	91.6 ± 0.5	92.9 ± 0.4	30.1	23.4
SVM (RBF)	96.7 ± 0.2	97.0 ± 0.2	9.9	6.9
k-NN	95.4 ± 0.8	95.9 ± 0.8	12.3	6.6
MLP	98.0 ± 0.4	98.1 ± 0.4	14.7	5.7
RF	**98.4 ± 0.3**	**98.5 ± 0.3**	6.9	2.4

tested [14]. The time aggregation was also tested with B_s ranging from 1 to 300 and T_h ranging from 0 to 1. Classifiers were tested on files that did not provide the training instances. To show the result of this time aggregation, for each method we selected every pair of TPR and FPR where $\mathrm{TPR} \in [94\%, 96\%]$ and gave the maximum and minimum FPR. The feature space was scaled before each training. The selected classifiers are: Logistic Regression (LR), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Support Vector Machine (SVM, with linear, polynomial and gaussian kernel), k-Nearest Neighbours (k-NN), Multi Layer Perceptron (MLP) and Random Forest [14]. To implement them, we used the Python machine learning library Scikit-learn [15] with default parameters.

Observing Accuracy and F1-score when classifying only instances (without taking time into account), we can say that parametric statistical methods (LR, LDA, QDA) give the poorest results while non-parametric methods that contain variable selection (RF) or automatic feature construction (MLP) give best results. When considering outputs aggregation, we can see that with different pairs of B_s and T_h we can reach a TPR of about 95% while greatly reducing the FPR (in this example, FPR is divided by two when averaging over the nine methods). In the case of RF, we obtain a TPR of 94.4% with:

- a FPR at 6.9% with $T_h = 0.9$ and $B_s = 30$
- a FPR at 2.4% with $T_h = 0.9$ and $B_s = 120$

Hence, this combination proved to be useful for eliminating false alarms that would have been physically to short to be falls. To conclude, the algorithm designed for this application gives very good results on our data considering standard supervised methods. Besides, it comes with a certain ease of implementation and to some extent, a rather good interpretation of the outputs (which is limited when regarding MLPs). To finish, the feature reduction analysis allowed operational team to make the algorithm *fit* into the embedded system. Our algorithm is embedded in a system with 16 kB RAM, 256 kB program memory and a CPU speed of 40 MIPS (million instruction per second).

VI. PERSPECTIVES

The macro-decision framework gives good results. However, in order to improve performances, we might consider in future work models like Markovian approaches as they bring more temporal information [10]. Our results are executed on fixed simulated dataset and feature pool, hence bringing out some questions for the following of this work. Our first concern refers to the evolution of the dataset with real data integration partially or not labelled at all. Indeed, falls are not always acknowledged by caregivers and they might make mistakes in the process. Besides, adding data to the training process will raise the question of dealing with an imbalanced dataset (in terms of classes proportions). Moreover, in this context of real-time embedded algorithms, feature selection question regarding computational resources will reappear with hardware evolution and feature construction. Thus, to pursue this work, we might consider a general analysis in order to aim for a systematic method for resolving the trade-off between variable computation complexity and the most discriminant and less redundant list among all considered features.

ACKNOWLEDGMENT

Authors would like to thank Tarkett GDL for providing data and financial support, and are grateful to all contributors to this work.

REFERENCES

[1] C. Griffiths, C. Rooney, and A. Brock, "Leading causes of death in England and Wales–how should we group causes?" *Health statistics quarterly / Office for National Statistics*, no. 28, pp. 6–17, 2005.
[2] World Health Organization, "WHO Global Report on Falls Prevention in Older Age." *Community Health*, p. 53, 2007.
[3] M. Mubashir, L. Shao, and L. Seed, "A survey on fall detection: Principles and approaches," *Neurocomputing*, vol. 100, pp. 144–152, 2013.
[4] N. Noury, a. Fleury, P. Rumeau, a.K. Bourke, G. Laighin, V. Rialle, and J. Lundy, "Fall detection - Principles and Methods," *Conf Proc IEEE-EMBS*, pp. 1663–1666, 2007.
[5] G. Williams, K. Doughty, K. Cameron, and D. Bradley, "A smart fall and activity monitor for telecare applications," *Conf Proc IEEE EMBS*, vol. 3, no. 3, pp. 1151–1154, 1998.
[6] R. Cucchiara, A. Prati, R. Vezzani, and R. Emilia, "A multi-camera vision system for fall detection and alarm generation," *Expert Systems*, vol. 24, no. 5, pp. 334–345, 2007.
[7] X. Zhuang, J. Huang, G. Potamianos, and M. Hasegawa-Johnson, "Acoustic fall detection using gaussian mixture models and gmm supervectors," *IEEE Acoustics, Speech and Signal Processing*.
[8] N. Noury, "A smart sensor for the remote follow up of activity and fall detection of the elderly," *Conf Proc IEEE-EMBS*, pp. 314–317, 2002.
[9] A. M. Tabar, A. Keshavarz, and H. Aghajan, "Smart home care network using sensor fusion and distributed vision-based reasoning," *Conf Proc ACM workshop on VSSN '06*, p. 145, 2006.
[10] B. U. Toreyin, E. B. Soyer, I. Onaran, and E. E. Cetin, "Falling person detection using multisensor signal processing," *Eurasip Journal on Advances in Signal Processing*, vol. 2008, 2008.
[11] R. Serra, D. Knittel, P. Di Croce, and R. Peres, "Activity Recognition with Smart Polymer Floor Sensor: Application to Human Footstep Recognition," *IEEE Sensors J.*, vol. 16, no. 14, pp. 5757–5775, 2016.
[12] R. Serra, "Floor-based person monitoring system," Patent application LU93 111, June 16, 2016.
[13] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
[14] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer.
[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.